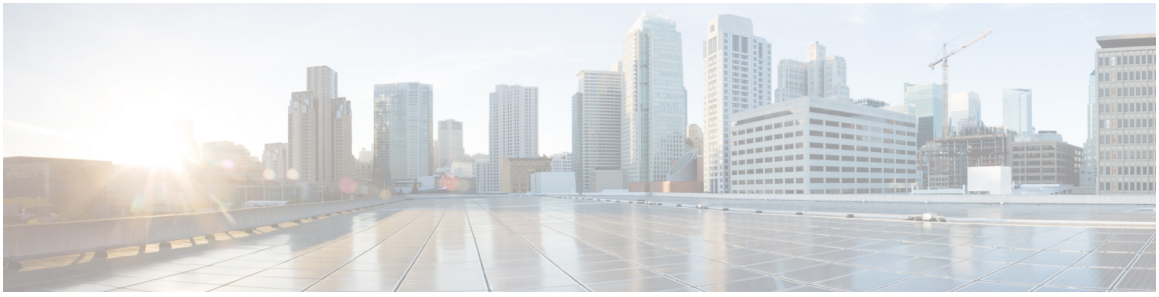
IP Addressing Configuration Guide, Cisco IOS XE 17.x

First Published: 2022-11-15

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

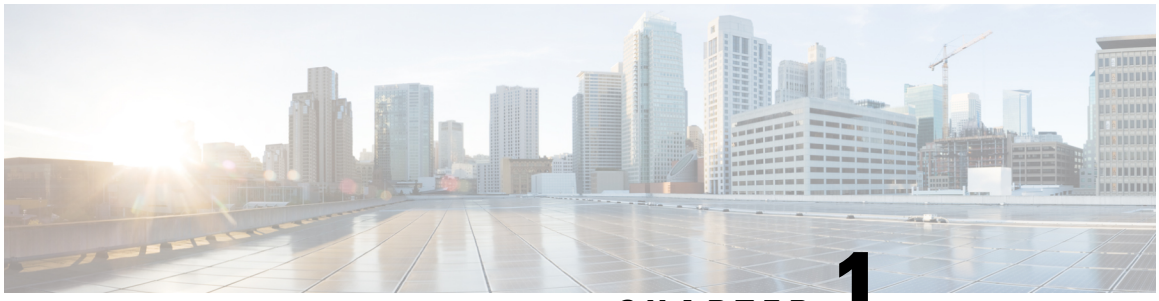
Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

PART I

IPv4 Addressing



CHAPTER 1

Configuring IPv4 Addresses

This chapter contains information about, and instructions for configuring IPv4 addresses on interfaces that are part of a networking device.

Note All further references to IPv4 addresses in this document use only IP in the text, not IPv4.

Reference the Chapter Map here

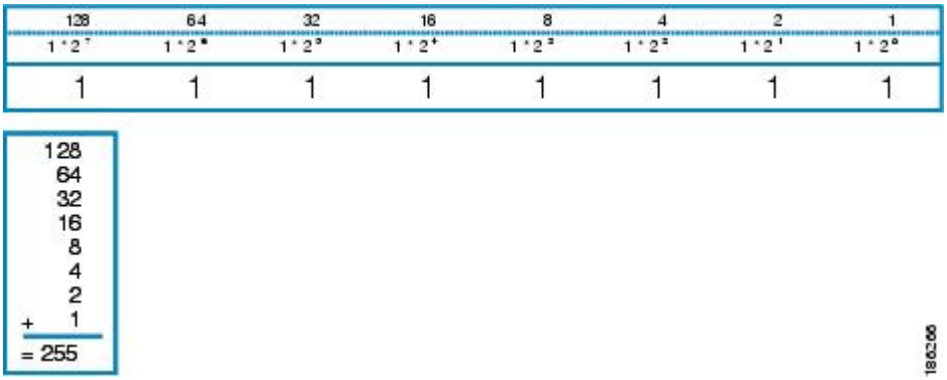
Information About IP Addresses

Binary Numbering

IP addresses are 32 bits long. The 32 bits are divided into four octets (8-bits). A basic understanding of binary numbering is very helpful if you are going to manage IP addresses in a network because changes in the values of the 32 bits indicate either a different IP network address or IP host address.

A value in binary is represented by the number (0 or 1) in each position multiplied by the number 2 to the power of the position of the number in sequence, starting with 0 and increasing to 7, working right to left. The figure below is an example of an 8-digit binary number.

Figure 1: Example of an 8-digit Binary Number



The figure below provides binary to decimal number conversion for 0 through 134.

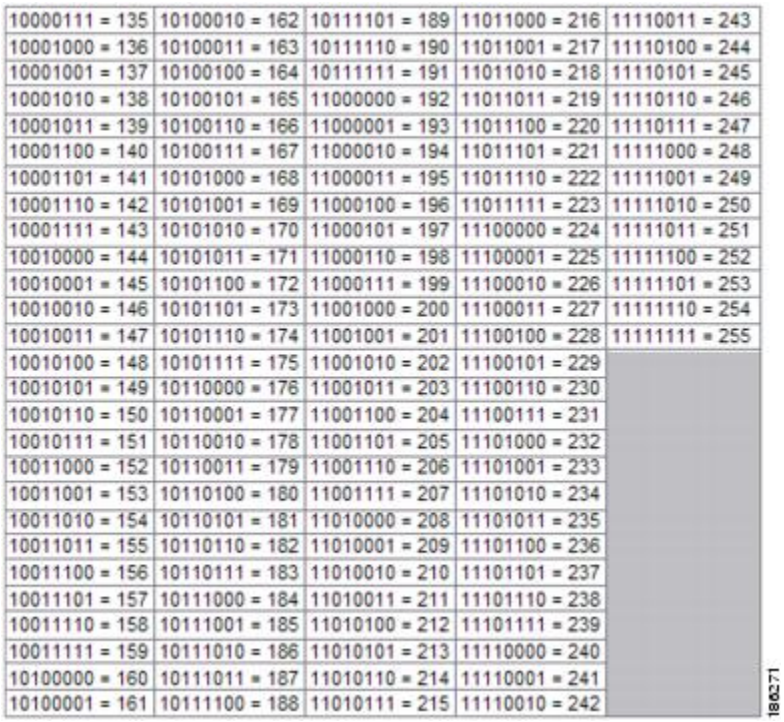
Figure 2: Binary to Decimal Number Conversion for 0 to 134

00000000 = 000	00011011 = 027	00110110 = 054	01010001 = 081	01101100 = 108
00000001 = 001	00011100 = 028	00110111 = 055	01010010 = 082	01101101 = 109
00000010 = 002	00011101 = 029	00111000 = 056	01010011 = 083	01101110 = 110
00000011 = 003	00011110 = 030	00111001 = 057	01010100 = 084	01101111 = 111
00000100 = 004	00011111 = 031	00111010 = 058	01010101 = 085	01110000 = 112
00000101 = 005	00100000 = 032	00111011 = 059	01010110 = 086	01110001 = 113
00000110 = 006	00100001 = 033	00111100 = 060	01010111 = 087	01110010 = 114
00000111 = 007	00100010 = 034	00111101 = 061	01011000 = 088	01110011 = 115
00001000 = 008	00100011 = 035	00111110 = 062	01011001 = 089	01110100 = 116
00001001 = 009	00100100 = 036	00111111 = 063	01011010 = 090	01110101 = 117
00001010 = 010	00100101 = 037	01000000 = 064	01011011 = 091	01110110 = 118
00001011 = 011	00100110 = 038	01000001 = 065	01011100 = 092	01110111 = 119
00001100 = 012	00100111 = 039	01000010 = 066	01011101 = 093	01111000 = 120
00001101 = 013	00101000 = 040	01000011 = 067	01011110 = 094	01111001 = 121
00001110 = 014	00101001 = 041	01000100 = 068	01011111 = 095	01111010 = 122
00001111 = 015	00101010 = 042	01000101 = 069	01100000 = 096	01111011 = 123
00010000 = 016	00101011 = 043	01000110 = 070	01100001 = 097	01111100 = 124
00010001 = 017	00101100 = 044	01000111 = 071	01100010 = 098	01111101 = 125
00010010 = 018	00101101 = 045	01001000 = 072	01100011 = 099	01111110 = 126
00010011 = 019	00101110 = 046	01001001 = 073	01100100 = 100	01111111 = 127
00010100 = 020	00101111 = 047	01001010 = 074	01100101 = 101	10000000 = 128
00010101 = 021	00110000 = 048	01001011 = 075	01100110 = 102	10000001 = 129
00010110 = 022	00110001 = 049	01001100 = 076	01100111 = 103	10000010 = 130
00010111 = 023	00110010 = 050	01001101 = 077	01101000 = 104	10000011 = 131
00011000 = 024	00110011 = 051	01001110 = 078	01101001 = 105	10000100 = 132
00011001 = 025	00110100 = 052	01001111 = 079	01101010 = 106	10000101 = 133
00011010 = 026	00110101 = 053	01010000 = 080	01101011 = 107	10000110 = 134

186267

The figure below provides binary to decimal number conversion for 135 through 255.

Figure 3: Binary to Decimal Number Conversion for 135 to 255



IP Address Structure

An IP host address identifies a device to which IP packets can be sent. An IP network address identifies a specific network segment to which one or more hosts can be connected. The following are characteristics of IP addresses:

- IP addresses are 32 bits long
- IP addresses are divided into four sections of one byte (octet) each
- IP addresses are typically written in a format known as dotted decimal

The table below shows some examples of IP addresses.

Table 1: Examples of IP Addresses

IP Addresses in Dotted Decimal	IP Addresses in Binary
10.34.216.75	00001010.00100010.11011000.01001011
172.16.89.34	10101100.00010000.01011001.00100010
192.168.100.4	11000000.10101000.01100100.00000100

Note The IP addresses in the table above are from RFC 1918, *Address Allocation for Private Internets* . These IP addresses are not routable on the Internet. They are intended for use in private networks. For more information on RFC1918, see http://www.ietf.org/rfc/rfc1918.txt .

IP addresses are further subdivided into two sections known as network and host. The division is accomplished by arbitrarily ranges of IP addresses to classes. For more information see RFC 791 Internet Protocol at http://www.ietf.org/rfc/rfc0791.txt .

IP Address Classes

In order to provide some structure to the way IP addresses are assigned, IP addresses are grouped into classes. Each class has a range of IP addresses. The range of IP addresses in each class is determined by the number of bits allocated to the network section of the 32-bit IP address. The number of bits allocated to the network section is represented by a mask written in dotted decimal or with the abbreviation /*n* where *n* = the numbers of bits in the mask.

The table below lists ranges of IP addresses by class and the masks associated with each class. The digits in bold indicate the network section of the IP address for each class. The remaining digits are available for host IP addresses. For example, IP address 10.90.45.1 with a mask of 255.0.0.0 is broken down into a network IP address of 10.0.0.0 and a host IP address of 0.90.45.1.

Table 2: IP Address Ranges by Class with Masks

Class	Range
A (range/mask in dotted decimal)	**0** .0.0.0 to 127.0.0.0/8 (255.0.0.0)
A (range in binary)	**00000000** .00000000.00000000.00000000 to**01111111**.00000000.00000000.00000000
A (mask in binary)	11111111.00000000.00000000.00000000/8
B (range/mask in dotted decimal)	**128** .0.0.0 to **191**.**255**.0.0/16 (255.255.0.0)
B (range in binary)	**10000000** .**00000000**.00000000.00000000 to**10111111**.**11111111**.00000000.00000000
B (mask in binary)	**11111111** .**11111111**.00000000.00000000/16
C (range/mask in dotted decimal)	**192** .**0**.**0**.0 to **223**.**255**.**255**.0/24 (255.255.255.0)
C (range in binary)	**11000000** .**00000000**.**00000000**.00000000 to**11011111**.**11111111**.**11111111**.00000000
C (mask in binary)	11111111.11111111.11111111.0000000/24
D[1] (range/mask in dotted decimal)	**224** .**0**.**0**.**0** to **239**.**255**.**255**.**255**/32 (255.255.255.255)
D (range in binary)	**11100000** .**00000000**.**00000000**.**00000000** to**11101111**.**11111111**.**11111111**.**11111111**

Class	Range
D (mask in binary)	11111111.11111111.11111111.11111111/32
E[2] (range/mask in dotted decimal)	**240** .**0**.**0**.**0** to **255**.**255**.**255**.**255**/32 (255.255.255.255)
E (range in binary)	**11110000** .**00000000**.**00000000**.**00000000** to**11111111**.**11111111**.**11111111**.**11111111**
E (mask in binary)	11111111.11111111.11111111.11111111/32

[1] Class D IP addresses are reserved for multicast applications.
[2] Class E IP addresses are reserved for broadcast traffic.

Note Some IP addresses in these ranges are reserved for special uses. For more information refer to RFC 3330, *Special-Use IP Addresses* , at http://www.ietf.org/rfc/rfc3330.txt .

When a digit that falls within the network mask changes from 1 to 0 or 0 to 1 the network address is changed. For example, if you change 10101100.00010000.01011001.00100010/16 to 10101100.00110000.01011001.00100010/16 you have changed the network address from 172.16.89.34/16 to 172.48.89.34/16.

When a digit that falls outside the network mask changes from 1 to 0 or 0 to 1 the host address is changed. For example, if you change 10101100.00010000.01011001.00100010/16 to 10101100.00010000.01011001.00100011/16 you have changed the host address from 172.16.89.34/16 to 172.16.89.35/16.

Each class of IP address supports a specific range of IP network addresses and IP host addresses. The range of IP network addresses available for each class is determined with the formula 2 to the power of the number of available bits. In the case of class A addresses, the value of the first bit in the 1st octet (as shown in the table above) is fixed at 0. This leaves 7 bits for creating additional network addresses. Therefore there are 128 IP network addresses available for class A (27 = 128).

The number of IP host addresses available for an IP address class is determined by the formula 2 to the power of the number of available bits minus 2. There are 24 bits available in a class A addresses for IP host addresses. Therefore there are 16,777,214 IP hosts addresses available for class A ((224) - 2 = 16,777,214)).

Note The 2 is subtracted because there are 2 IP addresses that cannot be used for a host. The all 0's host address cannot be used because it is the same as the network address. For example, 10.0.0.0 cannot be both a IP network address and an IP host address. The all 1's address is a broadcast address that is used to reach all hosts on the network. For example, an IP datagram addressed to 10.255.255.255 will be accepted by every host on network 10.0.0.0.

The table below shows the network and host addresses available for each class of IP address.

Table 3: Network and Host Addresses Available for Each Class of IP Address

Class	Network Addresses	Host Addresses
A	128	16,777,214

Class	Network Addresses	Host Addresses
B	16,384[3]	65534
C	2,097,152[4]	254

[3] Only 14 bits are available for class B IP network addresses because the first 2 bits are fixed at 10 as shown in Table 2 .

[4] Only 21 bits are available for class C IP network addresses because the first 3 bits are fixed at 110 as shown in Table 2 .

IP Network Subnetting

The arbitrary subdivision of network and host bits in IP address classes resulted in an inefficient allocation of IP space. For example, if your network has 16 separate physical segments you will need 16 IP network addresses. If you use 16 class B IP network addresses, you would be able to support 65,534 hosts on each of the physical segments. Your total number of supported host IP addresses is 1,048,544 (16 * 65,534 = 1,048,544). Very few network technologies can scale to having 65,534 hosts on a single network segment. Very few companies need 1,048,544 IP host addresses. This problem required the development of a new strategy that permitted the subdivision of IP network addresses into smaller groupings of IP subnetwork addresses. This strategy is known as subnetting.

If your network has 16 separate physical segments you will need 16 IP subnetwork addresses. This can be accomplished with one class B IP address. For example, start with the class B IP address of 172.16.0.0 you can reserve 4 bits from the third octet as subnet bits. This gives you 16 subnet IP addresses 24 = 16. The table below shows the IP subnets for 172.16.0.0/20.

Table 4: Examples of IP Subnet Addresses using 172.16.0.0/20

Number	IP Subnet Addresses in Dotted Decimal	IP Subnet Addresses in Binary
0[5]	172.16.0.0	10101100.00010000.00000000.00000000
1	172.16.16.0	10101100.00010000.00010000.00000000
2	172.16.32.0	10101100.00010000.00100000.00000000
3	172.16.48.0	10101100.00010000.00110000.00000000
4	172.16.64.0	10101100.00010000.01000000.00000000
5	172.16.80.0	10101100.00010000.01010000.00000000
6	172.16.96.0	10101100.00010000.01100000.00000000
7	172.16.112.0	10101100.00010000.01110000.00000000
8	172.16.128.0	10101100.00010000.10000000.00000000
9	172.16.144.0	10101100.00010000.10010000.00000000
10	172.16.160.0	10101100.00010000.10100000.00000000
11	172.16.176.0	10101100.00010000.10110000.00000000

Number	IP Subnet Addresses in Dotted Decimal	IP Subnet Addresses in Binary
12	172.16.192.0	10101100.00010000.11000000.00000000
13	172.16.208.0	10101100.00010000.11010000.00000000
14	172.16.224.0	10101100.00010000.11100000.00000000
15	172.16.240.0	10101100.00010000.11110000.00000000

[5] The first subnet that has all of the subnet bits set to 0 is referred to as subnet 0 . It is indistinguishable from the network address and must be used carefully.

When a digit that falls within the subnetwork (subnet) mask changes from 1 to 0 or 0 to 1 the subnetwork address is changed. For example, if you change 10101100.00010000.01011001.00100010/20 to 10101100.00010000.01111001.00100010/20 you have changed the network address from 172.16.89.34/20 to 172.16.121.34/20.

When a digit that falls outside the subnet mask changes from 1 to 0 or 0 to 1 the host address is changed. For example, if you change 10101100.00010000.01011001.00100010/20 to 10101100.00010000.01011001.00100011/20 you have changed the host address from 172.16.89.34/20 to 172.16.89.35/20.

Timesaver To avoid having to do manual IP network, subnetwork, and host calculations, use one of the free IP subnet calculators available on the Internet.

Some people get confused about the terms network address and subnet or subnetwork addresses and when to use them. In the most general sense the term network address means “the IP address that routers use to route traffic to a specific network segment so that the intended destination IP host on that segment can receive it”. Therefore the term network address can apply to both non-subnetted and subnetted IP network addresses. When you are troubleshooting problems with forwarding traffic from a router to a specific IP network address that is actually a subnetted network address, it can help to be more specific by referring to the destination network address as a subnet network address because some routing protocols handle advertising subnet network routes differently from network routes. For example, the default behavior for RIP v2 is to automatically summarize the subnet network addresses that it is connected to their non-subnetted network addresses (172.16.32.0/24 is advertised by RIP v2 as 172.16.0.0/16) when sending routing updates to other routers. Therefore the other routers might have knowledge of the IP network addresses in the network, but not the subnetted network addresses of the IP network addresses.

Tip The term IP address space is sometimes used to refer to a range of IP addresses. For example, “We have to allocate a new IP network address to our network because we have used all of the available IP addresses in the current IP address space”.

IP Network Address Assignments

Routers keep track of IP network addresses to understand the network IP topology (layer 3 of the OSI reference model) of the network to ensure that IP traffic can be routed properly. In order for the routers to understand

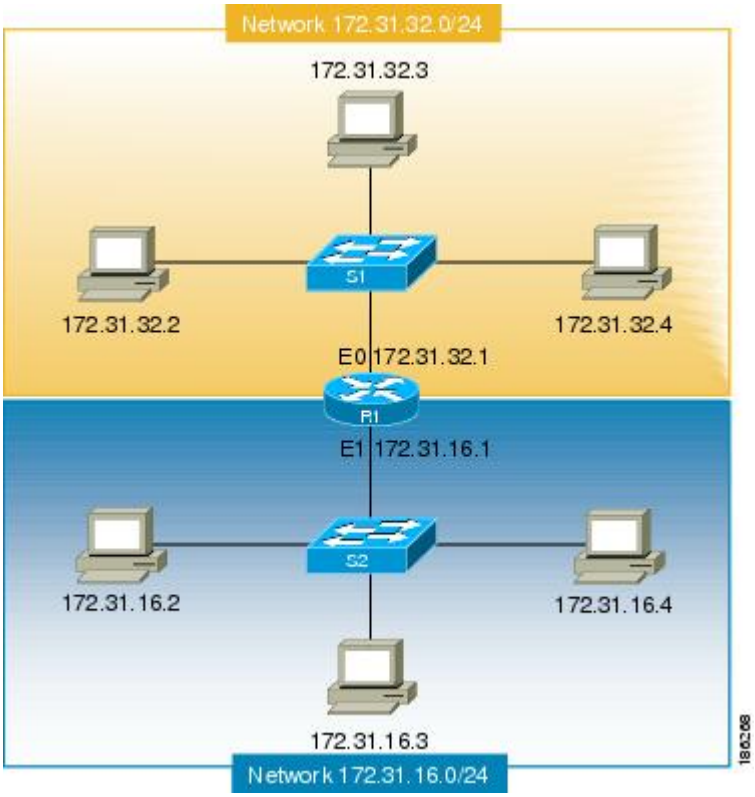
the network layer (IP) topology, every individual physical network segment that is separated from any other physical network segment by a router must have a unique IP network address.

The figure below shows an example of a simple network with correctly configured IP network addresses. The routing table in R1 looks like the table below.

Table 5: Routing Table for a Correctly Configured Network

Interface Ethernet 0	Interface Ethernet 1
172.31.32.0/24 (Connected)	172.31.16.0/24 (Connected)

Figure 4: Correctly Configured Network

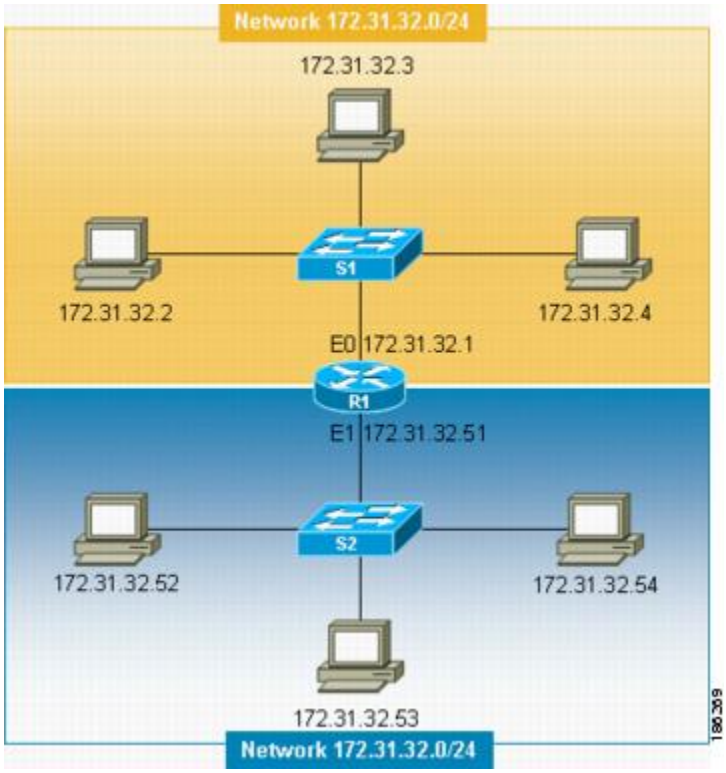


The figure below shows an example of a simple network with incorrectly configured IP network addresses. The routing table in R1 looks like the table below. If the PC with IP address 172.31.32.3 attempts to send IP traffic to the PC with IP address 172.31.32.54, router R1 cannot determine which interface that the PC with IP address 172.31.32.54 is connected to.

Table 6: Routing Table in Router R1 for an Incorrectly Configured Network (Example 1)

Ethernet 0	Ethernet 1
172.31.32.0/24 (Connected)	172.31.32.0/24 (Connected)

Figure 5: Incorrectly Configured Network (Example 1)



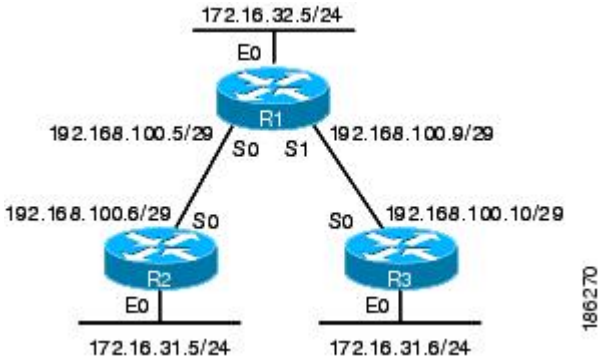
To help prevent mistakes as shown in the figure above, Cisco IOS-based networking devices will not allow you to configure the same IP network address on two or more interfaces in the router when IP routing is enabled.

The only way to prevent the mistake shown in the figure below, where 172.16.31.0/24 is used in R2 and R3, is to have very accurate network documentation that shows where you have assigned IP network addresses.

Table 7: Routing Table in Router R1 for an Incorrectly Configured Network (Example 2)

Ethernet 0	Serial 0	Serial 1
172.16.32.0/24 (Connected)	192.168.100.4/29 (Connected) 172.16.31.0/24 RIP	192.168.100.8/29 (Connected) 172.16.31.0/24 RIP

Figure 6: Incorrectly Configured Network (Example 2)



For a more thorough explanation of IP routing, see the "Related Documents" section for a list of documents related to IP routing.

Classless Inter-Domain Routing

Due to the continuing increase in internet use and the limitations on how IP addresses can be assigned using the class structure shown in the table above, a more flexible method for allocating IP addresses was required. The new method is documented in RFC 1519 *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. CIDR allows network administrators to apply arbitrary masks to IP addresses to create an IP addressing plan that meets the requirements of the networks that they administrate.

For more information on CIDR, refer to RFC 1519 at http://www.ietf.org/rfc/rfc1519.txt.

Prefixes

The term prefix is often used to refer to the number of bits of an IP network address that are of importance for building routing tables. If you are using only classful (strict adherence to A, B, and C network address boundaries) IP addresses, the prefixes are the same as the masks for the classes of addresses. For example, using classful IP addressing, a class C IP network address such as 192.168.10.0 uses a 24-bit mask (/24 or 255.255.255.0) and can also be said to have a 24-bit prefix.

If you are using CIDR, the prefixes are arbitrarily assigned to IP network addresses based on how you want to populate the routing tables in your network. For example, a group of class C IP addresses such as 192.168.10.0, 192.168.11.0, 192.168.12.0, 192.168.13.0 can be advertised as a single route to 192.168.0.0 with a 16-bit prefix (192.168.0.0/16). This results in a 4:1 reduction in the number of routes that the routers in your network need to manage.

How to Configure IP Addresses

Establishing IP Connectivity to a Network by Assigning an IP Address to an Interface

Perform this task to configure an IP address on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shutdown**
5. **ip address** *ip-address mask*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.16.1 255.255.240.0`	Configures the IP address on the interface.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface** --Displays the IP parameters for the interface.
- **show ip route connected** --Displays the IP networks the networking device is connected to.

Increasing the Number of IP Hosts that Are Supported on a Network by Using Secondary IP Addresses

If you have a situation in which you need to connect more IP hosts to a network segment and you have used all of the available IP host addresses for the subnet to which you have assigned the segment, you can avoid having to readdress all of the hosts with a different subnet by adding a second IP network address to the network segment.

Perform this task to configure a secondary IP address on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shutdown**
5. **ip address** *ip-address mask*
6. **ip address** *ip-address mask* **secondary**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.16.1 255.255.240.0`	Configures the IP address on the interface.

	Command or Action	Purpose
Step 6	**ip address** *ip-address mask* **secondary** **Example:** `Router(config-if)# ip address 172.16.32.1 255.255.240.0 secondary`	Configures the secondary IP address on the interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface** --Displays the IP parameters for the interface.
- **show ip route connected** --Displays the IP networks the networking device is connected to.

What to Do Next

If your network has two or more routers and you have already configured a routing protocol, make certain that the other routers can reach the new IP network that you assigned. You might need to modify the configuration for the routing protocol on the router so that it advertises the new network. Consult the *Cisco IOS IP Routing: Protocol-Independent Configuration Guide* for information on configuring routing protocols.

Maximizing the Number of Available IP Subnets by Allowing the Use of IP Subnet Zero

If you using subnetting in your network and you are running out of network addresses, you can configure your networking device to allow the configuration of subnet zero. This adds one more usable network address for every subnet in your IP addressing scheme. The table above shows the IP subnets (including subnet 0) for 172.16.0.0/20.

Perform this task to enable the use of IP subnet zero on your networking device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip subnet-zero**
4. **interface** *type number*
5. **no shutdown**
6. **ip address** *ip-address mask*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip subnet-zero** **Example:** `Router(config)# ip subnet-zero`	Enables the use of IP subnet zero.
Step 4	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 5	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.0.1 255.255.240.0`	Configures the subnet zero IP address on the interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface** --Displays the IP parameters for the interface.
- **show ip route connected** --Displays the IP networks the networking device is connected to.

Specifying the Format of Network Masks

By default, **show** commands display an IP address and then its netmask in dotted decimal notation. For example, a subnet would be displayed as 131.108.11.55 255.255.255.0.

You might find it more convenient to display the network mask in hexadecimal format or bit count format instead. The hexadecimal format is commonly used on UNIX systems. The previous example would be displayed as 131.108.11.55 0XFFFFFF00.

The bit count format for displaying network masks is to append a slash (/) and the total number of bits in the netmask to the address itself. The previous example would be displayed as 131.108.11.55/24.

Specifying the Format in Which Netmasks Appear for the Current Session

Perform this task to specify the format in which netmasks appear for the current session.

SUMMARY STEPS

1. **enable**
2. **term ip netmask-format** {**bitcount** | **decimal** | **hexadecimal**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**term ip netmask-format** {**bitcount**	**decimal**	**hexadecimal**} **Example:** `Router# term ip netmask-format hexadecimal`	Specifies the format the router uses to display network masks.

Specifying the Format in Which Netmasks Appear for an Individual Line

Perform this task to specify the format in which netmasks appear for an individual line.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line vty** *first last*
4. **term ip netmask-format** {**bitcount** | **decimal** | **hexadecimal**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line vty** *first last* **Example:** `Router(config)# line vty 0 4`	Enters line configuration mode for the range of lines specified by the *first* and *last* arguments.
Step 4	**term ip netmask-format** {**bitcount** | **decimal** | **hexadecimal**} **Example:** `Router(config-line)# ip netmask-format hexadecimal`	Specifies the format the router uses to display the network mask for an individual line.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Using IP Unnumbered Interfaces on Point-to-Point WAN Interfaces to Limit Number of IP Addresses Required

If you have a limited number of IP network or subnet addresses and you have point-to-point WANs in your network, you can use the IP Unnumbered Interfaces feature to enable IP connectivity on the point-to-point WAN interfaces without actually assigning an IP address to them.

Perform this task to configure the IP Unnumbered Interfaces feature on a point-to-point WAN interface.

IP Unnumbered Feature

The IP Unnumbered Interfaces feature enables IP processing on a point-to-point WAN interface without assigning it an explicit IP address. The IP unnumbered point-to-point WAN interface uses the IP address of another interface to enable IP connectivity, which conserves network addresses.

Note The following restrictions apply to the IP Unnumbered Interfaces feature:

- The IP Unnumbered Interfaces feature is only supported on point-to-point (non-multiaccess) WAN interfaces
- You cannot netboot a Cisco IOS image over an interface that is using the IP Unnumbered Interfaces feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shutdown**
5. **ip address** *ip-address mask*
6. **interface** *type number*
7. **no shutdown**
8. **ip unnumbered** *type number*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 5	**ip address** *ip-address mask* **Example:**	Configures the IP address on the interface.

	Command or Action	Purpose
	`Router(config-if)# ip address 172.16.16.1 255.255.240.0`	
Step 6	**interface** *type number* **Example:** `Router(config-if)# interface serial 0/0`	Specifies a point-to-point WAN interface and enters interface configuration mode.
Step 7	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the point-to-point WAN interface.
Step 8	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered fastethernet 0/0`	Enables the IP unnumbered feature on the point-to-point WAN interface. In this example the point-to-point WAN interface uses IP address 172.16.16.1 from Fast Ethernet 0/0.
Step 9	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface** --Displays the IP parameters for the interface.
- **show ip route connected** --Displays the IP networks the networking device is connected to.

Using IP addresses with 31-Bit Prefixes on Point-to-Point WAN Interfaces to Limit Number of IP Addresses Required

You can reduce the number of IP subnets used by networking devices to establish IP connectivity to point-to-point WANs that they are connected to by using IP Addresses with 31-bit Prefixes as defined in RFC 3021.

Perform this task to configure an IP address with a 31-bit prefix on a point-to-point WAN interface.

RFC 3021

Prior to RFC 3021, *Using 31-bit Prefixes on IPv4 Point-to-Point Links* , many network administrators assigned IP address with a 30-bit subnet mask (255.255.255.252) to point-to-point interfaces to conserve IP address space. Although this practice does conserve IP address space compared to assigning IP addresses with shorter subnet masks such as 255.255.255.240, IP addresses with a 30-bit subnet mask still require four addresses per link: two host addresses (one for each host interface on the link), one all-zeros network address, and one all-ones broadcast network address.

The table below shows an example of the four IP addresses that are created when a 30-bit (otherwise known as 255.255.255.252 or /30) subnet mask is applied to the IP address 192.168.100.4. The bits that are used to specify the host IP addresses in bold.

Table 8: Four IP Addresses Created When a 30-Bit Subnet Mask (/30) Is Used

Address	Description	Binary
192.168.100.4/30	All-zeros IP address	11000000.10101000.01100100.00000100
192.168.100.5/30	First host addresses	11000000.10101000.01100100.00000101
192.168.100.6/30	Second host address	11000000.10101000.01100100.00000110
192.168.100.7/30	All-ones broadcast address	11000000.10101000.01100100.00000111

Point-to-point links only have two endpoints (hosts) and do not require broadcast support because any packet that is transmitted by one host is always received by the other host. Therefore the all-ones broadcast IP address is not required for a point-to-point interface.

The simplest way to explain RFC 3021 is to say that the use of a 31-bit prefix (created by applying a 31-bit subnet mask to an IP address) allows the all-zeros and all-ones IP addresses to be assigned as host addresses on point-to-point networks. Prior to RFC 3021 the longest prefix in common use on point-to-point links was 30-bits, which meant that the all-zeros and all-ones IP addresses were wasted.

The table below shows an example of the two IP addresses that are created when a 31-bit (otherwise known as 255.255.255.254 or /31) subnet mask is applied to the IP address 192.168.100.4. The bit that is used to specify the host IP addresses in bold

Table 9: Two IP Addresses Created When a 31-Bit Subnet Mask (/31) Is Used

Address	Description	Binary
192.168.100.4/31	First host address	11000000.10101000.01100100.00000100
192.168.100.5/31	Second host address	11000000.10101000.01100100.00000101

The complete text for RFC 3021 is available at http://www.ietf.org/rfc/rfc3021.txt .

Before you begin

You must have classless IP addressing configured on your networking device before you configure an IP address with a 31-bit prefix on a point-to-point interface. Classless IP addressing is enabled by default in many versions of Cisco IOS software. If you are not certain that your networking device has IP classless addressing configured, enter the **ip classless** command in global configuration mode to enable it.

Note This task can only be performed on point-to-point (nonmultiaccess) WAN interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip classless**
4. **interface** *type number*
5. **no shutdown**
6. **ip address** *ip-address mask*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip classless** **Example:** `Router(config)# ip classless`	(Optional) Enables IP classless (CIDR). **Note** This command is enabled by default in many versions of Cisco IOS. If you are not certain if it is enabled by default in the version of Cisco IOS that your networking device is running, enter the **ip classless**command as shown. When you are done with this task view the configuration. If the **ip classless** command does not appear in your configuration, it is enabled by default.
Step 4	**interface** *type number* **Example:** `Router(config)# interface serial 0/0`	Specifies a point-to-point WAN interface and enters interface configuration mode.
Step 5	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.100.4 255.255.255.254`	Configures the 31bit prefix IP address on the point-to-point WAN interface.
Step 7	**end** **Example:**	Exits the current configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

Troubleshooting Tips

The following commands can help troubleshoot IP addressing:

- **show ip interface** --Displays the IP parameters for the interface.
- **show ip route connected** --Displays the IP networks the networking device is connected to.

Configuration Examples for IP Addresses

Example Establishing IP Connectivity to a Network by Assigning an IP Address to an Interface

The following example configures an IP address on three interfaces:

```
!
interface FastEthernet0/0
 no shutdown
 ip address 172.16.16.1 255.255.240.0
!
interface FastEthernet0/1
 no shutdown
 ip address 172.16.32.1 255.255.240.0
!
interface FastEthernet0/2
 no shutdown
 ip address 172.16.48.1 255.255.240.0
!
```

Example Increasing the Number of IP Hosts that are Supported on a Network by Using Secondary IP Addresses

The following example configures secondary IP addresses on three interfaces:

```
!
interface FastEthernet0/0
 no shutdown
 ip address 172.16.16.1 255.255.240.0
 ip address 172.16.32.1 255.255.240.0 secondary
!
!
interface FastEthernet0/1
 no shutdown
 ip address 172.17.16.1 255.255.240.0
 ip address 172.17.32.1 255.255.240.0 secondary
!
!
```

```
interface FastEthernet0/2
 no shutdown
 ip address 172.18.16.1 255.255.240.0
 ip address 172.18.32.1 255.255.240.0 secondary
!
```

Example Using IP Unnumbered Interfaces on Point-to-Point WAN Interfaces to Limit Number of IP Addresses Required

The following example configures the unnumbered IP feature on three interfaces:

```
!
interface FastEthernet0/0
 no shutdown
 ip address 172.16.16.1 255.255.240.0
!
interface serial0/0
 no shutdown
 ip unnumbered fastethernet0/0
!
interface serial0/1
 no shutdown
 ip unnumbered fastethernet0/0
!
interface serial0/2
 no shutdown
 ip unnumbered fastethernet0/0
!
```

Example Using IP addresses with 31-Bit Prefixes on Point-to-Point WAN Interfaces to Limit Number of IP Addresses Required

The following example configures 31-bit prefixes on two interfaces:

```
!
ip classless
!
interface serial0/0
 no shutdown
 ip address 192.168.100.2 255.255.255.254
!
!
interface serial0/1
 no shutdown
 ip address 192.168.100.4 255.255.255.254
```

Example Maximizing the Number of Available IP Subnets by Allowing the Use of IP Subnet Zero

The following example enables subnet zero:

```
!
interface FastEthernet0/0
 no shutdown
```

```
 ip address 172.16.16.1 255.255.240.0
!
ip subnet-zero
!
```

Where to Go Next

If your network has two or more routers and you have not already configured a routing protocol, consult the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4T, for information on configuring routing protocols.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP addressing commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
Fundamental principles of IP addressing and IP routing	*IP Routing Primer ISBN 1578701082*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC[6]	Title
RFC 791	*Internet Protocol* http://www.ietf.org/rfc/rfc0791.txt

RFC[6]	Title
RFC 1338	*Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy* http://www.ietf.org/rfc/rfc1519.txt
RFC 1466	*Guidelines for Management of IP Address Space* http://www.ietf.org/rfc/rfc1466.txt
RFC 1716	*Towards Requirements for IP Routers* http://www.ietf.org/rfc/rfc1716.txt
RFC 1918	*Address Allocation for Private Internets* http://www.ietf.org/rfc/rfc1918.txt
RFC 3330	*Special-Use IP Addresses* http://www.ietf.org/rfc/rfc3330.txt

[6] These references are only a sample of the many RFCs available on subjects related to IP addressing and IP routing. Refer to the IETF RFC site at http://www.ietf.org/rfc.html for a full list of RFCs.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.htmll

Feature Information for IP Addresses

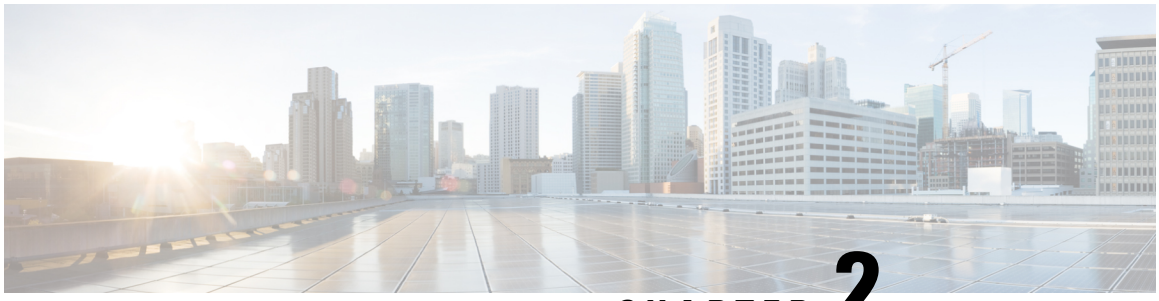
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for IP Addresses

Feature Name	Releases	Feature Information
Classless Inter-Domain Routing	10.0	CIDR is a new way of looking at IP addresses that eliminates the concept of classes (class A, class B, and so on). For example, network 192.213.0.0, which is an illegal class C network number, is a legal supernet when it is represented in CIDR notation as 192.213.0.0/16. The /16 indicates that the subnet mask consists of 16 bits (counting from the left). Therefore, 192.213.0.0/16 is similar to 192.213.0.0 255.255.0.0. The following command was introduced or modified: **ip classless**.

Feature Name	Releases	Feature Information
IP Subnet Zero	10.0	In order to conserve IP address space IP Subnet Zero allows the use of the all-zeros subnet as an IP address on an interface, such as configuring 172.16.0.1/24 on Fast Ethernet 0/0. The following command was introduced or modified: **ip subnet-zero**.
IP Unnumbered Interfaces	10.0	In order to conserve IP address space, IP unnumbered interfaces use the IP address of another interface to enable IP connectivity. The following command was introduced or modified: **ip unnumbered**.
Using 31-bit Prefixes on IP Point-to-Point Links	12.0(14)S 12.2(4)T	In order to conserve IP address space on the Internet, a 31-bit prefix length allows the use of only two IP addresses on a point-to-point link. Previously, customers had to use four IP addresses or unnumbered interfaces for point-to-point links.



CHAPTER 2

IP Overlapping Address Pools

The IP Overlapping Address Pools feature improves flexibility in assigning IP addresses dynamically. This feature allows you to configure overlapping IP address pool groups to create different address spaces and concurrently use the same IP addresses in different address spaces.

Restrictions for IP Overlapping Address Pools

The Cisco IOS XE software checks for duplicate addresses on a per-group basis. The check for duplicate addresses means that you can configure pools in multiple groups that could have possible duplicate addresses. The IP Overlapping Address Pools feature should be used only in cases where overlapping IP address pools make sense, such as Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) environments where multiple IP address spaces are supported.

Information About IP Overlapping Address Pools

Benefits

The IP Overlapping Address Pools gives greater flexibility in assigning IP addresses dynamically. It allows you to configure overlapping IP address pool groups to create different address spaces and concurrently use the same IP addresses in different address spaces.

How IP Address Groups Work

IP Control Protocol (IPCP) IP pool processing implements all IP addresses as belonging to a single IP address space, and a given IP address should not be assigned multiple times. IP developments such as virtual private dialup network (VPDN) and Network Address Translation (NAT) implement the concept of multiple IP

address spaces where it can be meaningful to reuse IP addresses, although such usage must ensure that these duplicate address are not placed in the same IP address space. An IP address group to support multiple IP address spaces and still allow the verification of nonoverlapping IP address pools within a pool group. Pool names must be unique within the router. The pool name carries an implicit group identifier because that pool name can be associated only with one group. Pools without an explicit group name are considered members of the base system group and are processed in the same manner as the original IP pool implementation.

Existing configurations are not affected by the new pool feature. The "group" concept is an extension of the existing **ip local pool** command. Processing of pools that are not specified as a member of a group is unchanged from the existing implementation.

How to Configure IP Overlapping Address Pools

Configuring and Verifying a Local Pool Group

Perform this task to configure a local pool group and verify that it exists.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip local pool** {**default** | *poolname*} {*low-ip-address* [*high-ip-address*] [**group** *group-name*] [**cache-size** *size*]}
4. **show ip local pool** [*poolname* | [**group** *group-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip local pool** {**default**	*poolname*} {*low-ip-address* [*high-ip-address*] [**group** *group-name*] [**cache-size** *size*]} **Example:** `Router(config)# ip local pool testpool 10.2.2.1 10.2.2.10 group testgroup cache-size 10000`	Configures a group of local IP address pools, gives this group a name, and specifies a cache size.
Step 4	**show ip local pool** [*poolname*	[**group** *group-name*]] **Example:**	Displays statistics for any defined IP address pools.

	Command or Action	Purpose
	`Router(config)# show ip local pool group testgroup testpool`	

Configuration Examples for Configuring IP Overlapping Address Pools

Define Local Address Pooling as the Global Default Mechanism Example

The following example shows how to configure local pooling as the global default mechanism:

```
ip address-pool local
ip local pool default 192.168.15.15 192.168.15.16
```

Configure Multiple Ranges of IP Addresses into One Pool Example

The following example shows how to configure two ranges of IP addresses for one IP address pool:

```
ip local pool default 192.169.10.10 192.169.10.20
ip local pool default 192.168.50.25 192.168.50.50
```

Additional References

The following sections provide references related to configuring IP Overlapping Address Pools.

Related Documents

Related Topic	Document Title
Dial commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Dial Services Command Reference*
IP address pooling	"Configuring Media-Independent PPP and Multilink PPP" chapter of the Cisco IOS XE Dial Technologies Configuration Guide

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 826	*Address Resolution Protocol*
RFC 903	*Reverse Address Resolution Protocol*
RFC 1027	*Proxy Address Resolution Protocol*
RFC 1042	*Standard for the Transmission of IP Datagrams over IEEE 802 Networks*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring IP Overlapping Address Pools

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for Configuring IP Overlapping Address Pools

Feature Name	Releases	Feature Information
IP Overlapping Address Pools	Cisco IOS XE Release 2.1	The IP Overlapping Address Pools feature improves flexibility in assigning IP addresses dynamically. This feature allows you to configure overlapping IP address pool groups to create different address spaces and concurrently use the same IP addresses in different address spaces. The following commands were modified by this feature:**ip local pool**and **show ip local pool.**

Glossary

IPCP --IP Control Protocol. Protocol that establishes and configures IP over PPP.

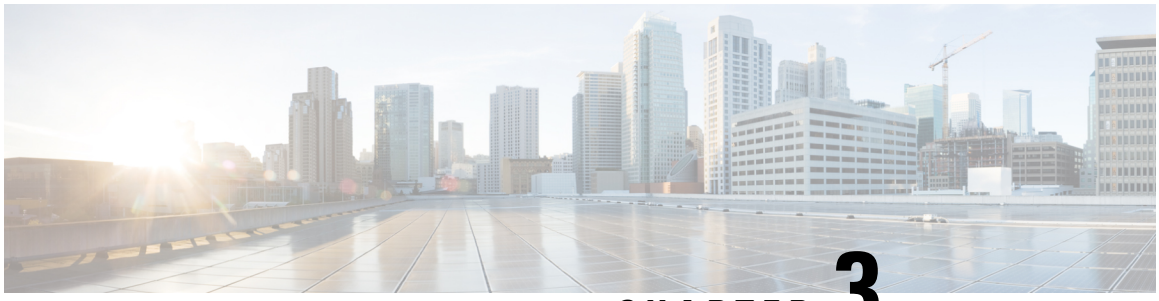
MPLS --Multiprotocol Label Switching. Switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

NAT --Network Address Translation. Mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routable address space. Also known as Network Address Translator.

VPDN --virtual private dialup network. Also known as virtual private dial network. A VPDN is a network that extends remote access to a private network using a shared infrastructure. VPDNs use Layer 2 tunnel technologies (L2F, L2TP, and PPTP) to extend the Layer 2 and higher parts of the network connection from a remote user across an ISP network to a private network. VPDNs are a cost-effective method of establishing a long distance, point-to-point connection between remote dial users and a private network. See also VPN.

VPN --Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses “tunneling” to encrypt all information at the IP level.

VRF --A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.



CHAPTER 3

IP Unnumbered Ethernet Polling Support

The IP Unnumbered Ethernet Polling Support feature provides IP unnumbered support for Ethernet physical interfaces. This support already exists for serial interfaces.

Information About IP Unnumbered Ethernet Polling Support

IP Unnumbered Ethernet Polling Support Overview

IP unnumbered support for serial interfaces is extended to Ethernet physical interfaces. Unnumbered Ethernet physical interfaces are used in the same manner as unnumbered serial interfaces. On a device, if a loopback interface is configured and an IP address is assigned to it, using the polling option more than one Ethernet physical interface can be unnumbered to the loopback.

The polling option enables the dynamic discovery of hosts (connected though the unnumbered interfaces) based on the Address Resolution Protocol (ARP) protocol.

How to Configure IP Unnumbered Ethernet Polling Support

Enabling Polling on an Ethernet Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **exit**

6. **interface** *type number*
7. **ip unnumbered** *type number* **poll**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface loopback 0`	Specifies an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.200.229 255.255.240.224`	Configures the IP address on the interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 7	**ip unnumbered** *type number* **poll** **Example:** `Device(config-if)# ip unnumbered loopback 0 poll`	Enables IP–connected host polling on the specified interface.
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring the Queue Size and the Packet Rate for IP ARP Polling for Unnumbered Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip arp poll queue** *queue-size*
4. **ip arp poll rate** *packet-rate*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip arp poll queue** *queue-size* **Example:** `Device(config)# ip arp poll queue 1000`	Configures the IP ARP polling queue size.
Step 4	**ip arp poll rate** *packet-rate* **Example:** `Device(config)# ip arp poll rate 1000`	Configures the IP ARP polling packet rate, in packets per second.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying IP Unnumbered Ethernet Polling Support

Perform this task to verify IP unnumbered Ethernet polling support.

Note The **show** commands are not in any specific order.

SUMMARY STEPS

1. **enable**
2. **show ip arp poll**
3. **show ip interface** *type number* **unnumbered**
4. **show ip interface** *type number* **unnumbered detail**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 show ip arp poll

Displays the IP ARP host polling status.

Example:

```
Device# show ip arp poll

Number of IP addresses processed for polling: 438
Number of entries in the queue: 100 (high water mark: 154, max: 1000)
Number of request dropped:
  Queue was full: 1288
  Request was throttled by incomplete ARP: 10
  Duplicate entry found in queue: 1431
```

Step 3 show ip interface *type number* unnumbered

Displays the status of unnumbered interface support on interfaces configured for IP.

Example:

```
Device# show ip interface loopback 0 unnumbered

Number of unnumbered interfaces with polling: 10
Number of IP addresses processed for polling: 15
Number of IP addresses in queue for polling: 4
```

Step 4 show ip interface *type number* unnumbered detail

Displays the detailed status of unnumbered interface support on interfaces configured for IP.

Example:

```
Device# show ip interface loopback 0 unnumbered detail

Number of unnumbered interfaces with polling: 10
Number of IP addresses processed for polling: 15
Last 10 IP addresses processed for polling:
  209.165.201.2
  209.165.201.3
  209.165.201.4
  209.165.201.5
  209.165.201.6
```

```
  209.165.201.7
  209.165.201.8
  209.165.201.9
  209.165.201.10
  209.165.201.11
Number of IP addresses in queue for polling: 4 (high water mark: 5)
  209.165.201.12
  209.165.201.13
  209.165.201.14
  209.165.201.15
```

Configuration Examples for IP Unnumbered Ethernet Polling Support

Example: Enabling Polling on an Ethernet Interface

```
Device> enable
Device# configure terminal
Device(config)# interface loopback 0
Device(config-if)# ip address 209.165.200.229 255.255.240.224
Device(config-if)# exit
Device(config)# interface ethernet 0/0
Device(config-if)# ip unnumbered loopback 0 poll
Device(config-if)# end
```

Example: Configuring the Queue Size and the Packet Rate for IP ARP Polling for Unnumbered Interfaces

```
Device> enable
Device# configure terminal
Device(config)# ip arp poll queue 1000
Device(config)# ip arp poll rate 1000
Device(config)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv4 Addressing commands	Cisco IOS IP Addressing Services Command Reference
Conceptual information about IPv4 addresses	"Configuring IPv4 Addresses" module in the *IP Addressing: IPv4 Addressing Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

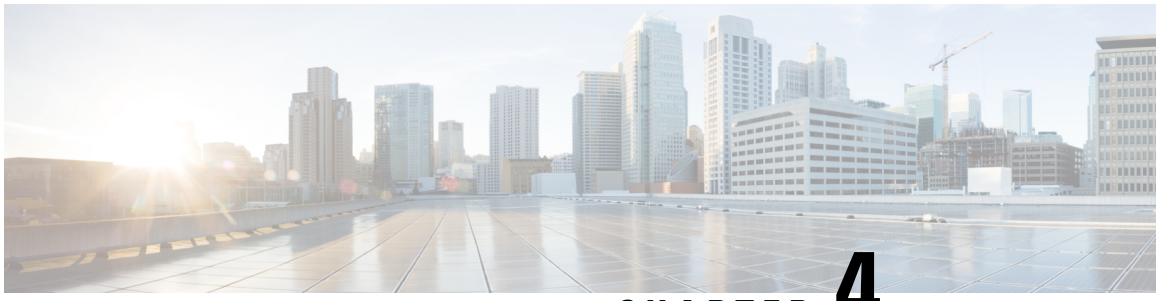
Feature Information for IP Unnumbered Ethernet Polling Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 12: Feature Information for IP Unnumbered Ethernet Polling Support

Feature Name	Releases	Feature Information
IP Unnumbered Ethernet Polling Support	Cisco IOS XE Release 3.8S	The IP Unnumbered Ethernet Polling Support feature provides IP unnumbered support for Ethernet physical interfaces. The following commands were introduced or modified: **clear ip arp poll statistics**, **clear ip interface**, **ip arp poll**, **ip unnumbered poll**, **show ip arp poll**, and **show ip interface unnumbered**.



CHAPTER 4

Auto-IP

The auto-IP feature automatically provides IP addresses to the nodes inserted into a ring. In ring topology, when a device is inserted into the ring, the neighboring node interfaces require manual reconfiguration. The auto-IP feature addresses the problem of manually reconfiguring nodes during insertion, deletion, and movement of nodes within the ring. The auto-IP feature is supported on the following:

- Ethernet interfaces and sub interfaces.
- Virtual routing and forwarding instance (VRF) interfaces.
- Switch Virtual Interfaces (SVIs).
- EtherChannels.

Attention To know the release versions that support the auto-IP feature on VRF interfaces, SVIs, and EtherChannels, refer Feature Information for Auto-IP, on page 56.

Note When a device is inserted into a ring, it is called a node.

Prerequisites for Auto-IP

- Link Layer Discovery Protocol (LLDP) must be enabled on the device before the auto-IP functionality is enabled on the node interface.

Auto-IP on an EtherChannel

- When you configure auto-IP on an EtherChannel, ensure that LLDP is enabled on the member interfaces of the EtherChannel.

- Auto-IP configuration on an interface must be removed before moving an interface into an EtherChannel.

Auto-IP on VRF interfaces

- If you intend to configure auto-IP on an interface for a specific virtual routing and forwarding instance (VRF), then ensure that the interface is presently within the VRF. If you enable auto-IP on an interface and then associate the interface to a VRF, the auto-IP settings on the interface will be cleared, and you will have to enable the auto-IP feature on the VRF interface again.

Restrictions for Auto-IP

- Auto-IP addresses must not contain an even number in the last octet (such as 10.1.1.2, where the number in the last octet is 2).

Auto-IP on VRF interfaces

- Auto-IP configuration on an interface is not retained when the interface is moved from one virtual routing and forwarding instance (VRF) to another, including the global VRF.
- Interface nodes in different VRFs cannot be configured for the same ring. Ensure that the nodes you select belong to the same VRF.
- If a VRF address family is IPv6, you cannot configure auto-IP on the interfaces within the VRF. You can configure auto-IP on a VRF interface if the VRF address family is IPv4.

Auto-IP on SVI interfaces

- Auto-IP configuration is not possible on a Switch Virtual Interface (SVI) with more than one physical interface. The SVI physical interface must be an access port or trunk port with only one associated VLAN or a bridge domain interface (BDI).

Auto-IP on EtherChannel interfaces

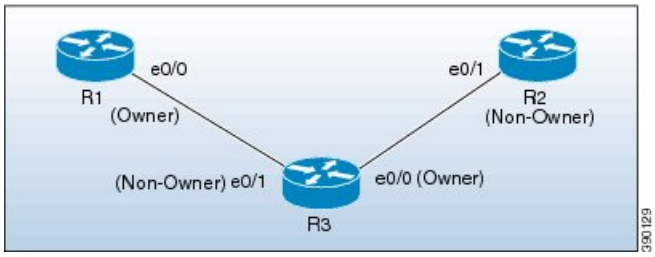
- Auto-IP configuration can be done on an EtherChannel interface, but not on a member interface of the EtherChannel.

Information About Auto-IP

Auto-IP Overview

The auto-IP feature is an enhancement of Link Layer Discovery Protocol (LLDP). LLDP uses a set of attributes to discover neighbor devices. This attribute set is called Type Length Value (TLV) as it contains type, length, and value descriptions.

In a ring topology, two network-to-network interfaces (NNIs or node interfaces) of a device are used to be part of the ring. For a ring to function as an auto-IP ring, you must configure the auto-IP feature on all the node interfaces within the ring. One node interface of a device is designated as the owner-interface and the other interface as the non-owner-interface. In an auto-IP ring, the owner-interface of a device is connected to a non-owner-interface of the neighbor device. A sample topology is given below:



When a new device is inserted into an auto-IP ring, owner and non-owner-interfaces of the inserted device are identified. The node interface of the inserted device that is connected to an owner-interface is designated as the non-owner-interface, and it automatically receives an IP address from the connected neighbor device. The IP address is automatically configured on the interface. Since the non-owner-interface is identified, the other node interface of the inserted device is designated as the owner-interface, and the device assigns a pre configured auto-IP address to its designated owner-interface.

An auto-IP address is a preconfigured address configured on a node interface to make the interface capable of automatically assigning an IP address to a new neighbor interface that is detected in the auto-IP ring. The configured auto-IP address is used for allocation purposes.

You must configure the same auto-IP address on the two node interfaces that are designated to be part of an auto-IP ring, and the auto-IP address must contain an odd number in the last octet. The auto-IP address is assigned to the owner-interface when the device is introduced into an auto-IP ring. Since each auto-IP address contains an odd number in the last octet, the IP address derived by subtracting 1 from the last octet is an even number, and is not used for designating auto-IP addresses. This IP address is allocated to a newly detected neighbor, non-owner-interface.

For example, if we assume that the device R3 is inserted between the devices R1 and R2 in the above topology, and the auto-IP address 10.1.1.3 is configured on e0/1 and e0/0, the two node interfaces on device R3, then R1 assigns an IP address to the non-owner-interface of R3, e0/1. The IP address 10.1.1.3 is assigned to the owner-interface of R3, e0/0. The IP address derived by subtracting 1 from the last octet of the auto-IP address is 10.1.1.2. 10.1.1.2 is assigned to the neighbor non-owner-interface of the connected neighbor device R2.

Auto-IP TLV exchange

Before insertion, the node interfaces are not designated as owner and non-owner. After insertion, the auto-IP TLV is exchanged between the neighbor devices. During this initial negotiation with the adjacent device interfaces, owner and non-owner-interfaces are determined automatically.

After a device is inserted into a ring, the auto-IP address configured for the device (such as 10.1.1.3) is assigned to the owner-interface for the /31 subnet. An owner-interface has a priority 2 in the auto-IP TLV, and a non-owner-interface has priority 0 in the auto-IP TLV. If there is no assigned IP address on the node interface (before the node is inserted into a ring), then the ring interface has priority 1 in the auto-IP TLV.

The IP address negotiation is based on priority; the higher value of priority wins the negotiation. If the priority is equal, then IP negotiation fails. This scenario usually occurs when there is an incorrect configuration or wiring. In such a scenario, you must ensure that the configuration and wiring is proper.

Auto-IP on VRF interfaces

Some points on auto-IP configuration on virtual routing and forwarding instance (VRF) interfaces are noted below:

- Auto-IP configuration on an interface is removed when the interface is moved from one VRF to another, including the global VRF. So, assign the interface to a VRF and then configure the auto-IP feature on the interface.
- You can configure auto-IP on a VRF interface only if the address family of the VRF is IPv4. If the IPv4 address family configuration is removed from a VRF, the auto-IP configuration is removed from all the interfaces within the VRF.
- If a VRF address family is IPv6, you cannot configure auto-IP on the interfaces within the VRF. However, if a VRF address family is IPv4 and IPv6, you can configure auto-IP on the interfaces within the VRF.
- If the IPv6 address family configuration is removed from a VRF with both IPv4 and IPv6 address-family configuration, the auto-IP configuration on the interfaces within the VRF remain intact.
- If a VRF is deleted, then the auto-IP configuration on all the interfaces assigned to the VRF are removed.
- A specific ring has two interface nodes. Ensure that the two nodes you select belong to the same VRF. Nodes in different VRFs cannot be configured for the same ring.
- Within a VRF, the same auto-IP address cannot be used for different ring IDs.

Auto-IP on EtherChannel interfaces

Some points on auto-IP configuration for an EtherChannel interface are noted below:

- You can configure auto-IP on an EtherChannel interface. If you configure the auto-IP feature on an EtherChannel and then add member interfaces to the EtherChannel, then auto-IP TLV information is carried to all the member interfaces. If you add member interfaces to the EtherChannel and then configure auto-IP on the EtherChannel, auto-IP TLV information is carried to all the member interfaces.

Attention LLDP must be enabled on the member interfaces.

- The list of EtherChannel member interfaces are maintained in ring interfaces corresponding to the EtherChannel. Auto-IP information is transmitted on all the EtherChannel member interfaces.
- If you remove a member interface from an EtherChannel, auto-IP TLV information is not carried to the removed interface.

Auto-IP on SVI interfaces

Some points on auto-IP configuration on a Switch Virtual Interface (SVI) are noted below:

- Auto-IP configuration on an SVI is possible only if a single physical interface is associated with an SVI.
- The SVI physical interface must be an access port or trunk port with only one associated VLAN or a bridge domain interface (BDI).
- If the SVI is mapped to more than one physical port, then the auto-IP configuration on the SVI will be removed.

Seed Device

Seed devices are the devices used to initiate network discovery. To initiate auto-IP capability in a ring, at least one device must be configured as a seed device in the ring. To configure a device as a seed device in an auto-IP ring, you must manually configure the IP address configured on one of its node interfaces with the auto-IP address of the interface, with the mask /31 (or 255.255.255.254).

A sample topology is given below. In this scenario, device R1 is being configured as the seed device.

The e0/0 interface on device R1 is configured with the auto-IP address 10.1.1.1 and the e0/1 interface on device R2 is configured with the auto-IP address 10.1.1.3.

To configure R1 as the seed device, 10.1.1.1 must be configured as the IP address of the interface e0/0. By configuring the IP address of e0/0 interface of R1 to its auto-IP address, R1 is configured as the seed device and the interface e0/0 becomes the owner of the subnet.

The process of configuring the device R1 as the seed device is given below:

After a connection is established between the devices R1 and R2, R1 sends a Link Layer Discovery Protocol(LLDP) packet which contains an auto-IP Type Length Value (TLV) with priority 2.

The auto-IP information for the e0/0 interface on R1 is given below:

Interface IP address	Auto-IP address	Priority
10.1.1.1	10.1.1.1	2

On receiving the auto-IP TLV from R1, R2 derives the IP address for the interface e0/1 (by subtracting 1 from the last octet of R1's auto-IP address), and assigns the IP address 10.1.1.0/31 to R2's e0/1 interface. The interface e0/1 on R2 becomes the non-owner interface on this subnet.

The IP address allocation is displayed in the illustration given below:

The device and node interface details for the subnet are given below:

Device	Interface	IP address	Designation
R1	e0/0	10.1.1.1/31	Owner
R2	e0/1	10.1.1.0/31	Non-owner

Note Since the auto-IP address configured on the e0/1 interface on R2 is 10.1.1.3, the other node interface of R2 is designated as the owner interface and 10.1.1.3 is automatically configured as the interface IP address of the other node interface.

Auto-IP Configuration for Inserting a Device into an Auto-IP Ring

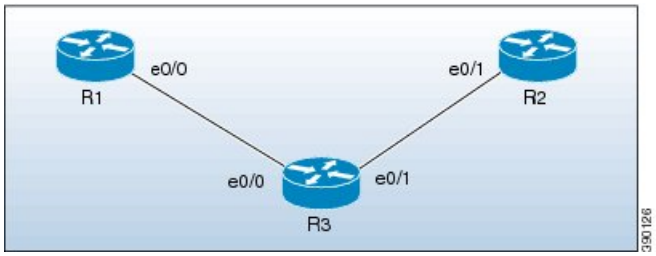
To insert a device into an existing auto-IP ring, the node interfaces of the device must be configured with the auto-IP address.

Note You can also configure the auto-IP feature on node interfaces that are part of an existing, but non-auto-IP ring.

The topology in the illustration below shows a sample scenario.

Device R1 is configured as the seed device. Interface e0/0 on R1 is configured with the IP address 10.1.1.1/31, and is the owner of the subnet connecting R1 and R2. Interface e0/1 on device R2 has the IP address 10.1.1.0/31, and is the non-owner interface of the subnet.

Device R3 is inserted between R1 and R2. The two designated node interfaces e0/0 and e0/1 of R3 are configured with the auto-IP address 10.1.1.5. After insertion of the device, the ring topology appears as shown in the illustration below:



Auto-IP TLV exchange between the devices R1 and R3 is given below:

1. R1 sends an auto-IP Type Length Value (TLV) with priority 2 to the e0/0 interface of R3.
2. After receiving the auto-IP TLV from R1, R3 sends an auto-IP TLV with priority 0 to the e0/0 interface of R1.
3. R1 wins the election process and the interface e0/0 of R1 is designated as the owner interface on the subnet connecting R1 and R3.
4. The e0/0 interface on R3 becomes the non-owner interface and the IP address 10.1.1.0 is assigned to it.
5. The other node interface on R3 is designated as an owner interface and its auto-IP address (10.1.1.5) is assigned as the IP address of the interface.

Auto-IP TLV exchange between the devices R3 and R2 is given below:

1. R3 sends an auto-IP TLV with priority 2 to the e0/1 interface of R2.
2. After receiving the auto-IP TLV from R3, R2 sends an auto-IP TLV with priority 0 to the e0/1 interface of R3.
3. R3 wins the election process and its interface e0/1 is designated as the owner interface on the subnet connecting R3 and R2.
4. The e0/1 interface on R2 is designated as the non-owner interface, and the IP address 10.1.1.4 is assigned to it.

5. The other node interface on R2 is designated as the owner interface and its auto-IP address is assigned as the IP address.

The IP addresses that are configured for the owner and non-owner interfaces on the devices R1, R2, and R3 are given below:

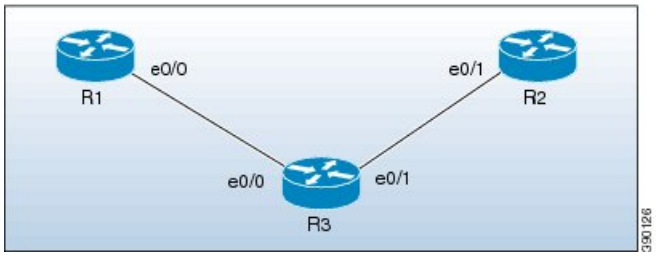
Device	Interface	IP Address	Designation
R1	e0/0	10.1.1.1/31	Owner
R3	e0/0	10.1.1.0/31	Non-owner
R3	e0/1	10.1.1.5/31	Owner
R2	e0/1	10.1.1.4/31	Non-owner

Device Removal from an Auto-IP Ring

You can manually remove a device from an existing auto-IP ring.

Note No configuration is required if you remove a device from an auto-IP ring and connect its neighbor devices.

The topology in the illustration below shows a sample scenario:



In the topology, device R3 is removed from the auto-IP ring and device R1 is connected to R2. As a result, auto-IP Type Length Value (TLVs) are exchanged between R1 and R2. Since the e0/0 interface of R1 sends an auto-IP TLV with priority 2 and the e0/1 interface of R2 sends an auto-IP TLV with priority 0 to the e0/0 interface on R1, the e0/0 interface of R1 is designated as the owner interface on the subnet connecting R1 and R2. R1 assigns the IP address to the e0/1 interface on R2, and it becomes the non-owner interface on this subnet.

After the removal of R3 from the auto-IP ring, the ring topology looks like this:

The IP address of the owner and non-owner interfaces on the subnet are given below:

Device	Interface	Designation
R1	e0/0	Owner

R2	e0/1	Non-owner

Conflict Resolution Using the Auto-Swap Technique

The auto-swap technique automatically resolves conflicts due to incorrect insertion of a device into an auto-IP ring.

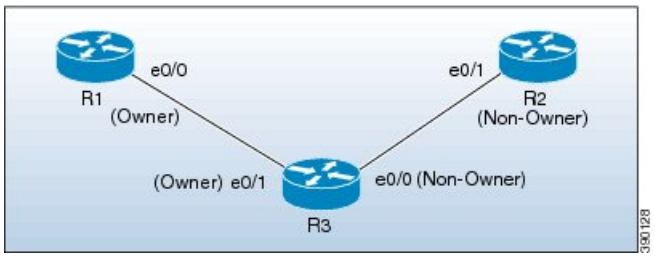
If you remove a device from an auto-IP ring, the owner and non-owner auto-IP configuration on the node interfaces is retained. You can insert the device back into an auto-IP ring.

If you incorrectly insert a device into a ring with its interfaces swapped (due to which two owner interfaces and two non-owner interfaces are connected to each other, rather than a connection between an owner and a non-owner interface), then identical priority values are exchanged between interfaces during the auto-IP Type Length Value (TLV) transmission. This leads to a tie in the priority value that is exchanged between the node interfaces of the inserted device, and a conflict is detected.

The auto-swap technique resolves conflicts on both the node interfaces of the inserted device and allows allocation of IP addresses for the interfaces.

Note No configuration is required to enable the auto-swap technique; it is enabled automatically. The auto-swap technique is used only when conflict is detected on both the node interfaces of the device.

The topology in the illustration below shows a sample scenario:



In this topology, device R3 is incorrectly inserted between the devices R1 and R2, with its interfaces swapped. The conflict arises due to incorrect insertion, as given below:

- An owner interface is connected to another owner interface; the e0/0 interface of R1 is connected to the e0/1 interface of R3.
- A non-owner interface is connected to another non-owner interface; the e0/1 interface of R2 is connected to the e0/0 interface of R3.

The auto-IP TLV exchange details between R1 and R3 are given below:

- The e0/0 interface on R1 sends an auto-IP TLV with priority 2 to the e0/1 interface on R3.
- The e0/1 interface on R3 sends an auto-IP TLV with priority 2 to the e0/0 interface on R1.

Since the same priority value of 2 is sent in both instances, there is a tie during the election process, leading to a conflict.

Similarly, the same priority value of 0 is exchanged between the e0/0 interface of R3 and the e0/1 interface of R2 since they are non-owner interfaces, leading to a conflict.

Auto Swap

The auto-IP feature uses the auto-swap technique to resolve conflicts on both the node interfaces of the inserted device.

The priority and the interface IP address of the e0/1 interface on R3 is swapped with the priority and the interface IP address of the e0/0 interface on R3, respectively.

After swapping, the following auto-IP TLV information is exchanged between R1 and R3:

- The e0/0 interface on R1 sends an auto-IP TLV with priority 2 to the e0/1 interface on R3.
- The e0/1 interface on R3 sends an auto-IP TLV with priority 0 to the e0/0 interface on R1.

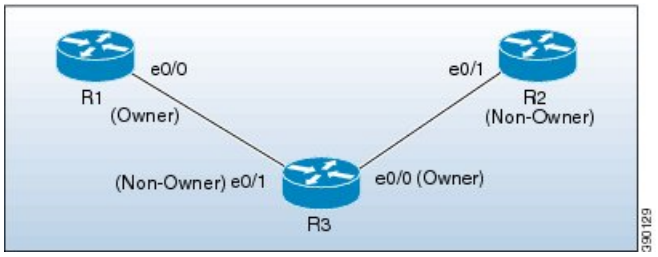
Since the priority sent by R1 to R3 is higher than the priority sent by the interface e0/1 on R3, R3 derives the IP address 10.1.1.0 for the e0/1 interface from the auto-IP address of R1 (10.1.1.1).

The following auto-IP TLV information is exchanged between R3 and R2:

- The e0/0 interface on R3 sends an auto-IP TLV with priority 2 to the e0/1 interface on R2.
- The e0/1 interface on R2 sends an auto-IP TLV with priority 0 to the e0/1 interface on R3.

R2 detects the priority sent by R3 to be higher than the priority sent by its interface e0/1 and derives the IP address 10.1.1.4 from the auto-IP address of R3 (10.1.1.5).

After conflict resolution, the topology looks like this:



The e0/1 interface on R3 is designated as a non-owner interface and the e0/0 interface on R3 is designated as the owner interface.

How to Configure Auto-IP

Configuring a Seed Device

You must configure at least one seed device in an auto-IP ring. To configure a seed device, you must configure the auto-IP address on the two node interfaces of the device (for a specific ring), and use the same IP address to configure the IP address on one of the two node interfaces.

Attention Understand these concepts before configuring auto-IP on virtual routing and forwarding instance (VRF) interfaces, Switch Virtual Interfaces (SVIs), and EtherChannels:

- VRF—If you intend to enable auto-IP on a VRF interface, ensure that the node interface is presently within the VRF. If the interface is not within a VRF presently, assign the interface to the VRF and then configure auto-IP on the VRF interface. Ensure that both node interfaces for the ring are assigned to the same VRF.
- SVI—Auto-IP configuration on an SVI is possible only if a single physical interface is associated with an SVI and the physical interface is an access port.
- EtherChannels—You can configure auto-IP on an EtherChannel interface, but not on a member interface of the EtherChannel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **lldp run**
4. **interface** *type number*
5. **auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address*
6. **exit**
7. **interface** *type number*
8. **auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address*
9. **ip address** *interface-ip-address subnet-mask*
10. **end**
11. **show auto-ip-ring** [*ring-id*][**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**lldp run** **Example:** `Device(config)# lldp run`	Enables Link Layer Discovery Protocol (LLDP) for the device.
Step 4	**interface** *type number* **Example:**	Specifies an interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface ethernet 0/0`	
Step 5	**auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address* **Example:** `Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.1`	Configures the auto-IP address on the specified interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Specifies an interface type and number, and enters interface configuration mode.
Step 8	**auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address* **Example:** `Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.1`	Configures the auto-IP address on the specified interface.
Step 9	**ip address** *interface-ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.254`	Configures the IP address on the specified interface. **Note** The specified interface is designated as the owner interface of the seed device.
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 11	**show auto-ip-ring** [*ring-id*][**detail**] **Example:** `Device# show auto-ip-ring 4 detail`	Displays auto-IP information.

Configuring the Auto-IP Functionality on Node Interfaces (for Inclusion in an Auto-IP Ring)

To insert a device into an auto-IP ring or to enable node interfaces in an existing ring, you must configure the auto-IP address on the 2 designated node interfaces of the device.

Attention Understand these concepts before configuring auto-IP on virtual routing and forwarding instance (VRF) interfaces, Switch Virtual Interfaces (SVIs), and EtherChannels:

- VRF—If you intend to enable auto-IP on a VRF interface, ensure that the node interface is presently within the VRF. If the interface is not within a VRF presently and you want the interface to be within a VRF, move the interface within the VRF and then configure auto-IP on the VRF interface. Ensure that both node interfaces are within the same VRF.
- SVI—Auto-IP configuration on an SVI is possible only if a single physical interface is associated with an SVI and the physical interface is an access port.
- EtherChannels—You can configure auto-IP on an EtherChannel interface, but not on a member interface of the EtherChannel.

This task is applicable for a non-seed device in an auto-IP ring. Ensure that a seed device is configured for the auto-IP ring before performing this task.

Perform the steps given below to configure the auto-IP functionality on the two node interfaces of a device:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **lldp run**
4. **interface** *type number*
5. **auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address*
6. **exit**
7. **interface** *type number*
8. **auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address*
9. **end**
10. **show auto-ip-ring** [*ring-id*][**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**lldp run** **Example:** `Device(config)# lldp run`	Enables Link Layer Discovery Protocol (LLDP) for the device.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Specifies an interface type and number, and enters interface configuration mode.
Step 5	**auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address* **Example:** `Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.3`	Configures the auto-IP address on the specified interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface ethernet 1/1`	Specifies an interface type and number, and enters interface configuration mode.
Step 8	**auto-ip-ring** *ring-id* **ipv4-address** *auto-ip-address* **Example:** `Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.3`	Configures the auto-IP address on the specified interface.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 10	**show auto-ip-ring** [*ring-id*][**detail**] **Example:** `Device# show auto-ip-ring 4 detail`	Displays auto-IP information.

Verifying and Troubleshooting Auto-IP

Perform this task to verify auto-IP functions.

Note The commands are not in any specific order. The **show auto-ip-ring** command is presented twice. One of the examples displays auto-IP ring information for virtual routing and forwarding instance (VRF) interfaces, and the other example displays auto-IP ring information for non-VRF interfaces.

SUMMARY STEPS

1. **enable**
2. **show auto-ip-ring** [*ring-id*][**detail**]
3. **show auto-ip-ring** [*ring-id*][**detail**]
4. **debug auto-ip-ring** {*ring-id* {**errors** | **events**} |**errors** | **events**}

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 show auto-ip-ring [*ring-id*][**detail**]

This command displays auto-IP ring information for a specific device or auto-IP ring.

Example:

```
Device# show auto-ip-ring

Auto-IP ring 1
Auto-IP Address         : 10.1.1.5

Ring Port0              : Ethernet0/0
My Current-IP           : 0.0.0.0
My Priority             : 1

Auto-IP ring 3
Auto-IP Address         : 10.1.1.3

Ring Port0              : Ethernet0/1
My Current-IP           : 0.0.0.0
My Priority             : 1
```

Step 3 show auto-ip-ring [*ring-id*][**detail**]

This command displays auto-IP ring information for VRF interfaces.

Example:

```
Device# show auto-ip-ring detail

Auto-IP ring 7
Auto-IP Address         : 10.1.1.11

VRF Name                : 3
Ring Port1              : Ethernet1/1
My Current-IP           : 10.1.1.11
My Priority             : 2

Rx Auto-IP Address      : 10.1.1.13
Rx Current-IP           : 10.1.1.10
```

```
Rx Priority             : 0


VRF Name                : 3
Ring Port0              : Ethernet1/0
My Current-IP           : 10.1.1.8
My Priority             : 0

Rx Auto-IP Address      : 10.1.1.9
Rx Current-IP           : 10.1.1.9
Rx Priority             : 2
```

Step 4 **debug auto-ip-ring** {*ring-id* {**errors** | **events**} |**errors** | **events**}

This command debugs errors and events for the specified auto-IP ring.

Example:

```
Device# debug auto-ip-ring 1 errors

Auto IP Ring errors debugging is on for the ring id : 1
*Jul 26 11:30:40.541: (Ethernet0/0) priority (value:1) conflict detected,  need admin intervention
```

Note A conflict is detected in the above debug example because the priority in the auto-IP Type Length Value (TLV) that is sent from the interface and the priority that is received from the neighbor interface is the same.

Configuration Examples for Auto-IP

Example: Configuring a Seed Device

```
Device> enable
Device# configure terminal
Device(config)# lldp run
Device(config)# interface ethernet 0/0
Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.1
Device(config-if)# exit
Device(config)# interface ethernet 1/0
Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.1
Device(config-if)# ip address 10.1.1.1 255.255.255.254
Device(config-if)# end
```

Example: Configuring the Auto-IP Functionality on Node Interfaces (for Inclusion in an Auto-IP Ring)

```
Device> enable
Device# configure terminal
Device(config)# lldp run
```

```
Device(config)# interface ethernet 0/1
Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.3
Device(config-if)# exit
Device(config)# interface ethernet 1/1
Device(config-if)# auto-ip-ring 4 ipv4-address 10.1.1.3
Device(config-if)# end
```

Additional References for Auto-IP

Related Documents

Related Topic	Document Title
Configuring IPv4 Addresses	IP Addressing: IPv4 Addressing Configuration Guide
Using Link Layer Discovery Protocol in Multivendor Networks	Carrier Ethernet Configuration Guide
IPv4 Addressing commands	Cisco IOS IP Addressing Services Command Reference
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

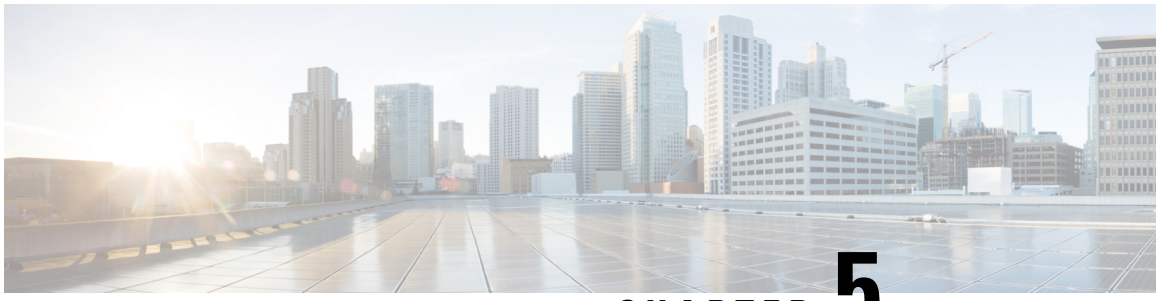
Feature Information for Auto-IP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13: Feature Information for Auto-IP

Feature Name	Releases	Feature Information
Auto-IP		The auto-IP feature addresses the problem of manually reconfiguring nodes during insertion, deletion, and movement of nodes within an auto-IP ring. The auto-IP feature automatically provides IP addresses to the node interfaces inserted into an auto-IP ring. The following commands were introduced or modified: **auto-ip-ring**, **debug auto-ip-ring**, **show auto-ip-ring**.
		The following commands were introduced or modified: **show auto-ip-ring**.



CHAPTER 5

Zero Touch Auto-IP

The Zero touch Auto-IP feature enables automatic allocation and configuration of IP addresses for nodes in a ring topology. The IP addresses are allocated from a pool of IP addresses that is predefined by you.

The advantages of Zero Touch Auto-IP over Auto-IP are:

- IP addresses can be configured automatically on ring nodes. Manual IP address configuration is not required on each node.
- IP addresses are allocated from a common IP address pool, and the IP address range can be predefined by you.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Zero Touch Auto-IP

- Link Layer Discovery Protocol (LLDP) must be enabled on all the Auto-IP ring device ports.
- In an Auto-IP ring, you must identify one Auto-IP device as an Auto-IP server.

- None of the ports identified to be part of the Zero Touch Auto-IP ring should be manually configured with the Auto-IP functionality. If a port that is identified for Zero touch Auto-IP configuration has a manual Auto-IP configuration, disable the manual Auto-IP configuration on that port.

Restrictions for Zero Touch Auto-IP

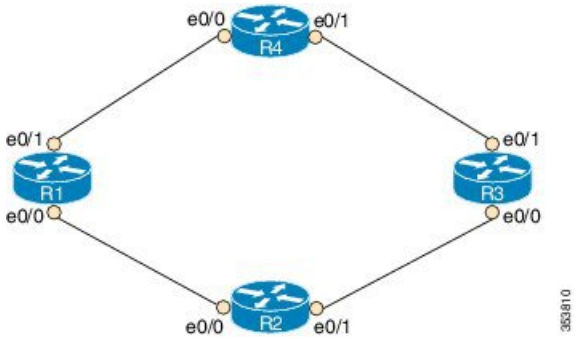
- Zero Touch Auto-IP and Auto-IP cannot coexist. To implement Zero Touch Auto-IP functionality, all the ports of the Auto-IP ring have to be configured as Zero Touch Auto-IP ports.
- Zero Touch Auto-IP works if the designated Auto-IP server is in an autonomic network.

Information About Zero Touch Auto-IP

The Zero Touch Auto-IP feature uses Autonomic Networking and Link Layer Discovery Protocol (LLDP) to achieve the objective of automatic IP address configuration on nodes in a ring network.

Consider the topology for Zero Touch Auto-IP configuration. The devices R1, R2, R3 and R4 are connected in a ring network and LLDP is enabled on all the ring ports.

Figure 7: Zero Touch Auto-IP Topology



To know about and configure the Zero touch Auto-IP functionality, use the information given below:

1. Associate one device in the ring network (say R1) to the autonomic network. Enable autonomic status for the other Auto-IP devices. For more information on autonomic networks, refer Autonomic Networking.

```
R1(config)# autonomic registrar
R1(config-registrar)# domain-id auto-addressing.com
R1(config-registrar)# no shutdown
R1(config-registrar)# CA local
R1(config-registrar)# exit
R1(config)# autonomic
```

```
R2(config)# autonomic
R3(config)# autonomic
R4(config)# autonomic
```

Note that R1 is configured on the registrar and receives a certificate. The remaining devices are configured as autonomic devices.

2. Enable the *auto* mode on all the ports in the ring to enable automatic IP address configuration. Auto mode must be enabled on the e0/0 and e0/1 ports on R1, R2, R3 and R4. For ports of the same device, the ring ID must be identical.

```
Device(config-if)# auto-ip-ring 1 ipv4-auto
```

3. Configure the device added to the autonomic network (R1) as the Auto-IP server. The server stores a pool of IP addresses.

```
R1(config)# auto-ip-ring server
```

4. Reserve a pool of IP addresses on the Auto-IP server for IP address allocation to the ring ports.

Note In Zero Touch and manual Auto-IP configuration, a /31 subnet is created for a pair of owner and nonowner ports (each device will have a owner and non owner port). An odd-numbered IP address (such as 10.1.1.11) is issued to an owner port and an even-numbered IP address (10.1.1.10) is reserved for a nonowner port. Therefore, specify the first IP address in the range along with the number of devices (or /31 subnets) that make up the Auto-IP ring

```
R1(config-auto-ip-server)# ipv4-address-pool 10.1.1.10 6
```

Result—A range of IP addresses from 10.1.1.10 to 10.1.1.21 is allocated for the Auto-IP ring. The Auto-IP server is added to the autonomic network and is reachable by other nodes in the autonomic network.

Note IP addresses for six devices will be reserved (though the requirement is for four devices); the additional IP addresses will be allocated when you add new devices to the ring.

5. Auto-IP negotiation process— IP addresses are allocated to the Auto-IP ring nodes through a negotiation process. To initiate the process, configure one port as the seed port in the Auto-IP ring.

```
R1(config-if)# auto-ip-ring 1 ipv4-seed
```

The negotiation process is explained below:

   a. The priority of the seed port (a port on R1, for example) is set to 2 and it is made an owner port. An IP address from the reserved pool is configured on the port.

   b. The seed port advertises its priority (2) to its connected neighbor, and makes the neighbor port a non owner. The seed port assigns an IP address to the neighbor port and the neighbor port's priority is changed to 0.

   c. Each owner port in the ring gets an IP address from the Auto IP server. The owner port, in turn, assigns an IP address to the connected neighbor port.

6. Auto-IP communication—After initial configuration, each owner port sends periodic messages to the Auto-IP server to continue preserving its IP address. If there is no message from the owner port to the Auto-IP server for 15 minutes, the server moves the IP address to the pool of free IP addresses.

The following are some points to keep in mind in the context of Zero Touch Auto-IP configuration:

- LLDP has to be enabled on all the Auto-IP ring ports before Auto-IP configuration.
- Before you insert a new interface into the ring, configure auto mode on the ring ports.
- For Zero Touch Auto-IP configuration, the number of devices (or /31 subnets) that make up the Auto-IP ring must be between 1 and 128.
- When you specify a pool of IP addresses, ensure that IP addresses in the specified range are not already in use.
- Ensure that you reserve some additional IP addresses for the Auto-IP ring, in case more devices are added to the ring topology at a later point in time.
- The starting IP address used for the Auto-IP address pool reservation must be an even number. For example, 10.1.1.10 is a valid IP address but 10.1.1.9 is not.
- If you remove a device from an Auto-IP ring, the Auto-IP addresses are released back to the Auto-IP server.

How to Configure Zero Touch Auto-IP

Associating an Auto-IP Server with an Autonomic Network

The Auto-IP server (R1) must be associated with the autonomic network, and configured in the Autonomic Network registrar. The other devices in the network (R2, R3, and R4) must be enabled with the autonomic status.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **autonomic registrar**
4. **domain-id auto-addressing.com**
5. **no shutdown**
6. **CA local**
7. **exit**
8. **autonomic**
9. **autonomic**
10. **autonomic**
11. **autonomic**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `R1> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `R1# configure terminal`	Enters global configuration mode.
Step 3	**autonomic registrar** **Example:** `R1(config)# autonomic registrar`	Enables the Auto-IP server in the Autonomic Network registrar and enters registrar configuration mode.
Step 4	**domain-id auto-addressing.com** **Example:** `R1(config-registrar)# domain-id auto-addressing.com`	Represents a common group of all devices registering with the registrar. **Note** If R1 is configured on the AN registrar, then R1 represents the Auto-IP ring devices R2, R3, and R4.
Step 5	**no shutdown** **Example:** `R1(config-registrar)# no shutdown`	Enables the autonomic registrar.
Step 6	**CA local** **Example:** `R1(config-registrar)# CA local`	Issues a Local CA certificate to the Auto-IP server.
Step 7	**exit** **Example:** `R1(config-registrar)# exit`	Exits registrar configuration mode and enters global configuration mode.
Step 8	**autonomic** **Example:** `R1(config)# autonomic`	Configures the Auto-IP server as an autonomic device. **Note** You should associate the remaining devices (R2, R3, and R4) in the Auto-IP ring with the autonomic network, as given in the next few steps.
Step 9	**autonomic** **Example:** `R2(config)# autonomic`	Configures R2 as an autonomic device.
Step 10	**autonomic** **Example:** `R3(config)# autonomic`	Configures R3 as an autonomic device.

	Command or Action	Purpose
Step 11	**autonomic** **Example:** `R4(config)# autonomic`	Configures R4 as an autonomic device.
Step 12	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

What to do next

Enable *auto* mode on Auto-IP ring ports

Enabling Auto Mode on Auto-IP Ring Ports

Before you begin

Identify the ports that will be part of the Auto-IP ring. Remember that you must enable Auto mode on all the ports in an Auto-IP ring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **lldp run**
4. **interface** *type number*
5. **auto-ip-ring** *ring-id* **ipv4-auto**
6. **exit**
7. Repeat steps to configure auto mode on each Auto-IP ring port.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**lldp run** **Example:** `Device(config)# lldp run`	Enables Link Layer Discovery Protocol (LLDP) for the device.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 5	**auto-ip-ring** *ring-id* **ipv4-auto** **Example:** `Device(config-if)# auto-ip-ring 1 ipv4-auto`	Configures auto mode on the Auto-IP ring port.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	Repeat steps to configure auto mode on each Auto-IP ring port.	---

What to do next

Configure an Auto-IP server and reserve a pool of IP addresses for the Auto-IP ring ports.

Configuring an Auto-IP Server and Reserving a Pool of IP Addresses on the Server

Before you begin

Ensure that all ports of the ring are identified and auto mode is enabled on the ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **auto-ip-ring server**
4. **ipv4-address-pool** *auto-ipv4-address number-of-subnets*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**auto-ip-ring server** **Example:** `Device(config)# auto-ip-ring server`	Configures the device as an Auto-IP server and enters Auto-IP server configuration mode.
Step 4	**ipv4-address-pool** *auto-ipv4-address number-of-subnets* **Example:** `Device(config-auto-ip-server)# ipv4-address-pool 10.1.1.10 6`	Reserves a pool of IP addresses on the Auto-IP server. The number of subnets should, at a minimum, be the total number of owner ports or devices in the ring. The odd-numbered IP addresses are assigned to the owner ports, and each non owner port fetches its IP address from the owner port through LLDP
Step 5	**exit** **Example:** `Device(config-auto-ip-server)# exit`	Exits Auto-IP server configuration mode and enters global configuration mode.

What to do next

Configure a seed port to start the Auto-IP negotiation process.

Configuring a Seed Port

Before you begin

Ensure all the Auto-IP ports are in auto mode, and a pool of IP addresses is reserved for the Auto-IP ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **auto-ip-ring** *ring-id* **ipv4-seed**
5. **exit**
6. **end**
7. **show auto-ip-ring** [*ring-id*][**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**auto-ip-ring** *ring-id* **ipv4-seed** **Example:** `Device(config-if)# auto-ip-ring 1 ipv4-seed`	Designates the port as the seed port and initiates the Auto-IP negotiation process.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show auto-ip-ring** [*ring-id*][**detail**] **Example:** `Device# show auto-ip-ring 4 detail`	Displays auto-IP information.

What to do next

Verify if the IP addresses have been configured.

Verifying and Troubleshooting Zero Touch Auto-IP

Perform this task to verify Zero touch Auto-IP functions.

Note The commands are not in any specific order.

SUMMARY STEPS

1. **enable**
2. **show auto-ip-ring** [*ring-id*][**detail**]
3. **show autonomic service**
4. **show autonomic device**
5. **show autonomic neighbors**
6. **debug auto-ip-ring** {*ring-id* {**errors** | **events**} |**errors** | **events**}

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 show auto-ip-ring [*ring-id*][**detail**]

This command displays Auto-IP ring information for a specific device or Auto-IP ring. The sample output given below displays two ports representing a ring, their IP addresses, and the connected ports and IP addresses (neighboring port information is denoted by Rx).

Example:

```
Device# show auto-ip-ring 1

Auto-IP ring 1

 Auto-IP Address          : 10.1.1.11

 Ring Port0               : Ethernet0/1
 My Current-IP            : 10.1.1.11
 My Priority              : 2

 Rx Auto-IP Address       : 10.1.1.13
 Rx Current-IP            : 10.1.1.12
 Rx-Priority              : 0

 Ring Port1               : Ethernet0/0
 My Current-IP            : 10.1.1.10
 My Priority              : 0

 Rx Auto-IP Address       : 10.1.1.17
 Rx Current-IP            : 10.1.1.17
 Rx-Priority              : 2
```

Step 3 **show autonomic service**

The following is sample output from this command, and it displays autonomic services configured on a device connected to an autonomic network.

Example:

```
Device# show autonomic service

Service                        IP-Addr
Autonomic registrar            FD53:EE55:A541:0:AABB:CC00:100:1
ANR type                       IOS CA
Auto IP Server                 FD53:EE55:A541:0:AABB:CC00:100:1
```

Step 4 **show autonomic device**

The following is sample output from this command, and it displays autonomic network configuration credentials for a device that is connected to the autonomic network. Details like unique identifier (UDI), device identifier (Device ID), associated domain (Domain ID), and so on, are displayed.

Example:

```
Device# show autonomic device

        UDI                            PID:Unix SN:655773698
        Device ID                      aabb.cc00.0100-2
        Domain ID                      auto-networking.com
        Domain Certificate             (sub:) ou=abcd.com+serialNumber=PID:Unix
SN:655773698,cn=aabb.cc00.0100-2
        Certificate Serial Number      03
        Device Address                 FD53:EE55:A541:0:AABB:CC00:100:2
        Domain Cert is Valid
```

Step 5 **show autonomic neighbors**

The following is sample output from this command, and it displays autonomic configuration details of connected, neighbor devices. Details such as unique identifier (UDI), device identifier (Device ID), and associated domain (Domain ID), are displayed.

Example:

```
Device# show autonomic neighbors

        UDI                                          Device-ID             Domain     Interface
------------------------------------------------------------------------------------------
PID:Unix SN:655773697                          aabb.cc00.0100-1      abcd.com
Ethernet0/0
PID:Unix SN:655773699                          aabb.cc00.0100-4      abcd.com   Ethernet0/1
```

Step 6 **debug auto-ip-ring** {*ring-id* {**errors** | **events**} |**errors** | **events**}

The following is sample output from this command, and it displays debug errors and events for the specified Auto-IP ring.

Note A conflict is detected in the sample debug output below because the priority in the Auto-IP Type Length Value (TLV) that is sent from the interface and the priority that is received from the neighbor interface are the same.

Example:

```
Device# debug auto-ip-ring 2 errors

Auto IP Ring errors debugging is on for the ring id : 2
*Jul 26 11:30:40.541: (Ethernet0/0) priority (value:1) conflict detected,  need admin intervention
```

Configuration Examples for Zero Touch Auto-IP

Example: Associating an Auto-IP Server with an Autonomic Network

Auto-IP server (R1) is associated with the autonomic network. The other devices in the network (R2, R3, and R4) are enabled with the autonomic status.

```
R1(config)# autonomic registrar
R1(config-registrar)# domain-id auto-addressing.com
R1(config-registrar)# no shutdown
R1(config-registrar)# CA local
R1(config-registrar)# exit
R1(config)# autonomic


R2(config)# autonomic
R3(config)# autonomic
R4(config)# autonomic
```

Example: Enabling Auto Mode on Auto-IP Ring Ports

```
Device> enable
Device# configure terminal
Device(config)# lldp run
Device(config)# interface ethernet 0/0
Device(config-if)# auto-ip-ring 1 ipv4-auto
Device(config-if)# exit

Repeat the preceding steps to configure the auto mode on each Auto-IP ring port
```

Example: Configuring an Auto-IP Server and Reserving a Pool of IP Addresses on the Server

```
Device> enable
Device# configure terminal
Device(config)# auto-ip-ring server
Device(config-auto-ip-server)# ipv4-address-pool 10.1.1.10 6
Device(config-auto-ip-server)# exit
```

Example: Configuring a Seed Port

```
Device> enable
Device# configure terminal
Device(config)# interface e0/0
Device(config-if)# auto-ip-ring 1 ipv4-seed
Device(config-if)# exit
```

Additional References for Zero Touch Auto-IP

Related Documents

Related Topic	Document Title
Auto-IP	IP Addressing: IPv4 Addressing Configuration Guide
Configuring IPv4 Addresses	IP Addressing: IPv4 Addressing Configuration Guide
Using Link Layer Discovery Protocol in Multivendor Networks	Carrier Ethernet Configuration Guide
IPv4 Addressing commands	Cisco IOS IP Addressing Services Command Reference
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

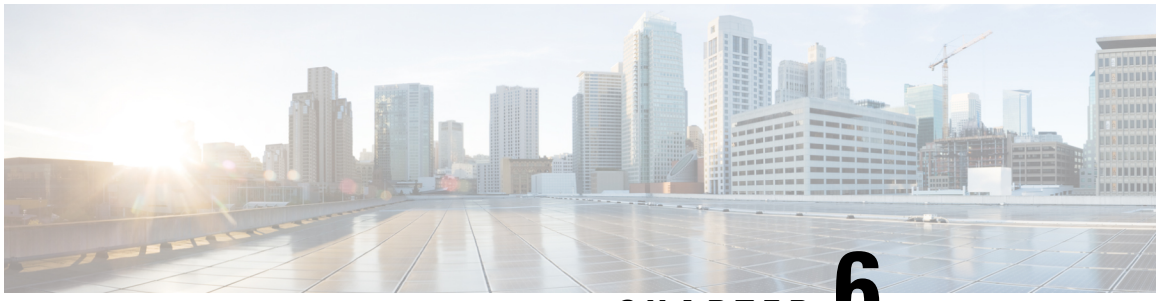
Feature Information for Auto-IP

Table 14: Feature Information for Auto-IP

Feature Name	Releases	Feature Information
Zero Touch Auto-IP		The Zero Touch Auto-IP feature enables automatic allocation and configuration of IP addresses for nodes in an Auto-IP ring. The IP addresses are allocated from a pool of IP addresses. The following commands were introduced or modified: **auto-ip-ring ipv4-auto**, **auto-ip-ring ipv4-seed**, **auto-ip-ring server**, **ipv4-address-pool**.

PART II

IPv6 Addressing



CHAPTER 6

IPv6 Addressing and Basic Connectivity

Internet Protocol version 6 (IPv6) expands the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides more than enough globally unique IP addresses for every networked device on the planet. The unlimited address space provided by IPv6 allows Cisco to deliver more and newer applications and services with reliability, improved user experience, and increased security.

Implementing basic IPv6 connectivity in the Cisco software consists of assigning IPv6 addresses to individual device interfaces. IPv6 traffic forwarding can be enabled globally, and Cisco Express Forwarding switching for IPv6 can also be enabled. The user can enhance basic connectivity functionality by configuring support for AAAA record types in the Domain Name System (DNS) name-to-address and address-to-name lookup processes, and by managing IPv6 neighbor discovery.

Restrictions for Implementing IPv6 Addressing and Basic Connectivity

- Multiple IPv6 global addresses within the same prefix can be configured on an interface; however, multiple IPv6 link-local addresses on an interface are not supported.
- IPv4 alias and IPv6 alias addresses used must be available in the global routing table and not under VRF.

Information About IPv6 Addressing and Basic Connectivity

IPv6 for Cisco Software

IPv6, formerly named IPng (next generation), is the latest version of the Internet Protocol (IP). IP is a packet-based protocol used to exchange data, voice, and video traffic over digital networks. IPv6 was proposed when it became clear that the 32-bit addressing scheme of IP version 4 (IPv4) was inadequate to meet the

demands of Internet growth. After extensive discussion it was decided to base IPng on IP but add a much larger address space and improvements such as a simplified main header and extension headers. IPv6 is described initially in RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*, issued by the Internet Engineering Task Force (IETF). Further RFCs describe the architecture and services supported by IPv6.

The architecture of IPv6 has been designed to allow existing IPv4 users to transition easily to IPv6 while providing services such as end-to-end security, quality of service (QoS), and globally unique addresses. The larger IPv6 address space allows networks to scale and provide global reachability. The simplified IPv6 packet header format handles packets more efficiently. IPv6 prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities provide an IPv6 addressing hierarchy that allows for more efficient routing. IPv6 supports widely deployed routing protocols such as Integrated Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF) for IPv6, and multiprotocol Border Gateway Protocol (BGP). Other available features include stateless autoconfiguration and an increased number of multicast addresses.

Large IPv6 Address Space for Unique Addresses

The primary motivation for IPv6 is the need to meet the demand for globally unique IP addresses. IPv6 quadruples the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides more than enough globally unique IP addresses for every networked device on the planet. By being globally unique, IPv6 addresses inherently enable global reachability and end-to-end security for networked devices, functionality that is crucial to the applications and services that are driving the demand for the addresses. Additionally, the flexibility of the IPv6 address space reduces the need for private addresses; therefore, IPv6 enables new application protocols that do not require special processing by border devices at the edge of networks.

IPv6 Address Formats

IPv6 addresses are represented as a series of 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x:x. Following are two examples of IPv6 addresses:

2001:DB8:7654:3210:FEDC:BA98:7654:3210

2001:DB8:0:0:8:800:200C:417A

IPv6 addresses commonly contain successive hexadecimal fields of zeros. Two colons (::) may be used to compress successive hexadecimal fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent successive hexadecimal fields of zeros). The table below lists compressed IPv6 address formats.

A double colon may be used as part of the *ipv6-address* argument when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.

Note Two colons (::) can be used only once in an IPv6 address to represent the longest successive hexadecimal fields of zeros. The hexadecimal letters in IPv6 addresses are not case-sensitive.

Table 15: Compressed IPv6 Address Formats

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	2001:0:0:0:DB8:800:200C:417A	2001::DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:0:101	FF01::101

IPv6 Address Type	Preferred Format	Compressed Format
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

The loopback address listed in the table above may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).

Note The IPv6 loopback address cannot be assigned to a physical interface. A packet that has the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 devices do not forward packets that have the IPv6 loopback address as their source or destination address.

The unspecified address listed in the table above indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.

Note The IPv6 unspecified address cannot be assigned to an interface. The unspecified IPv6 addresses must not be used as destination addresses in IPv6 packets or the IPv6 routing header.

An IPv6 address prefix, in the format *ipv6-prefix*/*prefix-length*, can be used to represent bit-wise contiguous blocks of the entire address space. The *ipv6-prefix* must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:DB8:8086:6502::/32 is a valid IPv6 prefix.

IPv6 Address Output Display

When IPv6 or IPv4 command output displays an IPv6 address, a long IPv6 address can overflow into neighboring fields, causing the output to be difficult to read. The output fields were designed to work with the longest possible IPv4 address, which has 15 characters; IPv6 addresses can be up to 39 characters long. The following scheme has been adopted in IPv4 and IPv6 commands to allow the appropriate length of IPv6 address to be displayed and move the following fields to the next line, if necessary. The fields that are moved are kept in alignment with the header row.

The following example displays eight connections. The first six connections feature IPv6 addresses; the last two connections feature IPv4 addresses.

```
Device# where
Conn Host                Address             Byte  Idle Conn Name
   1 test5               2001:DB8:3333:4::5    6    24 test5
   2 test4               2001:DB8:3333:44::5
                                               6    24 test4
   3 2001:DB8:3333:4::5 2001:DB8:3333:4::5     6    24 2001:DB8:3333:4::5
   4 2001:DB8:3333:44::5
                         2001:DB8:3333:44::5
                                               6    23 2001:DB8:3333:44::5
   5 2001:DB8:3000:4000:5000:6000:7000:8001
                         2001:DB8:3000:4000:5000:6000:7000:8001
                                               6    20 2001:DB8:3000:4000:5000:6000:
```

```
6 2001:DB8:1::1      2001:DB8:1::1          0     1 2001:DB8:1::1
7 10.1.9.1            10.1.9.1              0     0 10.1.9.1
8 10.222.111.222      10.222.111.222        0     0 10.222.111.222
```

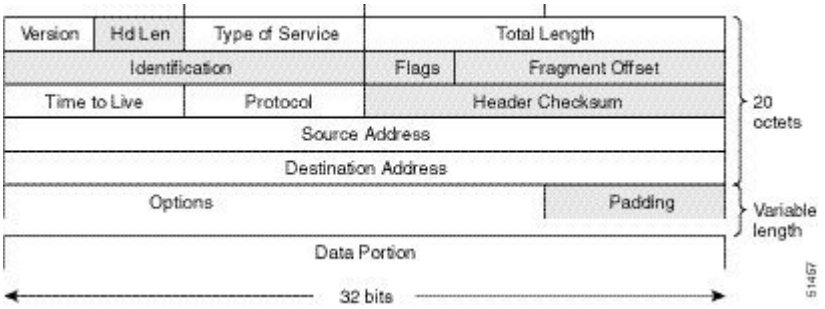
Connection 1 contains an IPv6 address that uses the maximum address length in the address field. Connection 2 shows the IPv6 address overflowing the address field and the following fields moved to the next line, but in alignment with the appropriate headers. Connection 3 contains an IPv6 address that fills the maximum length of the hostname and address fields without wrapping any lines. Connection 4 shows the effect of both the hostname and address fields containing a long IPv6 address. The output is shown over three lines keeping the correct heading alignment. Connection 5 displays a similar effect as connection 4 with a very long IPv6 address in the hostname and address fields. Note that the connection name field is actually truncated. Connection 6 displays a very short IPv6 address that does not require any change in the display. Connections 7 and 8 display short and long IPv4 addresses.

Note The IPv6 address output display applies to all commands that display IPv6 addresses.

Simplified IPv6 Packet Header

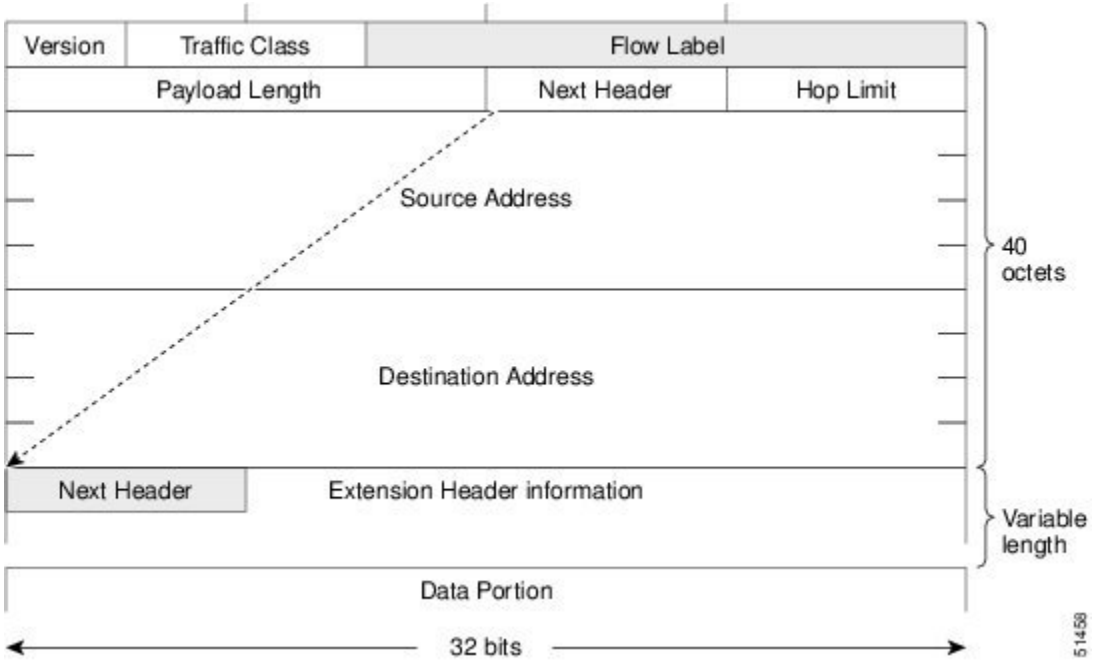
The basic IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see the figure below). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header shown in the figure below are not included in the IPv6 packet header.

Figure 8: IPv4 Packet Header Format



The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see the figure below). Fields were removed from the IPv6 header because, in IPv6, fragmentation is not handled by devices and checksums at the network layer are not used. Instead, fragmentation in IPv6 is handled by the source of a packet and checksums at the data link layer and transport layer are used. (In IPv4, the UDP transport layer uses an optional checksum. In IPv6, use of the UDP checksum is required to check the integrity of the inner packet.) Additionally, the basic IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

Figure 9: IPv6 Packet Header Format



The table below lists the fields in the basic IPv6 packet header.

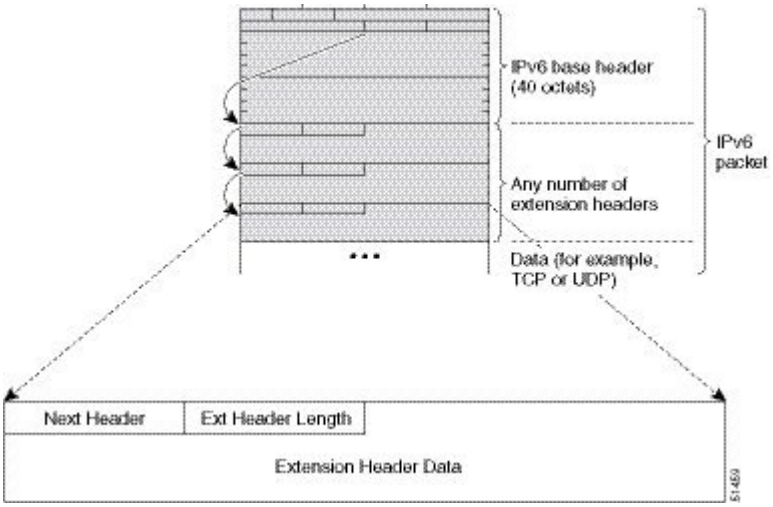
Table 16: Basic IPv6 Packet Header Fields

Field	Description
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information following the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in the figure immediately above.

Field	Description
Hop Limit	Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of devices that an IPv6 packet can pass through before the packet is considered invalid. Each device decrements the value by one. Because no checksum is in the IPv6 header, the device can decrement the value without needing to recalculate the checksum, which saves processing resources.
Source Address	Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Following the eight fields of the basic IPv6 packet header are optional extension headers and the data portion of the packet. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. The extension headers form a chain of headers. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. The figure below shows the IPv6 extension header format.

Figure 10: IPv6 Extension Header Format



The table below lists the extension header types and their Next Header field values.

Table 17: IPv6 Extension Header Types

Header Type	Next Header Value	Description
Hop-by-hop options header	0	This header is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the basic IPv6 packet header.

Header Type	Next Header Value	Description
Destination options header	60	The destination options header can follow any hop-by-hop options header, in which case the destination options header is processed at the final destination and also at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header, in which case the destination options header is processed only at the final destination.
Routing header	43	The routing header is used for source routing.
Fragment header	44	The fragment header is used when a source must fragment a packet that is larger than the maximum transmission unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.
Authentication header and ESP header	51 50	The Authentication header and the ESP header are used within IP Security Protocol (IPsec) to provide authentication, integrity, and confidentiality of a packet. These headers are identical for both IPv4 and IPv6.
Upper-layer headers	6 (TCP) 17 (UDP)	The upper-layer (transport) headers are the typical headers used inside a packet to transport the data. The two main transport protocols are TCP and UDP.
Mobility headers	135	Extension headers used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings.

DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses. IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

The table below lists the IPv6 DNS record types.

Table 18: IPv6 DNS Record Types

Record Type	Description	Format
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.)	www.abc.test AAAA 3FFE:YYYY:C18:1::2

Record Type	Description	Format
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.) **Note** Cisco software supports resolution of PTR records for the IP6.INT domain.	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

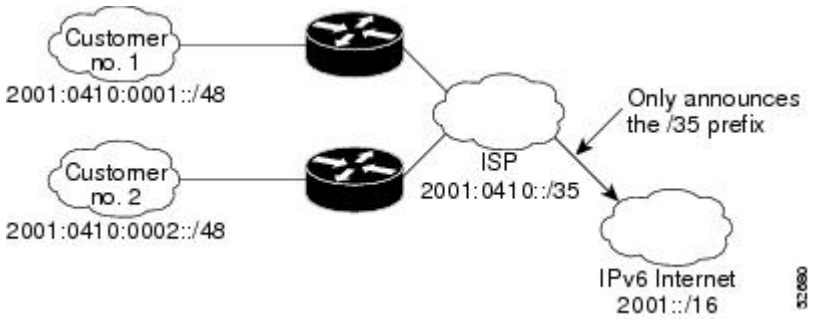
Cisco Discovery Protocol IPv6 Address Support

The Cisco Discovery Protocol IPv6 address support for neighbor information feature adds the ability to transfer IPv6 addressing information between two Cisco devices. Cisco Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

IPv6 Prefix Aggregation

The aggregatable nature of the IPv6 address space enables an IPv6 addressing hierarchy. For example, an enterprise can subdivide a single IPv6 prefix from a service provider into multiple, longer prefixes for use within its internal network. Conversely, a service provider can aggregate all of the prefixes of its customers into a single, shorter prefix that the service provider can then advertise over the IPv6 internet (see the figure below).

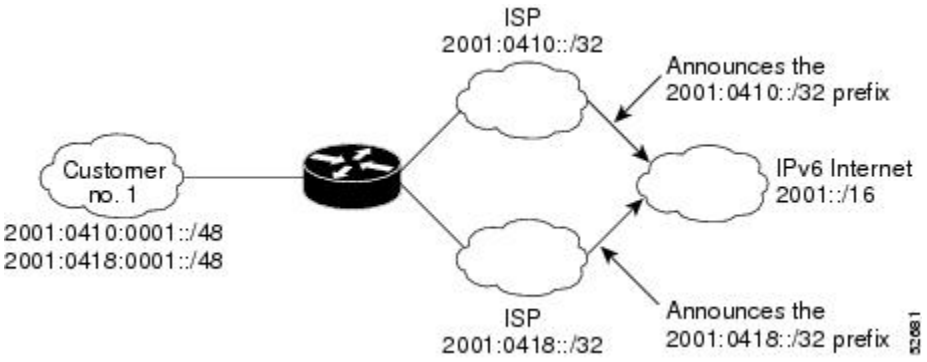
Figure 11: IPv6 Prefix Aggregation



IPv6 Site Multihoming

Multiple IPv6 prefixes can be assigned to networks and hosts. Having multiple prefixes assigned to a network allows that network to connect easily to multiple ISPs without breaking the global routing table (see the figure below).

Figure 12: IPv6 Site Multihoming



IPv6 Data Links

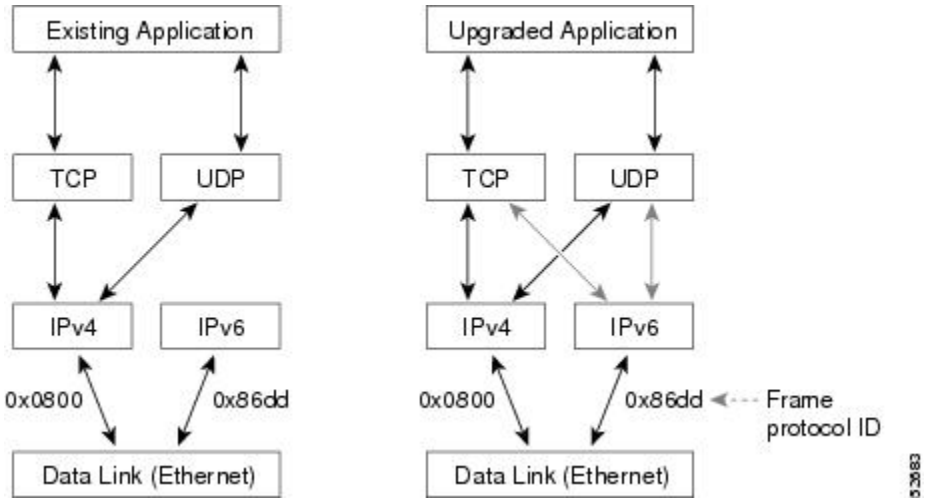
In IPv6 networks, a data link is a network sharing a particular link-local prefix. Data links are networks arbitrarily segmented by a network administrator in order to provide a multilevel, hierarchical routing structure while shielding the subnetwork from the addressing complexity of attached networks. The function of a subnetwork in IPv6 is similar to a subnetwork in IPv4. A subnetwork prefix is associated with one data link; multiple subnetwork prefixes may be assigned to the same data link.

The following data links are supported for IPv6: FDDI, Frame Relay PVC, Cisco High-Level Data Link Control (HDLC), PPP over Packet over SONET, ISDN, and serial interfaces.

Dual IPv4 and IPv6 Protocol Stacks

The dual IPv4 and IPv6 protocol stack technique can be used to transition to IPv6. It enables gradual, one-by-one upgrades to applications running on nodes. Applications running on nodes are upgraded to make use of the IPv6 protocol stack. Applications that are not upgraded (for example, they support only the IPv4 protocol stack) can coexist with upgraded applications on a node. New and upgraded applications make use of both the IPv4 and IPv6 protocol stacks (see the figure below).

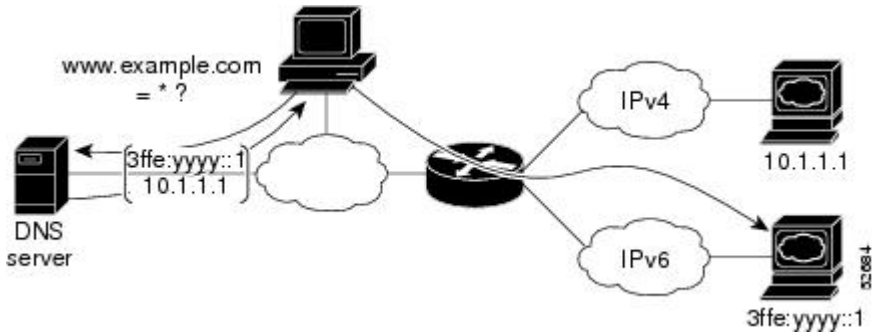
Figure 13: Dual IPv4 and IPv6 Protocol Stack Technique



One application program interface (API) supports both IPv4 and IPv6 addresses and DNS requests. An application can be upgraded to the new API and still use only the IPv4 protocol stack. The Cisco software supports the dual IPv4 and IPv6 protocol stack technique. When an interface is configured with both an IPv4 and an IPv6 address, the interface will forward both IPv4 and IPv6 traffic.

In the figure below, an application that supports dual IPv4 and IPv6 protocol stacks requests all available addresses for the destination hostname www.example.com from a DNS server. The DNS server replies with all available addresses (both IPv4 and IPv6 addresses) for www.example.com. The application chooses an address (in most cases, IPv6 addresses are the default choice), and connects the source node to the destination using the IPv6 protocol stack.

Figure 14: Dual IPv4 and IPv6 Protocol Stack Applications



How to Configure IPv6 Addressing and Basic Connectivity

Configuring IPv6 Addressing and Enabling IPv6 Routing

Perform this task to assign IPv6 addresses to individual device interfaces and enable IPv6 traffic forwarding globally on the device. By default, IPv6 addresses are not configured and IPv6 routing is disabled.

Note Multiple IPv6 link-local addresses on an interface are not supported.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ipv6 address** *ipv6-prefix /prefix-length* **eui-64**
   -
   - **ipv6 address** *ipv6-address / prefix-length* **link-local**
   -
   -

- **ipv6 enable**

5. **exit**
6. **ipv6 unicast-routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.

Step 4	Do one of the following:
	• **ipv6 address** *ipv6-prefix* /*prefix-length* **eui-64**
	•
	• **ipv6 address** *ipv6-address* / *prefix-length* **link-local**
	•
	•
	• **ipv6 enable**
	Example:
	`Device(config-if)# ipv6 address 2001:DB8:0:1::/64 eui-64`
	Example:
	Example:
	`Device(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local`
	Example:
	Example:
		Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface.
		or
		Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface.
		or
		Automatically configures an IPv6 link-local address on the interface while also enabling the interface for IPv6 processing. The link-local address can be used only to communicate with nodes on the same link.
		• Specifying the **ipv6 address eui-64** command configures global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID.
		• Specifying the **ipv6 address link-local** command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.

	Command or Action	Purpose
	Example: `Device(config-if)# ipv6 enable`	
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode, and returns the device to global configuration mode.
Step 6	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

Mapping Hostnames to IPv6 Addresses

Hostname-to-Address Mappings

A name server is used to track information associated with domain names. A name server can maintain a database of hostname-to-address mappings. Each name can map to one or more IPv4 addresses, IPv6 addresses, or both address types. In order to use this service to map domain names to IPv6 addresses, you must specify a name server and enable the DNS, which is the global naming scheme of the Internet that uniquely identifies network devices.

Cisco software maintains a cache of hostname-to-address mappings for use by the **connect**, **telnet**, and ping commands, related Telnet support operations, and many other commands that generate command output. This cache speeds the conversion of names to addresses.

Similar to IPv4, IPv6 uses a naming scheme that allows a network device to be identified by its location within a hierarchical name space that provides for domains. Domain names are joined with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that is identified by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the FTP server, for example, is identified as *ftp.cisco.com*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip domain name** [**vrf** *vrf-name*] *name*
   -
   -
   - **ip domain lis**t [**vrf** *vrf-name*] *name*
4. **ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*]
5. **ip domain-lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ip domain name** [**vrf** *vrf-name*] *name*
	•
	•
	• **ip domain lis**t [**vrf** *vrf-name*] *name*
	Example:
	`Device(config)# ip domain-name cisco.com`
	Example:
	Example:
	`Device(config)# ip domain list cisco1.com`
		(Optional) Defines a default domain name that Cisco software will use to complete unqualified hostnames.
		or
		(Optional) Defines a list of default domain names to complete unqualified hostnames.
		• You can specify a default domain name that Cisco software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any hostname that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up.
		Note The **ip domain name** and **ip domain list** commands are used to specify default domain names that can be used by both IPv4 and IPv6.

Step 4	**ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*] **Example:** `Device(config)# ip name-server 2001:DB8::250:8bff:fee8:f800 2001:DB8:0:f004::1`	Specifies one or more hosts that supply name information. • Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS. **Note** The *server-address*argument can be either an IPv4 or IPv6 address.
Step 5	**ip domain-lookup** **Example:** `Device(config)# ip domain-lookup`	Enables DNS-based address translation. • DNS is enabled by default.

Displaying IPv6 Redirect Messages

SUMMARY STEPS

1. **enable**
2. **show ipv6 interface** [**brief**] [*type number*] [**prefix**]
3. **show ipv6 route** [*ipv6-address* | *ipv6-prefix*/*prefix-length* | *protocol* | *interface-type interface-number*]
4. **show ipv6 traffic**
5. **show hosts** [**vrf** *vrf-name* | **all** | *hostname* | **summary**]
6. **enable**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 interface** [**brief**] [*type number*] [**prefix**] **Example:** `Device# show ipv6 interface gigabitethernet 0/0/0`	Displays the usability status of interfaces configured for IPv6.
Step 3	**show ipv6 route** [*ipv6-address* | *ipv6-prefix*/*prefix-length* | *protocol* | *interface-type interface-number*] **Example:** `Device# show ipv6 route`	(Optional) Displays the current contents of the IPv6 routing table.
Step 4	**show ipv6 traffic** **Example:** `Device# show ipv6 traffic`	(Optional) Displays statistics about IPv6 traffic.
Step 5	**show hosts** [**vrf** *vrf-name* | **all** | *hostname* | **summary**] **Example:** `Device# show hosts`	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.
Step 6	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 7	**show running-config** **Example:** `Device# show running-config`	Displays the current configuration running on the device.

Configuration Examples for IPv6 Addressing and Basic Connectivity

Example: IPv6 Addressing and IPv6 Routing Configuration

In the following example, IPv6 is enabled on the device with both a link-local address and a global address based on the IPv6 prefix 2001:DB8:c18:1::/64. The EUI-64 interface ID is used in the low-order 64 bits of both addresses. Output from the **show ipv6 interface** command is included to show how the interface ID (260:3EFF:FE47:1530) is appended to the link-local prefix FE80::/64 of Gigabit Ethernet interface 0/0/0.

```
ipv6 unicast-routing
interface gigabitethernet 0/0/0
  ipv6 address 2001:DB8:c18:1::/64 eui-64
Device# show ipv6 interface gigabitethernet 0/0/0
Gigabitethernet0/0/0 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::260:3EFF:FE47:1530
  Global unicast address(es):
    2001:DB8:C18:1:260:3EFF:FE47:1530, subnet is 2001:DB8:C18:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF47:1530
    FF02::9
  MTU is 1500 bytes
  ICMP error messages limited to one every 500 milliseconds
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

Example: Dual-Protocol Stacks Configuration

The following example enables the forwarding of IPv6 unicast datagrams globally on the device and configures Gigabit Ethernet interface 0/0/0 with both an IPv4 address and an IPv6 address:

```
ipv6 unicast-routing
interface gigabitethernet0/0/0
  ip address 192.168.99.1 255.255.255.0
 ipv6 address 2001:DB8:c18:1::3/64
```

Example: Hostname-to-Address Mappings Configuration

The following example defines two static hostname-to-address mappings in the hostname cache, establishes a domain list with several alternate domain names to complete unqualified hostnames, specifies host 2001:DB8::250:8bff:fee8:f800 and host 2001:DB8:0:f004::1 as the name servers, and reenables the DNS service:

```
ip domain list domain1-list.com
ip domain list serviceprovider2-name.com
ip domain list college2-name.edu
ip name-server 2001:DB8::250:8bff:fee8:f800 2001:DB8:0:f004::1
ip domain-lookup
```

Additional References for IPv6 Services: AAAA DNS Lookups

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv4 services configuration	*IP Application Services Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

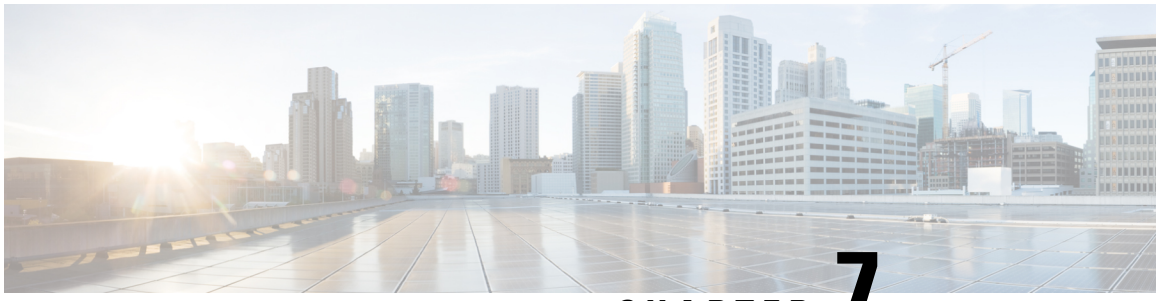
Feature Information for IPv6 Addressing and Basic Connectivity

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 19: Feature Information for IPv6 Addressing and Basic Connectivity

Feature Name	Releases	Feature Information
Internet Protocol version 6 (IPv6)	Cisco IOS XE 17.4.1	This feature was introduced in the Catalyst 8000 Series platforms.



CHAPTER 7

IPv6 Anycast Address

An IPv6 anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes. Anycast addresses are syntactically indistinguishable from unicast addresses, because anycast addresses are allocated from the unicast address space.

Information About IPv6 Anycast Address

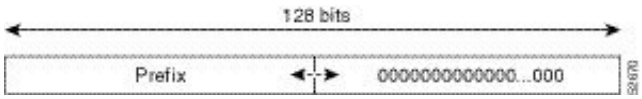
IPv6 Address Type: Anycast

An anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes. A packet sent to an anycast address is delivered to the closest interface (as defined by the routing protocols in use) identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses, because anycast addresses are allocated from the unicast address space. Assigning a unicast address to more than one interface makes a unicast address an anycast address. Nodes to which the anycast address is assigned must be explicitly configured to recognize that the address is an anycast address.

Note Anycast addresses can be used only by a , not a host, and anycast addresses must not be used as the source address of an IPv6 packet.

The figure below shows the format of the subnet anycast address; the address has a prefix concatenated by a series of zeros (the interface ID). The subnet anycast address can be used to reach a device on the link that is identified by the prefix in the subnet anycast address.

Figure 15: Subnet Anycast Address Format



How to Configure IPv6 Anycast Address

Configuring IPv6 Anycast Addressing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** *ipv6-prefix/prefix-length* **anycast**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface tunnel0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix/prefix-length* **anycast** **Example:** `Device(config-if)# ipv6 address 2002:db8:c058::/128 anycast`	Specifying the **ipv6 address anycast** command adds an IPv6 anycast address.

Configuration Examples for IPv6 Anycast Address

Example: Configuring IPv6 Anycast Addressing

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference, All Releases
Cisco IOS IP SLAs: general information	"Cisco IOS IP SLAs Overview" module of the *Cisco IOS IP SLAs Configuration Guide*.
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" module of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

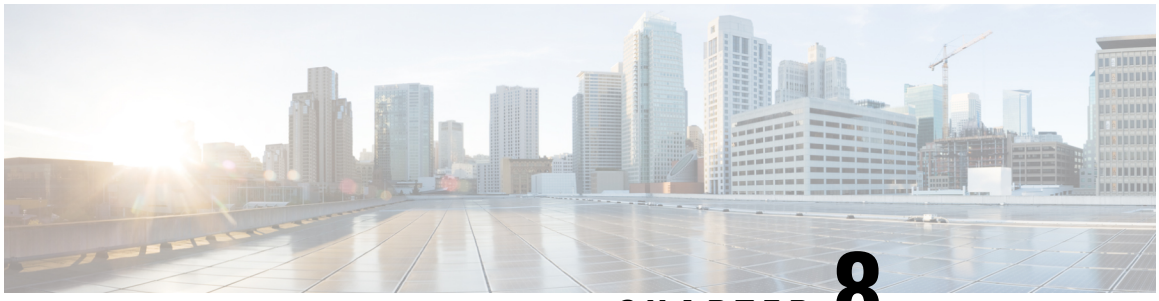
Feature Information for IPv6 Anycast Address

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 20: Feature Information for IPv6 Anycast Address

Feature Name	Releases	Feature Information
IPv6: Anycast Address	12.2(25)SEA 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.3(4)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2.0SG	An anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes. Anycast addresses are syntactically indistinguishable from unicast addresses, because anycast addresses are allocated from the unicast address space. The following commands were introduced or modified: **ipv6 address anycast**, **show ipv6 interface**.



CHAPTER 8

IPv6 Switching: Cisco Express Forwarding Support

The Cisco Express Forwarding feature is Layer 3 IP switching technology for the forwarding of IPv6 packets.

Prerequisites for IPv6 Switching: Cisco Express Forwarding

- To forward IPv6 traffic using Cisco Express Forwarding , you must configure forwarding of IPv6 unicast datagrams globally on the device, and you must configure an IPv6 address on an interface.
- You must enable Cisco Express Forwarding for IPv4 globally on the device before enabling Cisco Express Forwarding for IPv6 globally on the device.
- Nondistributed platforms do not support distributed Cisco Express Forwarding; however, some distributed platforms support both Cisco Express Forwarding and distributed Cisco Express Forwarding.
- To use Unicast Reverse Path Forwarding (uRPF), enable Cisco Express Forwarding switching in the device. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the device, individual interfaces can be configured with other switching modes.

The following restrictions apply to nondistributed and distributed architecture platforms configured for Cisco Express Forwarding :

- IPv6 packets that have global source and destination addresses are Cisco Express Forwarding-switched .
- IPv6 packets that have link-local source and destination addresses are process-switched.
- IPv6 packets that are tunneled within manually configured IPv6 tunnels are Cisco Express Forwarding-switched.

Information About IPv6 Switching: Cisco Express Forwarding Support

Cisco Express Forwarding for IPv6

Cisco Express Forwarding is advanced, Layer 3 IP switching technology for the forwarding of IPv6 packets.

Each IPv6 router interface has an association to one IPv6 global FIB and one IPv6 link-local FIB (multiple interfaces can have an association to the same FIB). All IPv6 router interfaces that are attached to the same IPv6 link share the same IPv6 link-local FIB. IPv6 packets that have an IPv6 global destination address are processed by the IPv6 global FIB; however, packets that have an IPv6 global destination address and an IPv6 link-local source address are sent to the RP for process switching and scope-error handling. Packets that have a link-local source address are not forwarded off of the local link and are sent to the RP for process switching and scope-error handling.

How to Configure IPv6 Switching: Cisco Express Forwarding Support

Configuring Cisco Express Forwarding

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do the following:
   - **ipv6 cef**
4. **ipv6 cef accounting** [**non-recursive** | **per-prefix** | **prefix-length**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	Do the following: • **ipv6 cef** **Example:** `Device(config)# ipv6 cef`	Enables Cisco Express Forwarding globally on the device.
Step 4	**ipv6 cef accounting** [**non-recursive** | **per-prefix** | **prefix-length**] **Example:** `Device(config)# ipv6 cef accounting`	Enables Cisco Express Forwarding network accounting globally on the device. • Network accounting for Cisco Express Forwarding enables you to better understand Cisco Express Forwarding traffic patterns within your network by collecting statistics specific to Cisco Express Forwarding traffic. For example, network accounting for Cisco Express Forwarding enables you to collect information such as the number of packets and bytes switched to a destination or the number of packets switched through a destination. • The optional **per-prefix** keyword enables the collection of the number of packets and bytes express forwarded to an IPv6 destination (or IPv6 prefix). • The optional **prefix-length** keyword enables the collection of the number of packets and bytes express forwarded to an IPv6 prefix length. **Note** When Cisco Express Forwarding is enabled globally on the device, accounting information is collected at the RP.

Configuration Examples for IPv6 Switching: Cisco Express Forwarding Support

Example: Cisco Express Forwarding Configuration

In the following example, both Cisco Express Forwarding for IPv6 and network accounting for Cisco Express Forwarding for IPv6 have been enabled globally on a nondistributed architecture device, and Cisco Express Forwarding for IPv6 has been enabled on Gigabit Ethernet interface 0/0/0. The example also shows that the forwarding of IPv6 unicast datagrams has been configured globally on the device with the **ipv6 unicast-routing** command, an IPv6 address has been configured on Gigabit Ethernet interface 0/0/0 with the **ipv6 address** command, and Cisco Express Forwarding for IPv4 has been configured globally on the device with the **ip cef** command.

```
ip cef
```

```
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
interface gigabitethernet0/0/0
 ip address 10.4.9.11 255.0.0.0
 media-type 10BaseT
 ipv6 address 2001:DB8:C18:1::/64 eui-64
```

Additional References

Related Documents

Related Topic	Document Title
Cisco Express Forwarding for IPv6	Implementing IPv6 Addressing and Basic Connectivity Guide, *IPv6 Configuration Guide*
Cisco IOS voice configuration	Cisco IOS Voice Configuration Library
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands, including voice commands	Cisco IOS IPv6 Command Reference
Cisco Unified Border Element configuration	Cisco Unified Border Element Configuration Guide
SIP Configuration Guide	*SIP Configuration Guide*
Troubleshooting and debugging guides	Cisco IOS Debug Command Reference *Troubleshooting and Debugging VoIP Call Basics*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

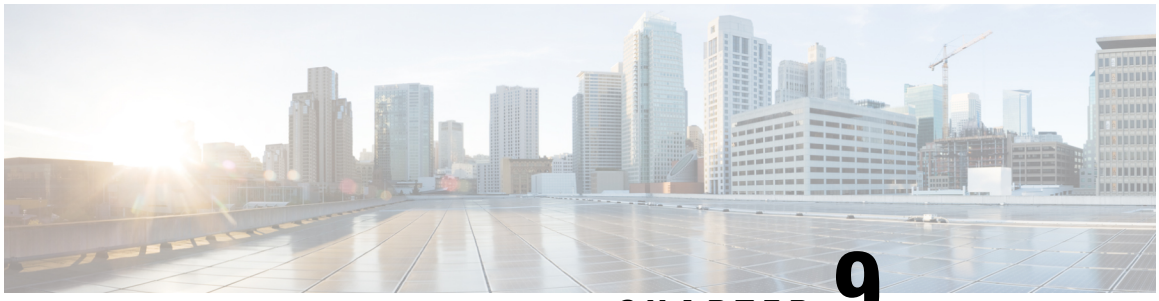
Feature Information for IPv6 Switching: Cisco Express Forwarding and Distributed Cisco Express Forwarding Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 21: Feature Information for IPv6 Switching: Cisco Express Forwarding and Distributed Cisco Express Forwarding Support

Feature Name	Releases	Feature Information
IPv6 Switching: Cisco Express Forwarding and Distributed Cisco Express Forwarding Support	12.2(13)T 12.2(17a)SX1 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 15.0(2)SG 15.3(1)S Cisco IOS XE Release 2.1 3.2.0SG	Cisco Express Forwarding for IPv6 is advanced, Layer 3 IP switching technology for the forwarding of IPv6 packets. Distributed Cisco Express Forwarding for IPv6 performs the same functions as Cisco Express Forwarding for IPv6 but for distributed architecture platforms. The following commands were introduced or modified: **ipv6 cef**, **ipv6 cef accounting**, **ipv6 cef distributed**.



CHAPTER 9

Unicast Reverse Path Forwarding for IPv6

The Unicast Reverse Path Forwarding(uRPF) for IPv6 feature mitigates problems caused by malformed or forged (spoofed) IPv6 source addresses that pass through an IPv6 device.

Prerequisites for Unicast Reverse Path Forwarding for IPv6

- Enable Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching in the device. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the device, individual interfaces can be configured with other switching modes.
- Cisco Express Forwarding must be configured globally in the device. uRPF will not work without Cisco Express Forwarding.
- uRPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry; this means that there are multiple routes to the source of a packet. uRPF should be applied only where there is natural or configured symmetry.

  For example, devices at the edge of the network of an ISP are more likely to have symmetrical reverse paths than devices that are in the core of the ISP network. Devices that are in the core of the ISP network have no guarantee that the best forwarding path out of the device will be the path selected for packets returning to the device. Therefore, we do not recommend that you apply uRPF where there is a chance of asymmetric routing. Place uRPF only at the edge of a network or, for an ISP, at the customer edge of the network.

Information About Unicast Reverse Path Forwarding for IPv6

Unicast Reverse Path Forwarding

Use the Unicast Reverse Path Forwarding for IPv6 feature to mitigate problems caused by malformed or spoofed IPv6 source addresses that pass through an IPv6 device. Malformed or forged source addresses can indicate denial-of-service (DoS) attacks based on source IPv6 address spoofing.

When uRPF is enabled on an interface, the device examines all packets received on that interface. The device verifies that the source address appears in the routing table and matches the interface on which the packet was received. This "look backward" ability is available only when Cisco Express Forwarding is enabled on the device; this is because the lookup relies on the presence of the Forwarding Information Bases (FIBs). Cisco Express Forwarding generates the FIB as part of its operation.

Note uRPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

Note With uRPF, all equal-cost "best" return paths are considered valid. uRPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB.

How to Configure Unicast Reverse Path Forwarding for IPv6

Configuring Unicast RPF

Before you begin

To use uRPF, enable Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching in the device. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the device, individual interfaces can be configured with other switching modes.

Note Cisco Express Forwarding must be configured globally in the device. uRPF does not work without Cisco Express Forwarding.

Note uRPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, meaning that there are multiple routes to the source of a packet. uRPF should be applied only where there is natural or configured symmetry.

For example, devices at the edge of the network of an ISP are more likely to have symmetrical reverse paths than devices that are in the core of the ISP network. Devices that are in the core of the ISP network have no guarantee that the best forwarding path out of the device will be the path selected for packets returning to the device. Therefore, we do not recommend that you apply uRPF where there is a chance of asymmetric routing. It is simplest to place uRPF only at the edge of a network or, for an ISP, at the customer edge of the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 verify unicast source reachable-via** {**rx** | **any**} [**allow-default**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 verify unicast source reachable-via** {**rx** | **any**} [**allow-default**] **Example:** `Device(config-if)# ipv6 verify unicast source reachable-via any`	Verifies that a source address exists in the FIB table and enables uRPF. "rx" is for strict mode and "any" is for loose mode.

Configuration Examples for Unicast Reverse Path Forwarding for IPv6

Example: Configuring Unicast Reverse Path Forwarding for IPv6

Additional References

Related Documents

Related Topic	Document Title
Cisco Express Forwarding for IPv6	Implementing IPv6 Addressing and Basic Connectivity Guide, *IPv6 Configuration Guide*
Cisco IOS voice configuration	Cisco IOS Voice Configuration Library
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands, including voice commands	Cisco IOS IPv6 Command Reference
Cisco Unified Border Element configuration	Cisco Unified Border Element Configuration Guide
SIP Configuration Guide	*SIP Configuration Guide*
Troubleshooting and debugging guides	Cisco IOS Debug Command Reference *Troubleshooting and Debugging VoIP Call Basics*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

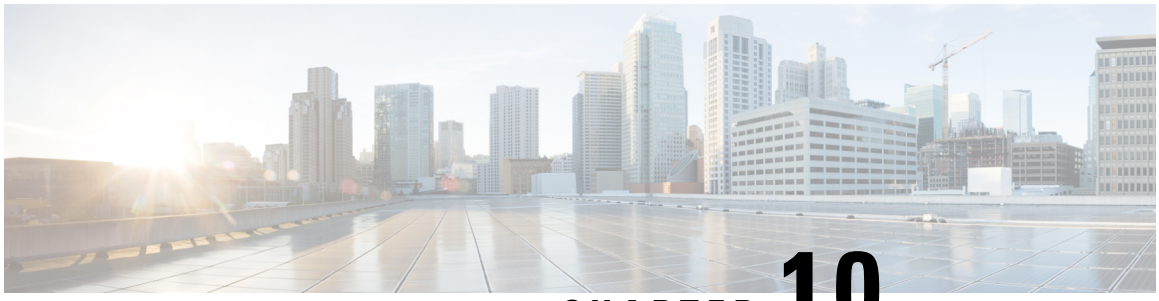
Feature Information for Unicast Reverse Path Forwarding for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 22: Feature Information for Unicast Reverse Path Forwarding for IPv6

Feature Name	Releases	Feature Information
Unicast Reverse Path Forwarding for IPv6		Use the uRPF feature to mitigate problems caused by malformed or spoofed IPv6 source addresses that pass through an IPv6 device. Malformed or forged source addresses can indicate DoS attacks based on source IPv6 address spoofing. The following commands were introduced or modified: **ipv6 verify unicast source reachable-via**, **show ipv6 traffic**.



CHAPTER 10

IPv6 Services: AAAA DNS Lookups over an IPv4 Transport

IPv6 basic connectivity can be enhanced by configuring support for AAAA record types in the DNS name-to-address and address-to-name lookup processes.

Information About IPv6 Services: AAAA DNS Lookups over an IPv4 Transport

DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses. IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

The table below lists the IPv6 DNS record types.

Table 23: IPv6 DNS Record Types

Record Type	Description	Format
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.)	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.) **Note** Cisco software supports resolution of PTR records for the IP6.INT domain.	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

Additional References for IPv6 Services: AAAA DNS Lookups

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv4 services configuration	*IP Application Services Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

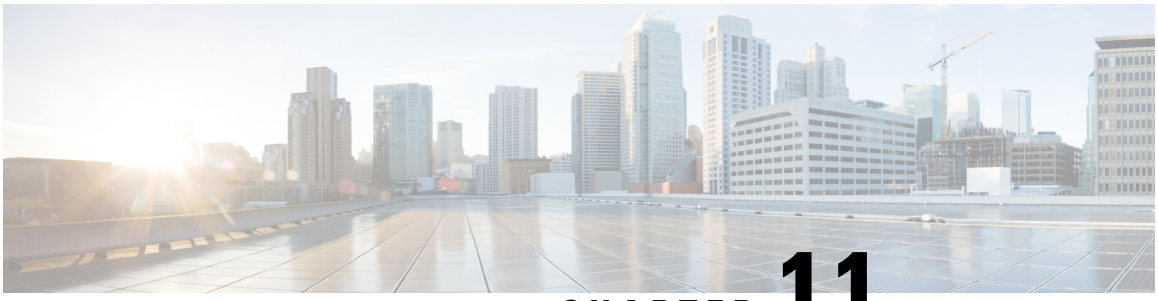
Feature Information for IPv6 Services: AAAA DNS Lookups over an IPv4 Transport

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 24: Feature Information for IPv6 Services: AAAA DNS Lookups over an IPv4 Transport

Feature Name	Releases	Feature Information
IPv6 Services: AAAA DNS Lookups over an IPv4 Transport	12.2(2)T 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 12.2(17a)SX1 15.0(2)SG 15.3(1)S Cisco IOS XE Release 2.1 3.2.0SG	IPv6 basic connectivity can be enhanced by configuring support for AAAA record types in the DNS name-to-address and address-to-name lookup processes. No commands were introduced or modified.



CHAPTER 11

IPv6 MTU Path Discovery

IPv6 MTU Path Discovery allows a host to dynamically discover and adjust to differences in the maximum transmission unit (MTU) size of every link along a given data path.

Information About IPv6 MTU Path Discovery

IPv6 MTU Path Discovery

As in IPv4, path MTU discovery in IPv6 allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 device processing resources and helps IPv6 networks run more efficiently.

Note In IPv6, the minimum link MTU is 1280 octets. We recommend using an MTU value of 1500 octets for IPv6 links.

With IPv6 path MTU discovery, a device originating IPv6 traffic has an MTU cache that contains MTU values received in ICMPv6 "toobig" messages. In order to prevent an attacker from filling the MTU cache, the device keeps track of the destinations to which it has originated (sent) traffic, and only accepts toobig ICMPv6 messages that have an inner destination matching one of these tracked destinations.

If a malicious device can learn to which destination the device is originating traffic, it could still send a toobig ICMPv6 message to the device for this destination, even if the attacker is not on the path to this destination, and succeeds in forcing his entry into the MTU cache. The device then starts fragmenting traffic to this destination, which significantly affects device performance.

Enabling flow-label marking for locally generated traffic can mitigate this attack. Originated packets are marked with a flow label (which is randomly generated and changed every minute), and toobig messages

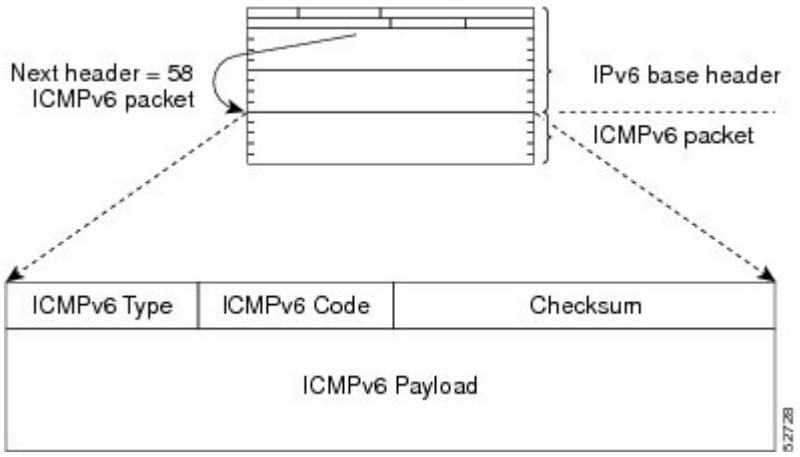
received are checked against the values sent. Unless an attacker can snoop traffic, the attacker will not know which flow label to use, and its toobig message will be dropped.

ICMP for IPv6

Internet Control Message Protocol (ICMP) in IPv6 functions the same as ICMP in IPv4. ICMP generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery process, path MTU discovery, and the Multicast Listener Discovery (MLD) protocol for IPv6. MLD is used by IPv6 devices to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. MLD is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. ICMP packets in IPv6 are like a transport-layer packet in the sense that the ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is derived (computed by the sender and checked by the receiver) from the fields in the IPv6 ICMP packet and the IPv6 pseudoheader. The ICMPv6 Data field contains error or diagnostic information relevant to IP packet processing. The figure below shows the IPv6 ICMP packet header format.

Figure 16: IPv6 ICMP Packet Header Format



How to Configure IPv6 MTU Path Discovery

Enabling Flow-Label Marking in Packets that Originate from the Device

This feature allows the device to track destinations to which the device has sent packets that are 1280 bytes or larger.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ipv6 flowset**
4. **exit**
5. **clear ipv6 mtu**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 flowset** **Example:** `Device(config)# ipv6 flowset`	Configures flow-label marking in 1280-byte or larger packets sent by the device.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode, and places the device in privileged EXEC mode.
Step 5	**clear ipv6 mtu** **Example:** `Device# clear ipv6 mtu`	Clears the MTU cache of messages.

Configuration Examples for IPv6 MTU Path Discovery

Example: Displaying IPv6 Interface Statistics

In the following example, the **show ipv6 interface** command is used to verify that IPv6 addresses are configured correctly for FastEthernet interface 1/0. Information may also be displayed about the status of IPv6 neighbor redirect messages, IPv6 neighbor discovery messages, stateless autoconfiguration, and MTU size.

```
Device# show ipv6 interface fastethernet 1/0

Ethernet0 is up, line protocol is up
  IPv6 is stalled, link-local address is FE80::1
  Global unicast address(es):
    2001:DB8:2000::1, subnet is 2001:DB8:2000::/64
    2001:DB8:3000::1, subnet is 2001:DB8:3000::/64
  Joined group address(es):
```

```
  FF02::1
  FF02::2
  FF02::1:FF00:1
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

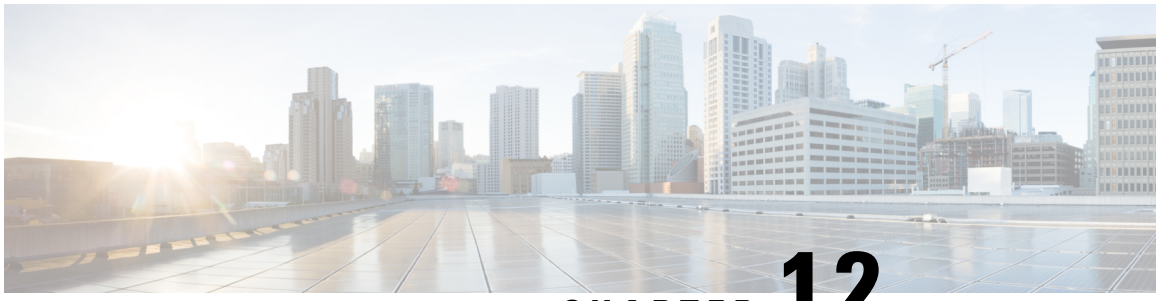
Feature Information for IPv6 MTU Path Discovery

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 25: Feature Information for IPv6 MTU Path Discovery

Feature Name	Releases	Feature Information
IPv6 MTU Path Discovery	12.2(2)T 12.2(17a)SX1 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 15.0(2)SG Cisco IOS XE Release 2.1 3.2.0SG	Path MTU discovery in IPv6 allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. The following commands were introduced or modified: **clear ipv6 mtu**, **ipv6 flowset**.



CHAPTER 12

ICMP for IPv6

ICMP in IPv6 functions the same as ICMP in IPv4. ICMP for IPv6 generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages.

Information About ICMP for IPv6

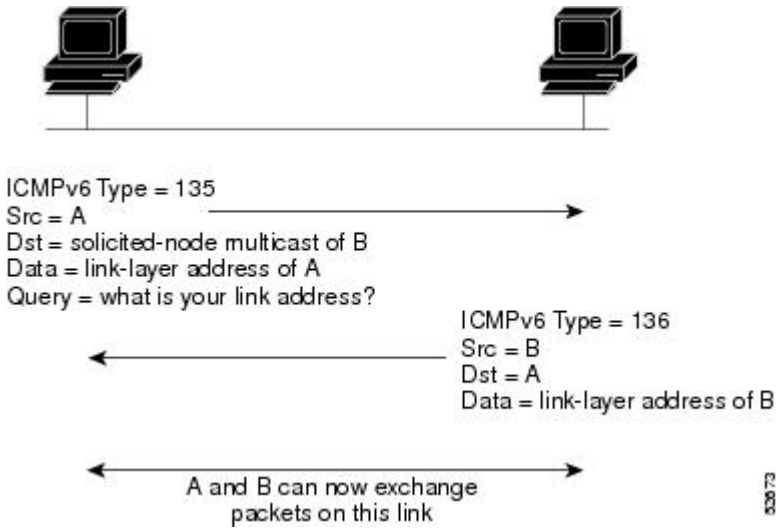
ICMP for IPv6

ICMP in IPv6 functions the same as ICMP in IPv4. ICMP for IPv6 generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages.

IPv6 Neighbor Solicitation Message

A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link (see the figure below). When a node wants to determine the link-layer address of another node, the source address in a neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The destination address in the neighbor solicitation message is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

Figure 17: IPv6 Neighbor Discovery: Neighbor Solicitation Message



After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node (more specifically, the IPv6 address of the node interface) sending the neighbor advertisement message. The destination address in the neighbor advertisement message is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verify the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor, and is used for all paths between hosts and neighboring nodes (hosts or devices). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment from an upper-layer protocol (such as TCP) indicates that a connection is making forward progress (reaching its destination) or the receipt of a neighbor advertisement message in response to a neighbor solicitation message. If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Therefore, forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop device is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working.

The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

Note A neighbor advertisement message that has the solicited flag set to a value of 0 must not be considered as a positive acknowledgment that the forward path is still working.

Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). Specifically, a node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

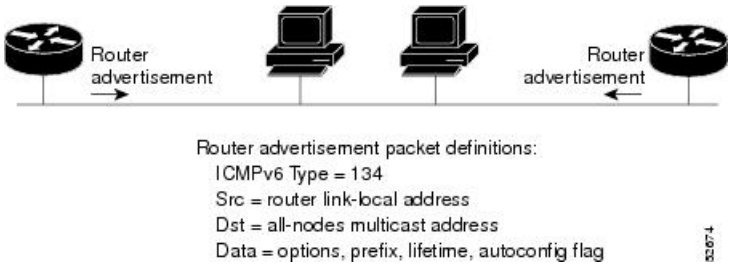
Every IPv6 unicast address (global or link-local) must be verified for uniqueness on the link; however, until the uniqueness of the link-local address is verified, duplicate address detection is not performed on any other IPv6 addresses associated with the link-local address. The Cisco implementation of duplicate address detection in the Cisco software does not verify the uniqueness of anycast or global addresses that are generated from 64-bit interface identifiers.

IPv6 Router Advertisement Message

Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out each configured interface of an IPv6 router. For stateless autoconfiguration to work properly, the advertised prefix length in RA messages must always be 64 bits.

The RA messages are sent to the all-nodes multicast address (see the figure below).

Figure 18: IPv6 Neighbor Discovery--RA Message



RA messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses

- Lifetime information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time (in seconds) the router should be used as a default router)
- Additional information for hosts, such as the hop limit and MTU a host should use in packets that it originates

RAs are also sent in response to router solicitation messages.

The following RA message parameters can be configured:

- The time interval between periodic RA messages
- The "router lifetime" value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time a node considers a neighbor reachable (for use by all nodes on a given link)

The configured parameters are specific to an interface. The sending of RA messages (with default values) is automatically enabled on FDDI interfaces when the **ipv6 unicast-routing** command is configured. For other interface types, the sending of RA messages must be manually configured by using the **no ipv6 nd ra suppress**command. The sending of RA messages can be disabled on individual interfaces by using the **ipv6 nd ra**suppress command.

Default Router Preferences for Traffic Engineering

Hosts discover and select default devices by listening to Router Advertisements (RAs). Typical default device selection mechanisms are suboptimal in certain cases, such as when traffic engineering is needed. For example, two devices on a link may provide equivalent but not equal-cost routing, and policy may dictate that one of the devices is preferred. Some examples are as follows:

- Multiple devices that route to distinct sets of prefixes—Redirects (sent by nonoptimal devices for a destination) mean that hosts can choose any device and the system will work. However, traffic patterns may mean that choosing one of the devices would lead to considerably fewer redirects.
- Accidentally deploying a new device—Deploying a new device before it has been fully configured could lead to hosts adopting the new device as a default device and traffic disappearing. Network managers may want to indicate that some devices are more preferred than others.
- Multihomed situations—Multihomed situations may become more common, because of multiple physical links and because of the use of tunneling for IPv6 transport. Some of the devices may not provide full default routing because they route only to the 6-to-4 prefix or they route only to a corporate intranet. These situations cannot be resolved with redirects, which operate only over a single link.

The default router preference (DRP) feature provides a basic preference metric (low, medium, or high) for default devices. The DRP of a default device is signaled in unused bits in RA messages. This extension is backward compatible, both for devices (setting the DRP bits) and hosts (interpreting the DRP bits). These bits are ignored by hosts that do not implement the DRP extension. Similarly, the values sent by devices that do not implement the DRP extension will be interpreted by hosts that do implement it as indicating a “medium” preference. DRPs need to be configured manually.

Additional References for IPv6 Neighbor Discovery Multicast Suppress

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

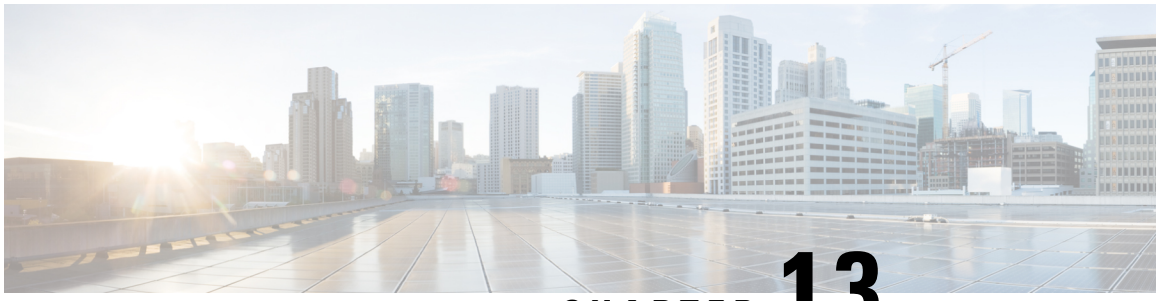
Feature Information for ICMP for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 26: Feature Information for ICMP for IPv6

Feature Name	Releases	Feature Information
IPv6: ICMPv6	12.0(22)S 12.2(2)T 12.2(14)S 12.2(17a)SX1 12.2(25)SG 12.2(28)SB 12.2(33)SRA 12.2(2)T 15.3(1)S Cisco IOS XE Release 2.1	ICMP in IPv6 functions similarly to ICMP in IPv4. ICMP generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages. No commands were introduced or modified.



CHAPTER 13

IPv6 ICMP Rate Limiting

The IPv6 ICMP rate limiting feature implements a token bucket algorithm for limiting the rate at which IPv6 Internet Control Message Protocol (ICMP) error messages are sent out on the network.

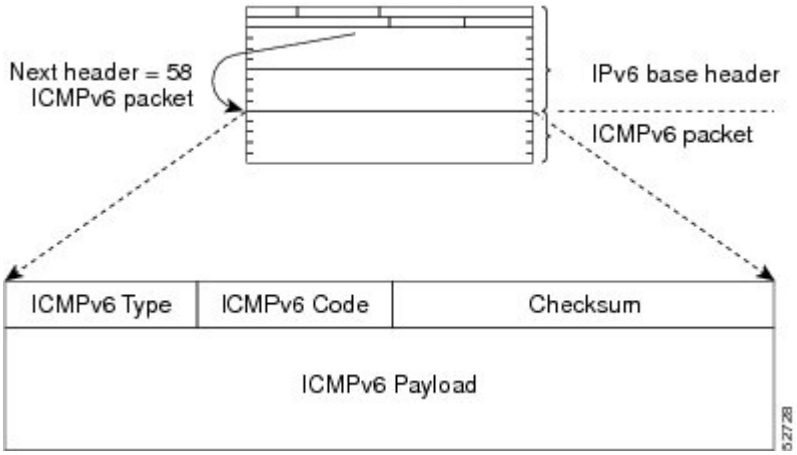
Information About IPv6 ICMP Rate Limiting

ICMP for IPv6

Internet Control Message Protocol (ICMP) in IPv6 functions the same as ICMP in IPv4. ICMP generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery process, path MTU discovery, and the Multicast Listener Discovery (MLD) protocol for IPv6. MLD is used by IPv6 devices to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. MLD is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. ICMP packets in IPv6 are like a transport-layer packet in the sense that the ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is derived (computed by the sender and checked by the receiver) from the fields in the IPv6 ICMP packet and the IPv6 pseudoheader. The ICMPv6 Data field contains error or diagnostic information relevant to IP packet processing. The figure below shows the IPv6 ICMP packet header format.

Figure 19: IPv6 ICMP Packet Header Format



IPv6 ICMP Rate Limiting

The IPv6 ICMP rate limiting feature implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out on the network. The initial implementation of IPv6 ICMP rate limiting defined a fixed interval between error messages, but some applications such as traceroute often require replies to a group of requests sent in rapid succession. The fixed interval between error messages is not flexible enough to work with applications such as traceroute and can cause the application to fail.

Implementing a token bucket scheme allows a number of tokens--representing the ability to send one error message each--to be stored in a virtual bucket. The maximum number of tokens allowed in the bucket can be specified, and for every error message to be sent, one token is removed from the bucket. If a series of error messages is generated, error messages can be sent until the bucket is empty. When the bucket is empty of tokens, no IPv6 ICMP error messages are sent until a new token is placed in the bucket. The token bucket algorithm does not increase the average rate limiting time interval, and it is more flexible than the fixed time interval scheme.

How to Configure IPv6 ICMP Rate Limiting

Customizing IPv6 ICMP Rate Limiting

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 icmp error-interval** *milliseconds* [*bucketsize*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 icmp error-interval** *milliseconds* [*bucketsize*] **Example:** `Device(config)# ipv6 icmp error-interval 50 20`	Customizes the interval and bucket size for IPv6 ICMP error messages.

Configuration Examples for IPv6 ICMP Rate Limiting

Example: IPv6 ICMP Rate Limiting Configuration

The following example shows an interval of 50 milliseconds and a bucket size of 20 tokens being configured for IPv6 ICMP error messages:

```
ipv6 icmp error-interval 50 20
```

Example: Displaying Information About ICMP Rate-Limited Counters

In the following example, information about ICMP rate-limited counters is displayed:

```
Device# show ipv6 traffic

ICMP statistics:
  Rcvd: 188 input, 0 checksum errors, 0 too short
        0 unknown info type, 0 unknown error type
        unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
        parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout,0 too big
        0 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        1 router solicit, 175 router advert, 0 redirects
        0 neighbor solicit, 12 neighbor advert
  Sent: 7376 output, 56 rate-limited
        unreach: 0 routing, 15 admin, 0 neighbor, 0 address, 0 port
        parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout,0 too big
        15 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        0 router solicit, 7326 router advert, 0 redirects
        2 neighbor solicit, 22 neighbor advert
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

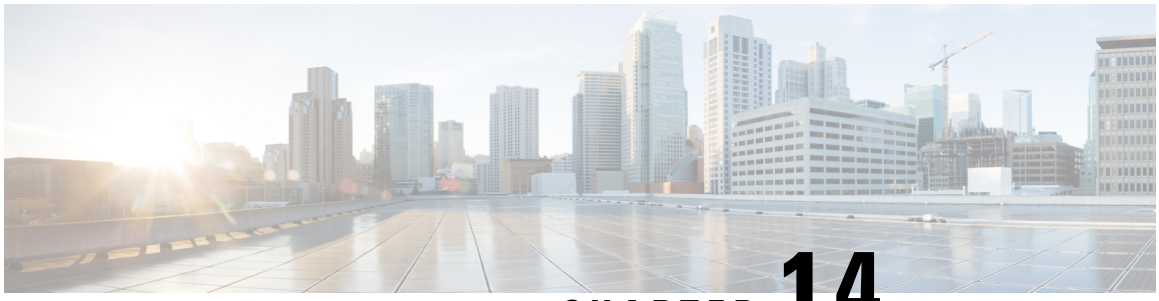
Feature Information for IPv6 ICMP Rate Limiting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 27: Feature Information for IPv6 ICMP Rate Limiting

Feature Name	Releases	Feature Information
IPv6 ICMP Rate Limiting	12.2(8)T 15.3(1)S Cisco IOS XE Release 2.1	The IPv6 ICMP Rate Limiting feature implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out on the network. The following commands were introduced or modified: **ipv6 icmp error-interval**.



CHAPTER 14

ICMP for IPv6 Redirect

The IPv6 Redirect Messages feature enables a device to send Internet Control Message Protocol (ICMP) IPv6 neighbor redirect messages to inform hosts of better first-hop nodes (devices or hosts) on the path to a destination.

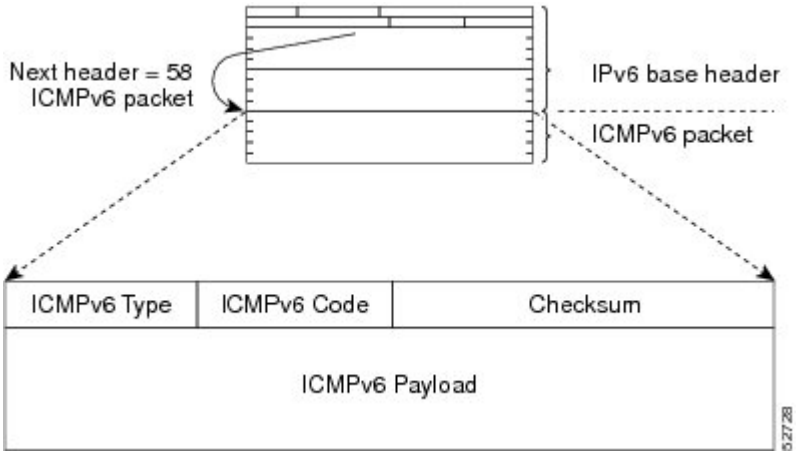
Information About ICMP for IPv6 Redirect

ICMP for IPv6

Internet Control Message Protocol (ICMP) in IPv6 functions the same as ICMP in IPv4. ICMP generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery process, path MTU discovery, and the Multicast Listener Discovery (MLD) protocol for IPv6. MLD is used by IPv6 devices to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. MLD is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. ICMP packets in IPv6 are like a transport-layer packet in the sense that the ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is derived (computed by the sender and checked by the receiver) from the fields in the IPv6 ICMP packet and the IPv6 pseudoheader. The ICMPv6 Data field contains error or diagnostic information relevant to IP packet processing. The figure below shows the IPv6 ICMP packet header format.

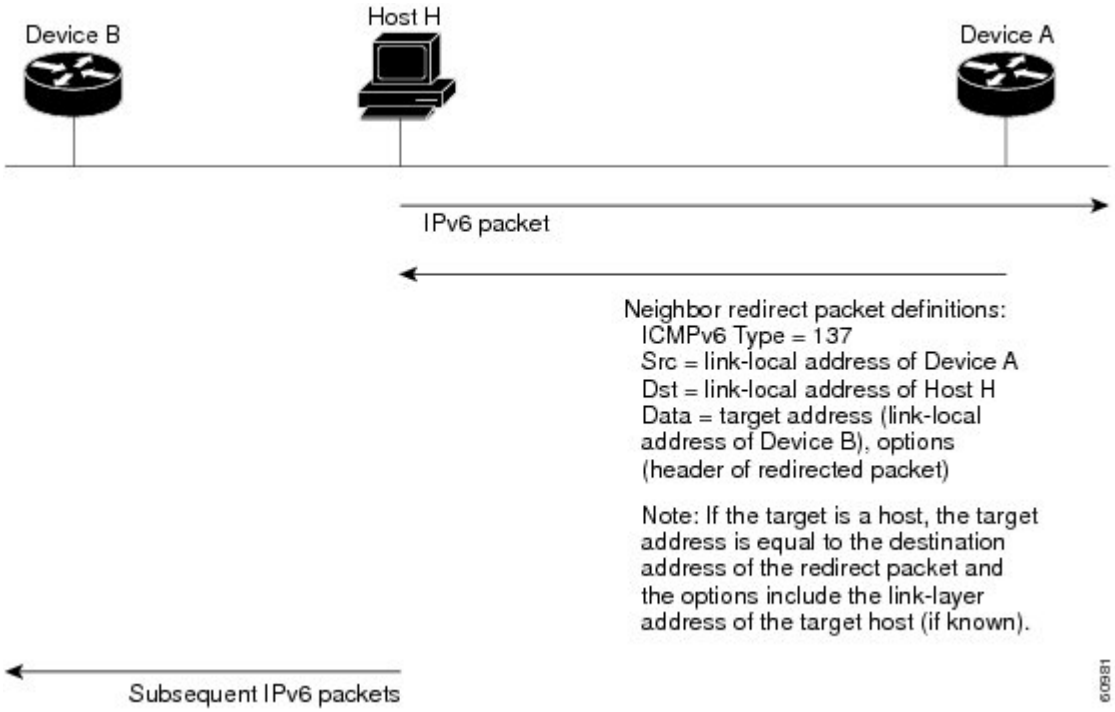
Figure 20: IPv6 ICMP Packet Header Format



IPv6 Neighbor Redirect Message

A value of 137 in the type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Devices send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination (see the figure below).

Figure 21: IPv6 Neighbor Discovery: Neighbor Redirect Message



Note A device must be able to determine the link-local address for each of its neighboring devices in order to ensure that the target address (the final destination) in a redirect message identifies the neighbor device by its link-local address. For static routing, the address of the next-hop device should be specified using the link-local address of the device; for dynamic routing, all IPv6 routing protocols must exchange the link-local addresses of neighboring devices.

After forwarding a packet, a device should send a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the device.
- The packet is about to be sent out the interface on which it was received.
- The device determines that a better first-hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link, or a link-local address.

Use the **ipv6 icmp error-interval** command to limit the rate at which the device generates all IPv6 ICMP error messages, including neighbor redirect messages, which ultimately reduces link-layer congestion.

Note A device must not update its routing tables after receiving a neighbor redirect message, and hosts must not originate neighbor redirect messages.

How to Display IPv6 Redirect Messages

Displaying IPv6 Redirect Messages

SUMMARY STEPS

1. **enable**
2. **show ipv6 interface** [**brief**] [*type number*] [**prefix**]
3. **show ipv6 route** [*ipv6-address* | *ipv6-prefix*/*prefix-length* | *protocol* | *interface-type interface-number*]
4. **show ipv6 traffic**
5. **show hosts** [**vrf** *vrf-name* | **all** | *hostname* | **summary**]
6. **enable**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 interface** [**brief**] [*type number*] [**prefix**] **Example:** `Device# show ipv6 interface gigabitethernet 0/0/0`	Displays the usability status of interfaces configured for IPv6.
Step 3	**show ipv6 route** [*ipv6-address* | *ipv6-prefix*/*prefix-length* | *protocol* | *interface-type interface-number*] **Example:** `Device# show ipv6 route`	(Optional) Displays the current contents of the IPv6 routing table.
Step 4	**show ipv6 traffic** **Example:** `Device# show ipv6 traffic`	(Optional) Displays statistics about IPv6 traffic.
Step 5	**show hosts** [**vrf** *vrf-name* | **all** | *hostname* | **summary**] **Example:** `Device# show hosts`	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.
Step 6	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 7	**show running-config** **Example:** `Device# show running-config`	Displays the current configuration running on the device.

Configuration Examples for ICMP for IPv6 Redirect

Example: Displaying IPv6 Interface Statistics

In the following example, the **show ipv6 interface** command is used to verify that IPv6 addresses are configured correctly for GigabitEthernet interface 0/0/0. Information is also displayed about the

status of IPv6 neighbor redirect messages, IPv6 neighbor discovery messages, and stateless autoconfiguration.

```
Device# show ipv6 interface gigabitethernet 0/0/0

GigabitEthernet0/0/0 is up, line protocol is up
  IPv6 is stalled, link-local address is FE80::1
  Global unicast address(es):
    2001:DB8:2000::1, subnet is 2001:DB8:2000::/64
    2001:DB8:3000::1, subnet is 2001:DB8:3000::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

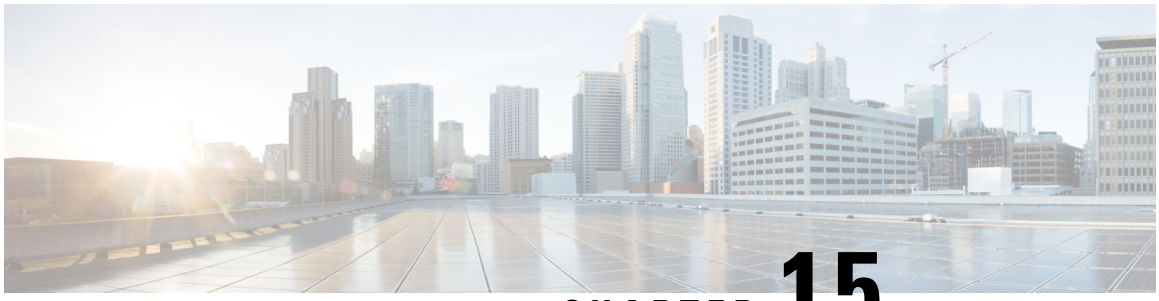
Feature Information for ICMP for IPv6 Redirect

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 28: Feature Information for ICMPv for IPv6 Redirect

Feature Name	Releases	Feature Information
IPv6: ICMPv6 Redirect	12.0(22)S 12.2(4)T 12.2(14)S 12.2(17a)SX1 12.2(25)SG 12.2(28)SB 12.2(33)SRA 15.3(1)S Cisco IOS XE Release 2.1	The IPv6 Redirect Messages feature enables a device to send ICMP IPv6 neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination. The following commands were introduced or modified: **show ipv6 interface**, **show ipv6 neighbors**, **show ipv6 route**, **show ipv6 traffic**.



CHAPTER 15

IPv6 Neighbor Discovery Cache

The IPv6 neighbor discovery cache feature allows static entries to be made in the IPv6 neighbor cache.

The per-interface neighbor discovery cache limit function can be used to prevent any particular customer attached to an interface from overloading the neighbor discovery cache, whether intentionally or unintentionally.

Information About IPv6 Static Cache Entry for Neighbor Discovery

IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring devices.

The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache. Static routing requires an administrator to manually enter IPv6 addresses, subnet masks, gateways, and corresponding Media Access Control (MAC) addresses for each interface of each device into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

Per-Interface Neighbor Discovery Cache Limit

The number of entries in the Neighbor Discovery cache can be limited by interface. Once the limit is reached, no new entries are allowed. The per-interface Neighbor Discovery cache limit function can be used to prevent any particular customer attached to an interface from overloading the Neighbor Discovery cache, whether intentionally or unintentionally.

When this feature is enabled globally, a common per-interface cache size limit is configured on all interfaces on the device. When this feature is enabled per interface, a cache size limit is configured on the associated interface. The per-interface limit overrides any globally configured limit.

How to Configure IPv6 Neighbor Discovery Cache

Configuring a Neighbor Discovery Cache Limit on a Specified Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd cache interface-limit** *size* [**log** *rate*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 nd cache interface-limit** *size* [**log** *rate*] **Example:** `Device(config-if)# ipv6 nd cache interface-limit 1`	Configures a Neighbor Discovery cache limit on a specified interface on the device. • Issuing this command overrides any configuration that may have been created by issuing the **ipv6 nd cache interface-limit** in global configuration mode.

Configuring a Neighbor Discovery Cache Limit on All Device Interfaces

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ipv6 nd cache interface-limit** *size* [**log** *rate*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd cache interface-limit** *size* [**log** *rate*] **Example:** `Device(config)# ipv6 nd cache interface-limit 4`	Configures a neighbor discovery cache limit on all interfaces on the device.

Configuration Examples for IPv6 Neighbor Discovery Cache

Example: Configuring a Neighbor Discovery Cache Limit

```
Device# show ipv6 interface GigabitEthernet2/0/0

Interface GigabitEthernet2/0/0, entries 2, static 0, limit 4

IPv6 Address               Age Link-layer Addr State  Interface
2001:0db8::94                 0 aabb.cc00.5d02    REACH  GE2/0/0
FE80::A8BB:CCFF:FE00:5D02  0 aabb.cc00.5d02    DELAY  GE2/0/0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

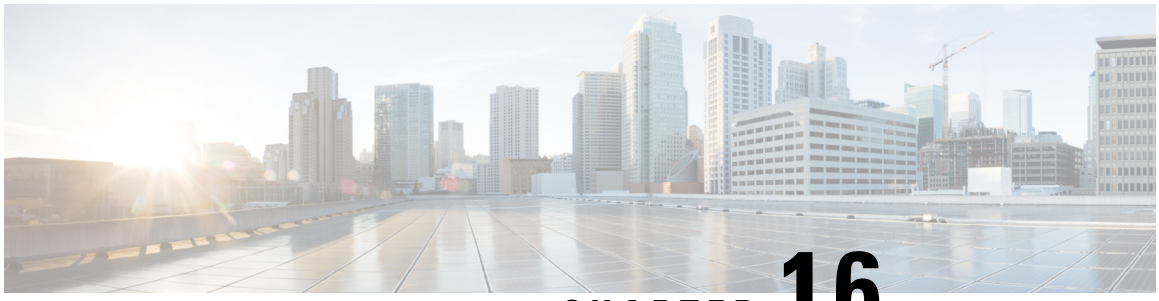
Feature Information for IPv6 Neighbor Discovery Cache

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 29: Feature Information for IPv6 Neighbor Discovery Cache

Feature Name	Releases	Feature Information
IPv6: Per-Interface Neighbor Discovery Cache Limit	15.1(1)SY 15.1(3)T Cisco IOS XE Release 2.6	The per-interface neighbor discovery cache limit function can be used to prevent any particular customer attached to an interface from overloading the neighbor discovery cache, whether intentionally or unintentionally. The following commands were introduced or modified: **ipv6 nd cache interface-limit**, **show ipv6 interface**.
IPv6 Static Cache Entry for Neighbor Discovery	12.2(8)T 12.2(17)SX1 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 15.3(1)S Cisco IOS XE Release 2.1 15.0(2)SG 3.2.0SG	The IPv6 static cache entry for Neighbor Discovery feature allows static entries to be made in the IPv6 neighbor cache. The following commands were introduced or modified: **ipv6 nd cache interface-limit**, **show ipv6 interface**.



CHAPTER 16

IPv6 Neighbor Discovery Cache

The IPv6 neighbor discovery cache feature allows static entries to be made in the IPv6 neighbor cache.

The per-interface neighbor discovery cache limit function can be used to prevent any particular customer attached to an interface from overloading the neighbor discovery cache, whether intentionally or unintentionally.

Information About IPv6 Static Cache Entry for Neighbor Discovery

IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring devices.

The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache. Static routing requires an administrator to manually enter IPv6 addresses, subnet masks, gateways, and corresponding Media Access Control (MAC) addresses for each interface of each device into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

Per-Interface Neighbor Discovery Cache Limit

The number of entries in the Neighbor Discovery cache can be limited by interface. Once the limit is reached, no new entries are allowed. The per-interface Neighbor Discovery cache limit function can be used to prevent any particular customer attached to an interface from overloading the Neighbor Discovery cache, whether intentionally or unintentionally.

When this feature is enabled globally, a common per-interface cache size limit is configured on all interfaces on the device. When this feature is enabled per interface, a cache size limit is configured on the associated interface. The per-interface limit overrides any globally configured limit.

How to Configure IPv6 Neighbor Discovery Cache

Configuring a Neighbor Discovery Cache Limit on a Specified Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd cache interface-limit** *size* [**log** *rate*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 nd cache interface-limit** *size* [**log** *rate*] **Example:** `Device(config-if)# ipv6 nd cache interface-limit 1`	Configures a Neighbor Discovery cache limit on a specified interface on the device. • Issuing this command overrides any configuration that may have been created by issuing the **ipv6 nd cache interface-limit** in global configuration mode.

Configuring a Neighbor Discovery Cache Limit on All Device Interfaces

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ipv6 nd cache interface-limit** *size* [**log** *rate*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd cache interface-limit** *size* [**log** *rate*] **Example:** `Device(config)# ipv6 nd cache interface-limit 4`	Configures a neighbor discovery cache limit on all interfaces on the device.

Configuration Examples for IPv6 Neighbor Discovery Cache

Example: Configuring a Neighbor Discovery Cache Limit

```
Device# show ipv6 interface GigabitEthernet2/0/0

Interface GigabitEthernet2/0/0, entries 2, static 0, limit 4

IPv6 Address               Age Link-layer Addr State  Interface
2001:0db8::94                 0 aabb.cc00.5d02    REACH  GE2/0/0
FE80::A8BB:CCFF:FE00:5D02  0 aabb.cc00.5d02    DELAY  GE2/0/0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

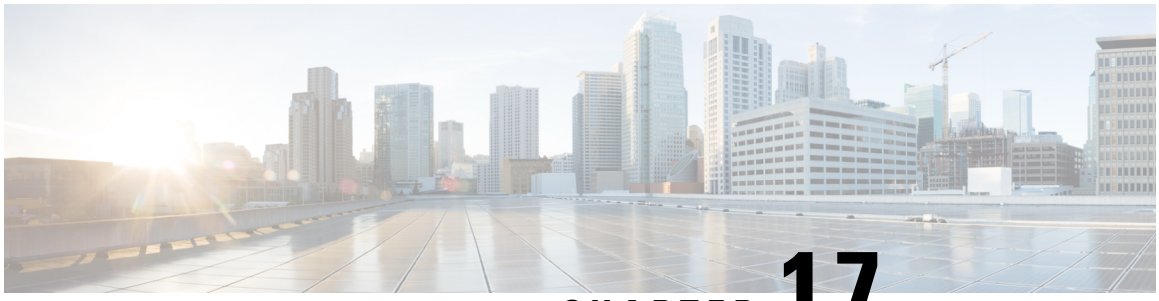
Feature Information for IPv6 Neighbor Discovery

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 30: Feature Information for IPv6 Neighbor Discovery

Feature Name	Releases	Feature Information
IPv6 Neighbor Discovery	12.0(22)S 12.2(2)T 12.2(14)S 12.2(17a)SX1 12.2(25)SG 12.2(28)SB 12.2(33)SRA Cisco IOS XE Release 2.1 12.2(50)SY 15.0(1)SY 3.2.0SG	The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring devices. The following commands were introduced or modified: **ipv6 nd cache expire**, **ipv6 nd na glean**, **ipv6 nd nud retry**.
IPv6: Neighbor Discovery Duplicate Address Detection	12.0(22)S 12.2(4)T 12.2(17a)SX1 12.2(14)S 12.2(25)SG 12.2(28)SB 12.2(33)SRA 12.2(50)SY 15.0(1)SY 15.1(1)SY 15.3(1)S Cisco IOS XE Release 2.1	IPv6 neighbor discovery duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). No commands were introduced or modified.
IPv6 Neighbor Discovery Nonstop Forwarding	12.2(33)SRE 15.0(1)S 15.0(1)SY 15.1(1)SY	The IPv6 Neighbor Discovery Nonstop Forwarding feature provides IPv6 high availability support. No commands were introduced or modified.



CHAPTER 17

IPv6 Default Router Preference

The IPv6 default router preference feature provides a coarse preference metric (low, medium, or high) for default devices.

Information About IPv6 Default Router Preference

Default Router Preferences for Traffic Engineering

Hosts discover and select default devices by listening to Router Advertisements (RAs). Typical default device selection mechanisms are suboptimal in certain cases, such as when traffic engineering is needed. For example, two devices on a link may provide equivalent but not equal-cost routing, and policy may dictate that one of the devices is preferred. Some examples are as follows:

- Multiple devices that route to distinct sets of prefixes—Redirects (sent by nonoptimal devices for a destination) mean that hosts can choose any device and the system will work. However, traffic patterns may mean that choosing one of the devices would lead to considerably fewer redirects.
- Accidentally deploying a new device—Deploying a new device before it has been fully configured could lead to hosts adopting the new device as a default device and traffic disappearing. Network managers may want to indicate that some devices are more preferred than others.
- Multihomed situations—Multihomed situations may become more common, because of multiple physical links and because of the use of tunneling for IPv6 transport. Some of the devices may not provide full default routing because they route only to the 6-to-4 prefix or they route only to a corporate intranet. These situations cannot be resolved with redirects, which operate only over a single link.

The default router preference (DRP) feature provides a basic preference metric (low, medium, or high) for default devices. The DRP of a default device is signaled in unused bits in RA messages. This extension is backward compatible, both for devices (setting the DRP bits) and hosts (interpreting the DRP bits). These bits are ignored by hosts that do not implement the DRP extension. Similarly, the values sent by devices that

do not implement the DRP extension will be interpreted by hosts that do implement it as indicating a “medium” preference. DRPs need to be configured manually.

How to Configure IPv6 Default Router Preference

Configuring the DRP Extension for Traffic Engineering

Perform this task to configure the DRP extension to RAs in order to signal the preference value of a default router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd router-preference** {**high** | **medium** | **low**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 nd router-preference** {**high**	**medium**	**low** **Example:** `Router(config-if)# ipv6 nd router-preference high`	Configures a DRP for a router on a specific interface

Configuration Examples for IPv6 Default Router Preference

Example: IPv6 Default Router Preference

The following example displays the state of the DRP preference value as advertised by this device through an interface:

```
Device# show ipv6 interface gigabitethernet 0/1

GigabitEthernet0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::130
  Description: Management network (dual stack)
  Global unicast address(es):
    FEC0:240:104:1000::130, subnet is FEC0:240:104:1000::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:130
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Low
  Hosts use stateless autoconfig for addresses.
```

The following example displays the state of the DRP preference value as advertised by other devices:

```
Device# show ipv6 routers

Router FE80::169 on GigabitEthernet0/1, last update 0 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
  Preference=Medium
  Reachable time 0 msec, Retransmit time 0 msec
  Prefix FEC0:240:104:1000::/64 onlink autoconfig
    Valid lifetime 2592000, preferred lifetime 604800
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

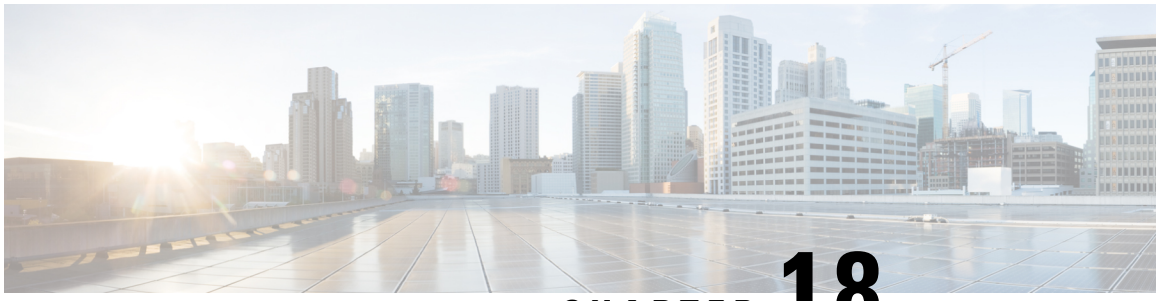
Feature Information for IPv6 Default Router Preference

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 31: Feature Information for IPv6 Default Router Preference

Feature Name	Releases	Feature Information
IPv6 Default Router Preference	Cisco IOS XE 17.4.1	This feature was introduced for the Cisco Catalyst 8000 Series platforms. This feature provides a basic preference metric (low, medium, or high) for default devices.



CHAPTER 18

IPv6 Stateless Autoconfiguration

The IPv6 stateless autoconfiguration feature can be used to manage link, subnet, and site addressing changes.

Information About IPv6 Stateless Autoconfiguration

IPv6 Stateless Autoconfiguration

All interfaces on IPv6 nodes must have a link-local address, which is usually automatically configured from the identifier for an interface and the link-local prefix FE80::/10. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

Nodes can connect to a network and automatically generate global IPv6 addresses without the need for manual configuration or help of a server, such as a Dynamic Host Configuration Protocol (DHCP) server. With IPv6, a device on the link advertises any global prefixes in Router Advertisement (RA) messages, as well as its willingness to function as a default device for the link. RA messages are sent periodically and in response to device solicitation messages, which are sent by hosts at system startup.

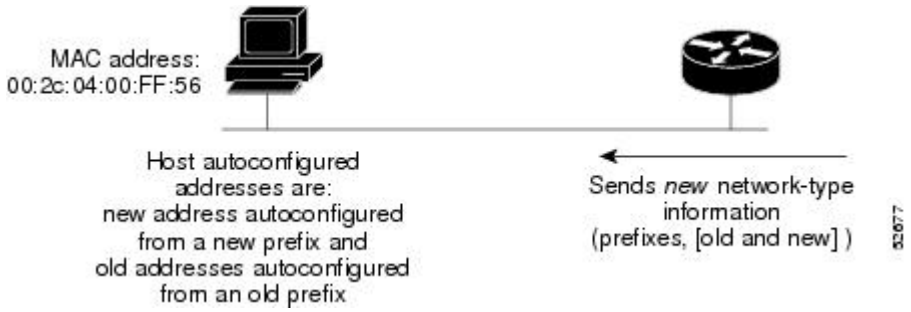
A node on the link can automatically configure global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Device solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

Simplified Network Renumbering for IPv6 Hosts

The strict aggregation of the global routing table requires that networks be renumbered when the service provider for the network is changed. When the stateless autoconfiguration functionality in IPv6 is used to renumber a network, the prefix from a new service provider is added to RA messages that are sent on the link. (The RA messages contain both the prefix from the old service provider and the prefix from the new service

provider.) Nodes on the link automatically configure additional addresses by using the prefix from the new service provider. The nodes can then use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. Configuration of the lifetime parameters associated with the old and new prefixes means that nodes on the link can make the transition to using only addresses created from the new prefix. During a transition period, the old prefix is removed from RA messages and only addresses that contain the new prefix are used on the link (the renumbering is complete) (see the figure below).

Figure 22: IPv6 Network Renumbering for Hosts Using Stateless Autoconfiguration



How to Configure IPv6 Stateless Autoconfiguration

Enabling IPv6 Stateless Autoconfiguration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address autoconfig**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies an interface type and number, and places the device in interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface GigabitEthernet 0/0/0`	
Step 4	**ipv6 address autoconfig** **Example:** `Device(config-if)# ipv6 address autoconfig`	Enables automatic configuration of IPv6 addresses using stateless autoconfiguration on an interface and enables IPv6 processing on the interface.

Configuration Examples for IPv6 Stateless Autoconfiguration

Example: Displaying IPv6 Interface Statistics

In the following example, the **show ipv6 interface** command is used to verify that IPv6 addresses are configured correctly for GigabitEthernet interface 0/0/0. Information is also displayed about the status of IPv6 neighbor redirect messages, IPv6 neighbor discovery messages, and stateless autoconfiguration.

```
Device# show ipv6 interface gigabitethernet 0/0/0

GigabitEthernet0/0/0 is up, line protocol is up
  IPv6 is stalled, link-local address is FE80::1
  Global unicast address(es):
    2001:DB8:2000::1, subnet is 2001:DB8:2000::/64
    2001:DB8:3000::1, subnet is 2001:DB8:3000::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

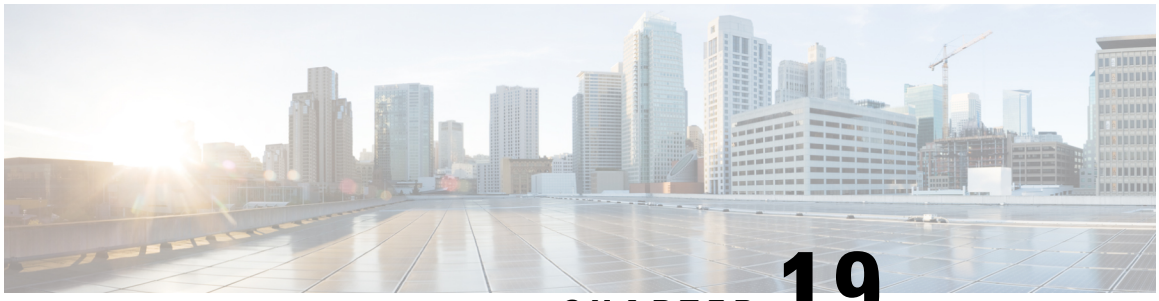
Feature Information for IPv6 Stateless Autoconfiguration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 32: Feature Information for IPv6 Stateless Autoconfiguration

Feature Name	Releases	Feature Information
IPv6 Stateless Autoconfiguration	12.2(2)T 12.2(17a)SX1 12.2(25)SEA 12.2(33)SRA 12.2(25)SG 15.0(2)SG 15.3(1)S Cisco IOS XE Release 2.1 3.2.0SG	The IPv6 stateless autoconfiguration feature can be used to manage link, subnet, and site addressing changes. The following command was introduced or modified: **ipv6 address autoconfig**.



CHAPTER 19

IPv6 RFCs

Standards and RFCs

RFCs	Title
RFC 1195	*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*
RFC 1267	*A Border Gateway Protocol 3 (BGP-3)*
RFC 1305	*Network Time Protocol (Version 3) Specification, Implementation and Analysis*
RFC 1583	*OSPF version 2*
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1886	*DNS Extensions to Support IP version 6*
RFC 1918	*Address Allocation for Private Internets*
RFC 1981	*Path MTU Discovery for IP version 6*
RFC 2080	*RIPng for IPv6*
RFC 2281	*Cisco Hot Standby Router Protocol (HSRP)*
RFC 2332	*NBMA Next Hop Resolution Protocol (NHRP)*
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2374	*An Aggregatable Global Unicast Address Format*
RFC 2375	*IPv6 Multicast Address Assignments*
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2402	*IP Authentication Header*
RFC 2404	*The Use of Hash Message Authentication Code Federal Information Processing Standard 180-1 within Encapsulating Security Payload and Authentication Header*
RFC 2406	*IP Encapsulating Security Payload (ESP)*

RFCs	Title
RFC 2407	*The Internet Security Domain of Interpretation for ISAKMP*
RFC 2408	*Internet Security Association and Key Management Protocol*
RFC 2409	*Internet Key Exchange (IKE)*
RFC 2427	*Multiprotocol Interconnect over Frame Relay*
RFC 2428	*FTP Extensions for IPv6 and NATs*
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2461	*Neighbor Discovery for IP Version 6 (IPv6)*
RFC 2462	*IPv6 Stateless Address Autoconfiguration*
RFC 2463	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 2464	*Transmission of IPv6 Packets over Ethernet*
RFC 2467	*Transmission of IPv6 Packets over FDDI*
RFC 2472	*IP Version 6 over PPP*
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*
RFC 2474	*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
RFC 2475	*An Architecture for Differentiated Services Framework*
RFC 2492	*IPv6 over ATM*
RFC 2545	*Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*
RFC 2590	*Transmission of IPv6 Packets over Frame Relay Specification*
RFC 2597	*Assured Forwarding PHB*
RFC 2598	*An Expedited Forwarding PHB*
RFC 2640	*Internet Protocol, Version 6 Specification*
RFC 2684	*Multiprotocol Encapsulation over ATM Adaptation Layer 5*
RFC 2697	*A Single Rate Three Color Marker*
RFC 2698	*A Two Rate Three Color Marker*
RFC 2710	*Multicast Listener Discovery (MLD) for IPv6*
RFC 2711	*IPv6 Router Alert Option*
RFC 2732	*Format for Literal IPv6 Addresses in URLs*

RFCs	Title
RFC 2765	*Stateless IP/ICMP Translation Algorithm (SIIT)*
RFC 2766	*Network Address Translation-Protocol Translation (NAT-PT)*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2893	*Transition Mechanisms for IPv6 Hosts and Routers*
RFC 3056	*Connection of IPv6 Domains via IPv4 Clouds*
RFC 3068	*An Anycast Prefix for 6to4 Relay Routers*
RFC 3095	*RObust Header Compression (ROHC): Framework and Four Profiles: RTP, UDP, ESP, and Uncompressed*
RFC 3107	*Carrying Label Information in BGP-4*
RFC 3137	*OSPF Stub Router Advertisement*
RFC 3147	*Generic Routing Encapsulation over CLNS*
RFC 3152	*Delegation of IP6.ARPA*
RFC 3162	*RADIUS and IPv6*
RFC 3315	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3319	*Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiated Protocol (SIP) Servers*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 3414	*User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3484	*Default Address Selection for Internet Protocol version 6 (IPv6)*
RFC 3513	*Internet Protocol Version 6 (IPv6) Addressing Architecture*
RFC 3576	*Change of Authorization*
RFC 3587	*IPv6 Global Unicast Address Format*
RFC 3590	*Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*
RFC 3596	*DNS Extensions to Support IP Version 6*
RFC 3633	*DHCP IPv6 Prefix Delegation*
RFC 3646	*DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3697	*IPv6 Flow Label Specification*
RFC 3736	*Stateless DHCP Service for IPv6*

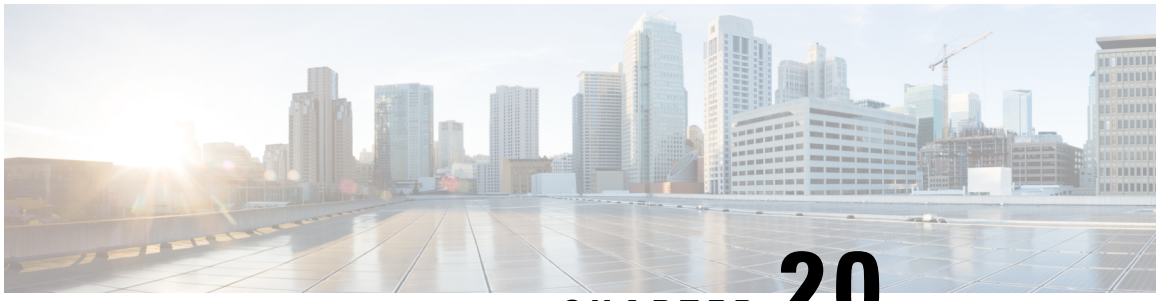
RFCs	Title
RFC 3756	*IPv6 Neighbor Discovery (ND) Trust Models and Threats*
RFC 3759	*RObust Header Compression (ROHC): Terminology and Channel Mapping Examples*
RFC 3775	*Mobility Support in IPv6*
RFC 3810	*Multicast Listener Discovery Version 2 (MLDv2) for IPv6*
RFC 3846	*Mobile IPv4 Extension for Carrying Network Access Identifiers*
RFC 3879	*Deprecating Site Local Addresses*
RFC 3898	*Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3954	*Cisco Systems NetFlow Services Export Version 9*
RFC 3956	*Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*
RFC 3963	*Network Mobility (NEMO) Basic Support Protocol*
RFC 3971	*SEcure Neighbor Discovery (SEND)*
RFC 3972	*Cryptographically Generated Addresses (CGA)*
RFC 4007	*IPv6 Scoped Address Architecture*
RFC 4075	*Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6*
RFC 4087	*IP Tunnel MIB*
RFC 4091	*The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework*
RFC 4092	*Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)*
RFC 4109	*Algorithms for Internet Key Exchange version 1 (IKEv1)*
RFC 4191	*Default Router Preferences and More-Specific Routes*
RFC 4193	*Unique Local IPv6 Unicast Addresses*
RFC 4214	*Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*
RFC 4242	*Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 4282	*The Network Access Identifier*
RFC 4283	*Mobile Node Identifier Option for Mobile IPv6*
RFC 4285	*Authentication Protocol for Mobile IPv6*
RFC 4291	*IP Version 6 Addressing Architecture*

RFCs	Title
RFC 4292	*IP Forwarding Table MIB*
RFC 4293	*Management Information Base for the Internet Protocol (IP)*
RFC 4302	*IP Authentication Header*
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
RFC 4308	*Cryptographic Suites for IPsec*
RFC 4364	*BGP MPLS/IP Virtual Private Networks (VPNs)*
RFC 4382	*MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base*
RFC 4443	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 4552	*Authentication/Confidentiality for OSPFv3*
RFC 4594	*Configuration Guidelines for DiffServ Service Classes*
RFC 4601	*Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification*
RFC 4610	*Anycast-RP Using Protocol Independent Multicast (PIM)*
RFC 4649	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option*
RFC 4659	*BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*
RFC 4724	*Graceful Restart Mechanism for BGP*
RFC 4798	*Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*
RFC 4818	*RADIUS Delegated-IPv6-Prefix Attribute*
RFC 4861	*Neighbor Discovery for IP version 6 (IPv6)*
RFC 4862	*IPv6 Stateless Address Autoconfiguration*
RFC 4884	*Extended ICMP to Support Multi-Part Messages*
RFC 4885	*Network Mobility Support Terminology*
RFC 4887	*Network Mobility Home Network Models*
RFC 5015	*Bidirectional Protocol Independent Multicast (BIDIR-PIM)*
RFC 5059	*Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*
RFC 5072	*IPv6 over PPP*
RFC 5095	*Deprecation of Type 0 Routing Headers in IPv6*
RFC 5120	*M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)*

RFCs	Title
RFC 5130	*A Policy Control Mechanism in IS-IS Using Administrative Tags*
RFC 5187	*OSPFv3 Graceful Restart*
RFC 5213	*Proxy Mobile IPv6*
RFC 5308	*Routing IPv6 with IS-IS*
RFC 5340	*OSPF for IPv6*
RFC 5460	*DHCPv6 Bulk Leasequery*
RFC 5643	*Management Information Base for OSPFv3*
RFC 5838	*Support of Address Families in OSPFv3*
RFC 5844	*IPv4 Support for Proxy Mobile IPv6*
RFC 5845	*Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6*
RFC 5846	*Binding Revocation for IPv6 Mobility*
RFC 5881	*Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)*
RFC 5905	*Network Time Protocol Version 4: Protocol and Algorithms Specification*
RFC 5969	*IPv6 Rapid Deployment on IPv4 Infrastructures (6RD) -- Protocol Specification*
RFC 6105	*IPv6 Router Advertisement Guard*
RFC 6620	*FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses*

PART III

IP Application Services



CHAPTER 20

Configuring Enhanced Object Tracking

Before the introduction of the Enhanced Object Tracking feature, the Hot Standby Router Protocol (HSRP) had a simple tracking mechanism that allowed you to track the interface line-protocol state only. If the line-protocol state of the interface went down, the HSRP priority of the router was reduced, allowing another HSRP router with a higher priority to become active.

The Enhanced Object Tracking feature separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by other Cisco IOS XE processes and HSRP. This feature allows tracking of other objects in addition to the interface line-protocol state.

A client process such as HSRP, Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP), can register its interest in tracking objects and then be notified when the tracked object changes state.

Restrictions for Enhanced Object Tracking

Enhanced Object Tracking is not stateful switchover (SSO)-aware and cannot be used with Hot Standby Routing Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) in SSO mode.

Information About Enhanced Object Tracking

Feature Design of Enhanced Object Tracking

The Enhanced Object Tracking feature provides complete separation between the objects to be tracked and the action to be taken by a client when a tracked object changes. Thus, several clients such as HSRP, VRRP,

or GLPB can register their interest with the tracking process, track the same object, and each take different action when the object changes.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes use this number to track a specific object.

The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

You can configure a combination of tracked objects in a list and a flexible method for combining objects using Boolean logic. This functionality includes the following capabilities:

- Threshold—The tracked list can be configured to use a weight or percentage threshold to measure the state of the list. Each object in a tracked list can be assigned a threshold weight. The state of the tracked list is determined by whether the threshold has been met.
- Boolean "and" function—When a tracked list has been assigned a Boolean "and" function, each object defined within a subset must be in an up state so that the tracked object can become up.
- Boolean "or" function—When the tracked list has been assigned a Boolean "or" function, at least one object defined within a subset must be in an up state so that the tracked object can become up.

With CSCtg75700, a maximum of 1000 objects can be tracked. Although 1000 tracked objects can be configured, each tracked object uses CPU resources. The amount of available CPU resources on a router depends on variables such as traffic load and how other protocols are configured and run. The ability to use 1000 tracked objects depends on the available CPU. Testing should be conducted on site to ensure that the service works under the specific site traffic conditions.

Interface State Tracking

An IP-routing object is considered up when the following criteria exist:

- IP routing is enabled and active on the interface.
- The interface line-protocol state is up.
- The interface IP address is known. The IP address is configured or received through Dynamic Host Configuration Protocol (DHCP) or IP Control Protocol (IPCP) negotiation.

Interface IP routing will go down when one of the following criteria exists:

- IP routing is disabled globally.
- The interface line-protocol state is down.
- The interface IP address is unknown. The IP address is not configured or received through DHCP or IPCP negotiation.

Tracking the IP-routing state of an interface using the **track interface ip routing** command can be more useful in some situations than just tracking the line-protocol state using the **track interface line-protocol** command, especially on interfaces for which IP addresses are negotiated. For example, on a serial interface that uses the PPP, the line protocol could be up (link control protocol [LCP] negotiated successfully), but IP could be down (IPCP negotiation failed).

The **track interface ip routing** command supports the tracking of an interface with an IP address acquired through any of the following methods:

- Conventional IP address configuration
- PPP/IPCP
- DHCP
- Unnumbered interface

You can configure Enhanced Object Tracking to consider the carrier-delay timer when tracking the IP-routing state of an interface by using the **carrier-delay** command in tracking configuration mode.

Scaled Route Metrics

The **track ip route** command enables tracking of a route in the routing table. If a route exists in the table, the metric value is converted into a number. To provide a common interface to tracking clients, normalize route metric values to the range from 0 to 255, where 0 is connected and 255 is inaccessible. Scaled metrics can be tracked by setting thresholds. Up and down state notification occurs when the thresholds are crossed. The resulting value is compared against threshold values to determine the tracking state as follows:

- State is up if the scaled metric for that route is less than or equal to the up threshold.
- State is down if the scaled metric for that route is greater than or equal to the down threshold.

Tracking uses a per-protocol configurable resolution value to convert the real metric to the scaled metric. The table below shows the default values used for the conversion. You can use the **track resolution** command to change the metric resolution default values.

Table 33: Metric Conversion

Route Type[7]	Metric Resolution
Static	10
Enhanced Interior Gateway Routing Protocol (EIGRP)	2560
Open Shortest Path First (OSPF)	1
Intermediate System-to-Intermediate System (IS-IS)	10

[7] RIP is scaled directly to the range from 0 to 255 because its maximum metric is less than 255.

For example, a change in 10 in an IS-IS metric results in a change of 1 in the scaled metric. The default resolutions are designed so that approximately one 2-Mbps link in the path will give a scaled metric of 255.

Scaling the very large metric ranges of EIGRP and IS-IS to a 0 to 255 range is a compromise. The default resolutions will cause the scaled metric to exceed the maximum limit with a 2-Mb/s link. However, this scaling allows a distinction between a route consisting of three Fast-Ethernet links and a route consisting of four Fast-Ethernet links.

IP SLA Operation Tracking

Object tracking of IP Service Level Agreements (SLAs) operations allows tracking clients to track the output from IP SLAs objects and use the provided information to trigger an action.

Cisco IOS IP SLAs is a network performance measurement and diagnostics tool that uses active monitoring. Active monitoring is the generation of traffic in a reliable and predictable manner to measure network performance. Cisco IOS XE software uses IP SLAs to collect real-time metrics such as response time, network resource availability, application performance, jitter (interpacket delay variance), connect time, throughput, and packet loss.

These metrics can be used for troubleshooting, for proactive analysis before problems occur, and for designing network topologies.

Every IP SLAs operation maintains an operation return-code value. This return code is interpreted by the tracking process. The return code can return OK, OverThreshold, and several other return codes. Different operations can have different return-code values, so only values common to all operation types are used.

Two aspects of an IP SLAs operation can be tracked: state and reachability. The difference between these aspects is the acceptance of the OverThreshold return code. The table below shows the state and reachability aspects of IP SLAs operations that can be tracked.

Table 34: Comparison of State and Reachability Operations

Tracking	Return Code	Track State
State	OK (all other return codes)	Up Down
Reachability	OK or OverThreshold (all other return codes)	Up Down

Enhanced Object Tracking and Embedded Event Manager

Enhanced Object Tracking (EOT) is now integrated with Embedded Event Manager (EEM) to allow EEM to report on status change of a tracked object and to allow EOT to track EEM objects. A new type of tracking object--a stub object--is created. The stub object can be modified by an external process through a defined Application Programming Interface (API). See the Embedded Event Manager Overview document in the *Cisco IOS XE Network Management Configuration Guide* for more information on how EOT works with EEM.

Benefits of Enhanced Object Tracking

- Increases the availability and speed of recovery of a network.
- Decreases the number of network outages and their duration.
- Enables client processes such as VRRP and GLBP to track objects individually or as a list of objects. Prior to the introduction of this functionality, the tracking process was embedded within HSRP.

How to Configure Enhanced Object Tracking

Tracking the Line-Protocol State of an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer interface** {*seconds* | **msec** *milliseconds*}
4. **track** *object-number* **interface** *type number* **line-protocol**
5. **carrier-delay**
6. **delay** {**up** *seconds* [**down** [*seconds*] | [**up** *seconds*] **down** *seconds*]}
7. **end**
8. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track timer interface** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config)# track timer interface 5`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls interface objects is 1 second. **Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the **msec** keyword and *milliseconds* argument.
Step 4	**track** *object-number* **interface** *type number* **line-protocol** **Example:** `Device(config)# track 3 interface Gigabitethernet 0/0 line-protocol`	Tracks the line-protocol state of an interface and enters tracking configuration mode.

	Command or Action	Purpose
Step 5	**carrier-delay** **Example:** `Device(config-track)# carrier-delay`	(Optional) Enables EOT to consider the carrier-delay timer when tracking the status of an interface.
Step 6	**delay** {**up** *seconds* [**down** [*seconds*] | [**up** *seconds*] **down** *seconds*]} **Example:** `Device(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 7	**end** **Example:** `Device(config-track)# end`	Exits to privileged EXEC mode.
Step 8	**show track** *object-number* **Example:** `Device# show track 3`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Example

The following example shows the state of the line protocol on an interface when it is tracked:

```
Device# show track 3

Track 3
   Interface GigabitEthernet 0/0 line-protocol
   Line protocol is Up
     1 change, last change 00:00:05
   Tracked by:
     HSRP GigabitEthernet 0/3 1
```

Tracking the IP-Routing State of an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer interface** {*seconds* | **msec** *milliseconds*}
4. **track** *object-number* **interface** *type number* **ip routing**
5. **carrier-delay**
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**
8. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track timer interface** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config)# track timer interface 5`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls interface objects is 1 second. **Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the **msec** keyword and *milliseconds* argument.
Step 4	**track** *object-number* **interface** *type number* **ip routing** **Example:** `Device(config)# track 1 interface Gigabitethernet 0/0 ip routing`	Tracks the IP-routing state of an interface and enters tracking configuration mode. • IP-route tracking tracks an IP route in the routing table and the ability of an interface to route IP packets.
Step 5	**carrier-delay** **Example:** `Device(config-track)# carrier-delay`	(Optional) Enables EOT to consider the carrier-delay timer when tracking the status of an interface.
Step 6	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 7	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.
Step 8	**show track** *object-number* **Example:** `Device# show track 1`	Displays tracking information. • Use this command to verify the configuration.

Example

The following example shows the state of IP routing on an interface when it is tracked:

```
Device# show track 1

Track 1
   Interface GigabitEthernet 0/1 ip routing
   IP routing is Up
     1 change, last change 00:01:08
   Tracked by:
     HSRP GigabitEthernet 0/3 1
```

Tracking IP-Route Reachability

Perform this task to track the reachability of an IP route. A tracked object is considered up when a routing table entry exists for the route and the route is accessible.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer ip route** {*seconds* | **msec** *milliseconds*}
4. **track** *object-number* **ip route** *ip-address/prefix-length* **reachability**
5. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
6. **ip vrf** *vrf-name*
7. **end**
8. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track timer ip route** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config)# track timer ip route 20`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls IP-route objects is 15 seconds.

	Command or Action	Purpose
		Note All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the **msec** keyword and *milliseconds* argument.
Step 4	**track** *object-number* **ip route** *ip-address*/*prefix-length* **reachability** **Example:** `Device(config)# track 4 ip route 10.16.0.0/16 reachability`	Tracks the reachability of an IP route and enters tracking configuration mode.
Step 5	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 6	**ip vrf** *vrf-name* **Example:** `Device(config-track)# ip vrf VRF2`	(Optional) Configures a VPN routing and forwarding (VRF) table.
Step 7	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.
Step 8	**show track** *object-number* **Example:** `Device# show track 4`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Example

The following example shows the state of the reachability of an IP route when it is tracked:

```
Device# show track 4

Track 4
   IP route 10.16.0.0 255.255.0.0 reachability
   Reachability is Up (RIP)
     1 change, last change 00:02:04
   First-hop interface is Ethernet0/1
   Tracked by:
     HSRP Ethernet0/3 1
```

Tracking the Threshold of IP-Route Metrics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer ip route** {*seconds* | **msec** *milliseconds*}
4. **track resolution ip route** {**eigrp** | **isis** | **ospf** | **static**} *resolution-value*
5. **track** *object-number* **ip route** *ip-address/prefix-length* **metric threshold**
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **ip vrf** *vrf-name*
8. **threshold metric** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]}
9. **end**
10. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track timer ip route** {*seconds*	**msec** *milliseconds*} **Example:** `Device(config)# track timer ip route 20`	(Optional) Specifies the interval in which the tracking process polls the tracked object. • The default interval that the tracking process polls IP-route objects is 15 seconds. **Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the **msec** keyword and *milliseconds* argument.
Step 4	**track resolution ip route** {**eigrp**	**isis**	**ospf**	**static**} *resolution-value* **Example:** `Device(config)# track resolution ip route eigrp 300`	(Optional) Specifies resolution parameters for a tracked object. • Use this command to change the default metric resolution values.

	Command or Action	Purpose
Step 5	**track** *object-number* **ip route** *ip-address/prefix-length* **metric threshold** **Example:** `Device(config)# track 6 ip route 10.16.0.0/16 metric threshold`	Tracks the scaled metric value of an IP route to determine if it is above or below a threshold and enters tracking configuration mode. • The default down value is 255, which equates to an inaccessible route. • The default up value is 254.
Step 6	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 7	**ip vrf** *vrf-name* **Example:** `Device(config-track)# ip vrf VRF1`	(Optional) Configures a VRF table.
Step 8	**threshold metric** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]} **Example:** `Device(config-track)# threshold metric up 254 down 255`	(Optional) Sets a metric threshold other than the default value.
Step 9	**end** **Example:** `Device(config-track)# end`	Exits to privileged EXEC mode.
Step 10	**show track** *object-number* **Example:** `Device# show track 6`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Example

The following example shows the metric threshold of an IP route when it is tracked:

```
Device# show track 6

Track 6
   IP route 10.16.0.0 255.255.0.0 metric threshold
   Metric threshold is Up (RIP/6/102)
     1 change, last change 00:00:08
   Metric threshold down 255 up 254
   First-hop interface is Ethernet0/1
```

```
Tracked by:
  HSRP Ethernet0/3 1
```

Tracking the State of an IP SLAs Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **ip sla** *operation-number* **state**
4. **delay** {**up** *seconds* [**down** *seconds* | [**up** *seconds*] **down** *seconds*}
5. **end**
6. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *object-number* **ip sla** *operation-number* **state** **Example:** `Device(config)# track 2 ip sla 4 state`	Tracks the state of an IP SLAs object and enters tracking configuration mode. With CScsf08092, the **track rtr** command was replaced by the **track ip sla** command.
Step 4	**delay** {**up** *seconds* [**down** *seconds* | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 60 down 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 5	**end** **Example:** `Device(config-track)# end`	Exits to privileged EXEC mode.
Step 6	**show track** *object-number* **Example:**	(Optional) Displays tracking information. • Use this command to verify the configuration.

	Command or Action	Purpose
	`Device# show track 2`	

Example

The following example shows the state of the IP SLAs tracking:

```
Device# show track 2

Track 2
   IP SLA 1 state
   State is Down
     1 change, last change 00:00:47
   Latest operation return code: over threshold
   Latest RTT (millisecs) 4
   Tracked by:
     HSRP Ethernet0/1 3
```

Tracking the Reachability of an IP SLAs IP Host

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **ip sla** *operation-number* **reachability**
4. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down***seconds*}
5. **end**
6. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *object-number* **ip sla** *operation-number* **reachability** **Example:** `Device(config)# track 2 ip sla 4 reachability`	Tracks the reachability of an IP SLAs IP host and enters tracking configuration mode. **Note** With CScsf08092, the **track rtr** command was replaced by the **track ip sla** command.

	Command or Action	Purpose
Step 4	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down***seconds*} **Example:** `Device(config-track)# delay up 30 down 10`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.
Step 5	**end** **Example:** `Device(config-track)# end`	Exits to privileged EXEC mode.
Step 6	**show track** *object-number* **Example:** `Device# show track 3`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Example

The following example shows whether the route is reachable:

```
Device# show track 3

Track 3
   IP SLA 1 reachability
   Reachability is Up
     1 change, last change 00:00:47
   Latest operation return code: over threshold
   Latest RTT (millisecs) 4
   Tracked by:
     HSRP Ethernet0/1 3
```

Configuring a Tracked List and Boolean Expression

Perform this task to configure a tracked list of objects and a Boolean expression to determine the state of the list. A tracked list contains one or more objects. The Boolean expression enables two types of calculations by using either “and” or “or” operators. For example, when you configure tracking for two interfaces using the “and” operator up means that *both* interfaces are up, and down means that either interface is down.

You may configure a tracked list state to be measured using a weight or percentage threshold. See the Configuring a Tracked List and Threshold Weight section and the Configuring a Tracked List and Threshold Percentage section.

Before you begin

An object must exist before it can be added to a tracked list.

Note The "not" operator is specified for one or more objects and negates the state of the object.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list boolean** {**and** | **or**}
4. **object** *object-number* [**not**]
5. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **list boolean** {**and**	**or**} **Example:** `Device(config)# track 100 list boolean and`	Configures a tracked list object and enters tracking configuration mode.
Step 4	**object** *object-number* [**not**] **Example:** `Device(config-track)# object 3 not`	Specifies the object to be tracked. • The*object-number* argument has a valid range from 1 to 500. There is no default. The optional **not** keyword negates the state of the object. **Note** The example means that when object 3 is up, the tracked list detects object 3 as down.
Step 5	**delay** {**up** *seconds* [**down** *seconds*]	[**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 3`	(Optional) Specifies a tracking delay in seconds between up and down states.
Step 6	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-track)# end`	

Configuring a Tracked List and Threshold Weight

Perform this task to configure a list of tracked objects, to specify that weight be used as the threshold, and to configure a weight for each of the objects in the list of tracked objects. A tracked list contains one or more objects. Enhanced object tracking uses a threshold weight to determine the state of each object by comparing the total weight of all objects that are up against a threshold weight for each object.

You can also configure a tracked list state to be measured using a Boolean calculation or threshold percentage. See the Configuring a Tracked List and Boolean Expression section and the Configuring a Tracked List and Threshold Percentage section.

Before you begin

An object must exist before it can be added to a tracked list.

Note You cannot use the Boolean "not" operator in a weight or percentage threshold list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list threshold weight**
4. **object** *object-number* [**weight** *weight-number*]
5. **threshold weight** {**up** *number* **down** *number* | **up** *number* | **down** *number*}
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **list threshold weight** **Example:**	Configures a tracked list object and enters tracking configuration mode. The keywords are as follows:

	Command or Action	Purpose
	`Device(config)# track 100 list threshold weight`	• **threshold** —Specifies that the state of the tracked list is based on a threshold. • **weight** —Specifies that the threshold is based on a specified weight.
Step 4	**object** *object-number* [**weight** *weight-number*] **Example:** `Device(config-track)# object 3 weight 30`	Specifies the object to be tracked. The *object-number* argument has a valid range from 1 to 500. There is no default. The optional **weight** keyword specifies a threshold weight for each object.
Step 5	**threshold weight** {**up** *number* **down** *number* | **up** *number* | **down** *number*} **Example:** `Device(config-track)# threshold weight up 30`	Specifies the threshold weight. • **up** *number* —Valid range is from 1 to 255. • **down** *number*—Range depends upon what you select for the **up** keyword. For example, if you configure 25 for up, you will see a range from 0 to 24 for down.
Step 6	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 3`	(Optional) Specifies a tracking delay in seconds between up and down states.
Step 7	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.

Configuring a Tracked List and Threshold Percentage

Perform this task to configure a tracked list of objects, to specify that a percentage will be used as the threshold, and to specify a percentage for each object in the list. A tracked list contains one or more objects. Enhanced object tracking uses the threshold percentage to determine the state of the list by comparing the assigned percentage of each object to the list.

You may also configure a tracked list state to be measured using a Boolean calculation or threshold weight. See the Configuring a Tracked List and Boolean Expression section and theConfiguring a Tracked List and Threshold Weight section.

Note You cannot use the Boolean “not” operator in a weight or percentage threshold list.

Before you begin

An object must exist before it can be added to a tracked list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list threshold percentage**
4. **object** *object-number*
5. **threshold percentage** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]}
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **list threshold percentage** **Example:** `Device(config)# track 100 list threshold percentage`	Configures a tracked list object and enters tracking configuration mode. The keywords are as follows: • **threshold** —Specifies that the state of the tracked list is based on a threshold. • **percentage** —Specifies that the threshold is based on a percentage.
Step 4	**object** *object-number* **Example:** `Device(config-track)# object 3`	Specifies the object to be tracked. • The *object-number* argument has a valid range from 1 to 500. There is no default.
Step 5	**threshold percentage** {**up** *number* [**down** *number*]	**down** *number* [**up** *number*]} **Example:** `Device(config-track)# threshold percentage up 30`	Specifies the threshold percentage. • **up** *number*—Valid range is from 1 to 100. • **down** *number* —Range depends upon what you have selected for the **up** keyword. For example, if you specify 25 as up, a range from 26 to 100 is displayed for the **down** keyword.
Step 6	**delay** {**up** *seconds* [**down** *seconds*]	[**up** *seconds*] **down** *seconds*} **Example:**	(Optional) Specifies a tracking delay in seconds between up and down states.

	Command or Action	Purpose
	`Device(config-track)# delay up 3`	
Step 7	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.

Configuring Track List Defaults

Perform this task to configure a default delay value for a tracked list, a default object, and default threshold parameters for a tracked list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number*
4. **default** {**delay** | **object** *object-number* | **threshold percentage**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **Example:** `Device(config)# track 3`	Enters tracking configuration mode.
Step 4	**default** {**delay**	**object** *object-number*	**threshold percentage**} **Example:** `Device(config-track)# default delay`	Specifies a default delay value for a tracked list, a default object, and default threshold parameters for a tracked list. • **delay** —Reverts to the default delay. • **object** *object-number*—Specifies a default object for the track list. The valid range is from 1 to 1000.

	Command or Action	Purpose
		• **threshold percentage**—Specifies a default threshold percentage.
Step 5	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.

Configuring Tracking for Mobile IP Applications

Perform this task to configure a tracked list of Mobile IP application objects.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **application home-agent**
4. **exit**
5. **track** *track-number* **application pdsn**
6. **exit**
7. **track** *track-number* **application ggsn**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *track-number* **application home-agent** **Example:** `Device(config)# track 100 application home-agent`	(Optional) Tracks the presence of Home Agent traffic on a router and enters tracking configuration mode.
Step 4	**exit** **Example:** `Device(config-track)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 5	**track** *track-number* **application pdsn** **Example:** `Device(config)# track 100 application pdsn`	(Optional) Tracks the presence of Packet Data Serving Node (PDSN) traffic on a router tracking configuration mode.
Step 6	**exit** **Example:** `Device(config-track)# exit`	Returns to global configuration mode.
Step 7	**track** *track-number* **application ggsn** **Example:** `Device(config)# track 100 application ggsn`	(Optional) Tracks the presence of Gateway GPRS Support Node (GGSN) traffic on a router tracking configuration mode.
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuration Examples for Enhanced Object Tracking

Example: Interface Line Protocol

In the following example, the tracking process is configured to track the line-protocol state of GigabitEthernet interface 1/0/0. HSRP on GigabitEthernet interface 0/0/0 then registers with the tracking process to be informed of any changes to the line-protocol state of GigabitEthernet interface 1/0/0. If the line protocol on GigabitEthernet interface 1/0/0 goes down, the priority of the HSRP group is reduced by 10.

Router A Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 line-protocol
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 line-protocol
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.22 255.255.0.0
```

```
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: Interface IP Routing

In the following example, the tracking process is configured to track the IP-routing capability of GigabitEthernet interface 1/0/0. HSRP on GigabitEthernet interface 0/0/0 then registers with the tracking process to be informed of any changes to the IP-routing state of GigabitEthernet interface 1/0/0. If the IP-routing state on GigabitEthernet interface 1/0/0 goes down, the priority of the HSRP group is reduced by 10.

If both serial interfaces are operational, Router A will be the HSRP active router because it has the higher priority. However, if IP on GigabitEthernet interface 1/0/0 in Router A fails, the HSRP group priority will be reduced and Router B will take over as the active router, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

See the figure below for a sample topology.

Figure 23: Topology for IP-Routing Support

Router A Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 ip routing
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 interface GigabitEthernet1/0/0 ip routing
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.0.22 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: IP-Route Reachability

In the following example, the tracking process is configured to track the reachability of IP route 10.2.2.0/24:

Router A Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 reachability
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.21 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 reachability
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.22 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: IP-Route Threshold Metric

In the following example, the tracking process is configured to track the threshold metric of IP route 10.2.2.0/24:

Router A Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 metric threshold
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.21 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
```

Router B Configuration

```
Device(config)# track 100 ip route 10.2.2.0/24 metric threshold
!
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.22 255.255.255.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.1.1
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
```

Example: IP SLAs IP Host Tracking

The following example shows how to configure IP host tracking for IP SLAs operation 1 prior to CSCsf08092:

```
Device(config)# ip sla 1
Device(config-ip-sla)# icmp-echo 10.51.12.4
```

```
Device(config-ip-sla-echo)# timeout 1000
Device(config-ip-sla-echo)# threshold 2
Device(config-ip-sla-echo)# frequency 3
Device(config-ip-sla-echo)# request-data-size 1400
Device(config-ip-sla-echo)# exit
Device(config)# ip sla schedule 1 start-time now life forever
Device(config-ip-sla)# track 2 rtr 1 state
Device(config-ip-sla)# exit
Device(config)# track 3 rtr 1 reachability
Device(config-track)# exit
Device(config)# interface ethernet0/1
Device(config-if)# ip address 10.21.0.4 255.255.0.0
Device(config-if)# no shutdown
Device(config-if)# standby 3 ip 10.21.0.10
Device(config-if)# standby 3 priority 120
Device(config-if)# standby 3 preempt
Device(config-if)# standby 3 track 2 decrement 10
Device(config-if)# standby 3 track 3 decrement 10
```

The following example shows how to configure IP host tracking for IP SLAs operation 1 prior to CSCsf08092:

```
Device(config)# ip sla 1
Device(config-ip-sla)# icmp-echo 10.51.12.4
Device(config-ip-sla-echo)# threshold 2
Device(config-ip-sla-echo)# timeout 1000
Device(config-ip-sla-echo)# frequency 3
Device(config-ip-sla-echo)# request-data-size 1400
Device(config-ip-sla-echo)# exit
Device(config)# ip sla schedule 1 start-time now life forever
Device(config)# track 2 ip sla 1 state
Device(config-track)# exit
Device(config)# track 3 ip sla 1 reachability
Device(config-track)# exit
Device(config)# interface ethernet0/1
Device(config-if)# ip address 10.21.0.4 255.255.0.0
Device(config-if)# no shutdown
Device(config-if)# standby 3 ip 10.21.0.10
Device(config-if)# standby 3 priority 120
Device(config-if)# standby 3 preempt
Device(config-if)# standby 3 track 2 decrement 10
Device(config-if)# standby 3 track 3 decrement 10
```

Example: Boolean Expression for a Tracked List

In the following example, a track list object is configured to track two GigabitEthernet interfaces when both interfaces are up and when either interface is down:

```
Device(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
Device(config-track)# exit
Device(config)# track 100 list boolean and
Device(config-track)# object 1
Device(config-track)# object 2
```

In the following example, a track list object is configured to track two GigabitEthernet interfaces when either interface is up and when both interfaces are down:

```
Device(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
```

```
Device(config-track)# exit
Device(config)# track 101 list boolean or
Device(config-track)# object 1
Device(config-track)# object 2
```

The following configuration example shows that tracked list 4 has two objects and one object state is negated (if the list is up, the list detects that object 2 is down):

```
Device(config)# track 4 list boolean and
Device(config-track)# object 1
Device(config-track)# object 2 not
```

Example: Threshold Weight for a Tracked List

In the following example, three GigabitEtherent interfaces in tracked list 100 are configured with a threshold weight of 20 each. The down threshold is configured to 0 and the up threshold is configured to 40:

```
Device(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
Device(config)# track 3 interface GigabitEthernet2/2/0 line-protocol
Device(config-track)# exit
Device(config)# track 100 list threshold weight
Device(config-track)# object 1 weight 20
Device(config-track)# object 2 weight 20
Device(config-track)# object 3 weight 20
Device(config-track)# threshold weight  up 40 down 0
```

In the example above the track-list object goes down only when all three serial interfaces go down, and comes up again only when at least two interfaces are up (because 20 + 20 >= 40). The advantage of this configuration is that it prevents the track-list object from coming up if two interfaces are down and the third interface is flapping.

The following configuration example shows that if object 1 and object 2 are down, then track list 4 is up, because object 3 satisfies the up threshold value of up 30. But, if object 3 is down, both objects 1 and 2 need to be up in order to satisfy the threshold weight.

```
Device(config)# track 4 list threshold weight
Device(config-track)# object 1 weight 15
Device(config-track)# object 2 weight 20
Device(config-track)# object 3 weight 30
Device(config-track)# threshold weight up 30 down 10
```

This configuration may be useful to you if you have two small bandwidth connections (represented by object 1 and 2) and one large bandwidth connection (represented by object 3). Also the down 10 value means that once the tracked object is up, it will not go down until the threshold value is lower or equal to 10, which in this example means that all connections are down.

Example: Threshold Percentage for a Tracked List

In the following example, four GigabitEthernet interfaces in track list 100 are configured for an up threshold percentage of 75. The track list is up when 75 percent of the interfaces are up and down when fewer than 75 percent of the interfaces are up.

```
Device(config)# track 1 interface GigabitEthernet2/0/0 line-protocol
Device(config)# track 2 interface GigabitEthernet2/1/0 line-protocol
```

```
Device(config)# track 3 interface GigabitEthernet2/2/0 line-protocol
Device(config)# track 4 interface GigabitEthernet2/3/0 line-protocol
Device(config-track)# exit
Device(config)# track 100 list threshold percentage
Device(config-track)# object 1
Device(config-track)# object 2
Device(config-track)# object 3
Device(config-track)# object 4
Device(config-track)# threshold percentage up 75
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Embedded Event Manager	*Embedded Event Manager Overview*
HSRP concepts and configuration tasks	*Configuring HSRP*
GLBP concepts and configuration tasks	*Configuring GLBP*
IP SLAs commands	*Cisco IOS IP SLAs Command Reference*
VRRP concepts and configuration tasks	*Configuring VRRP*
GLBP, HSRP, and VRRP commands	*Cisco IOS IP Application Services Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Enhanced Object Tracking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 35: Feature Information for Enhanced Object Tracking

Feature Name	Releases	Feature Information
Enhanced Tracking Support	15.0(1)SY	The Enhanced Tracking Support feature separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by other Cisco IOS processes and HSRP. This feature allows tracking of other objects in addition to the interface line-protocol state. The following commands were introduced or modified: **show track**, **standby track**, **threshold metric**, **track interface**, **track ip route**, **track timer**.

Feature Name	Releases	Feature Information
FHRP—Enhanced Object Tracking Integration with Embedded Event Manager	15.0(1)SY	EOT is integrated with Embedded Event Manager (EEM) to allow EEM to report on a status change of a tracked object and to allow EOT to track EEM objects. The following commands were introduced or modified by this feature: **default-state**, **event resource**, **event rf**, **event track**, **show track**, **track stub**.
FHRP—Enhanced Object Tracking of IP SLAs Operations	15.0(1)SY	This feature enables First Hop Redundancy Protocols (FHRPs) and other Enhanced Object Tracking (EOT) clients to track the output from IP SLAs objects and use the provided information to trigger an action. The following command was introduced by this feature: **track rtr**.
FHRP—EOT Deprecation of rtr Keyword	15.0(1)SY	This feature replaces the **track rtr** command with the **track ip sla** command.
FHRP—Object Tracking List	15.0(1)SY	This feature enhances the tracking capabilities to enable the configuration of a combination of tracked objects in a list, and a flexible method of combining objects using Boolean logic. The following commands were introduced or modified by this feature: **show track**, **threshold percentage**, **threshold weight**, **track list**, **track resolution**.

Glossary

DHCP—Dynamic Host Configuration Protocol. DHCP is a protocol that delivers IP addresses and configuration information to network clients.

GGSN—Gateway GPRS Support Node. A wireless gateway that allows mobile cell phone users to access the public data network (PDN) or specified private IP networks. The GGSN function is implemented on the Cisco routers.

GLBP—Gateway Load Balancing Protocol. Provides automatic router backup for IP hosts that are configured with a single default gateway on an IEEE 802.3 LAN. Multiple first-hop routers on the LAN combine to offer a single virtual first-hop IP router while sharing the IP packet forwarding load. Other routers on the LAN may act as redundant (GLBP) routers that will become active if any of the existing forwarding routers fail.

GPRS—General Packet Radio Service. A 2.5G mobile communications technology that enables mobile wireless service providers to offer their mobile subscribers with packet-based data services over GSM networks.

GSM network—Global System for Mobile Communications network. A digital cellular technology that is used worldwide, predominantly in Europe and Asia. GSM is the world's leading standard in digital wireless communications.

Home Agent—A Home Agent is a router on the home network of the Mobile Node (MN) that maintains an association between the home IP address of the MN and its care-of address, which is the current location of the MN on a foreign or visited network. The HA redirects packets by tunneling them to the MN while it is away from the home network.

HSRP—Hot Standby Router Protocol. Provides high network availability and transparent network topology changes. HSRP creates a Hot Standby router group with a lead router that services all packets sent to the Hot Standby address. The lead router is monitored by other routers in the group, and if it fails, one of these standby routers inherits the lead position and the Hot Standby group address.

IPCP—IP Control Protocol. The protocol used to establish and configure IP over PPP.

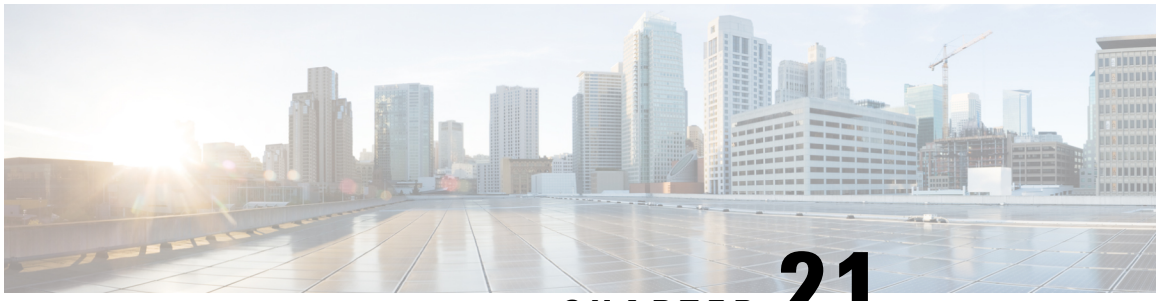
LCP—Link Control Protocol. The protocol used to establish, configure, and test data-link connections for use by PPP.

PDSN—Packet Data Serving Node. The Cisco PDSN is a standards-compliant, wireless gateway that enables packet data services in a Code Division Multiplex Access (CDMA) environment. Acting as an access gateway, the Cisco PDSN provides simple IP and Mobile IP access, foreign-agent support, and packet transport for Virtual Private Networks (VPN).

PPP—Point-to-Point Protocol. Provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. PPP is most commonly used for dial-up Internet access. Its features include address notification, authentication via CHAP or PAP, support for multiple protocols, and link monitoring.

VRF—VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge router.

VRRP—Virtual Router Redundancy Protocol. Eliminates the single point of failure inherent in the static default routed environment. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router that controls the IP addresses associated with a virtual router is called the primary, and forwards packets sent to these IP addresses. The election process provides dynamic failover in the forwarding responsibility should the primary become unavailable. Any of the virtual router IP addresses on a LAN can then be used as the default first-hop router by end hosts.



CHAPTER 21

Configuring IP Services

This module describes how to configure optional IP services. For a complete description of the IP services commands in this chapter, refer to the *Cisco IOS IP Application Services Command Reference*. To locate documentation of other commands that appear in this module, use the master command list, or search online.

Information About IP Services

IP Source Routing

The Cisco IOS XE software examines IP header options on every packet. It supports the IP header options Strict Source Route, Loose Source Route, Record Route, and Time Stamp, which are defined in RFC 791. If the software finds a packet with one of these options enabled, it performs the appropriate action. If it finds a packet with an invalid option, it sends an Internet Control Message Protocol (ICMP) parameter problem message to the source of the packet and discards the packet.

IP provides a provision known as source routing that allows the source IP host to specify a route through the IP network. Source routing is specified as an option in the IP header. If source routing is specified, the software forwards the packet according to the specified source route. IP source routing is employed when you want to force a packet to take a certain route through the network. The default is to perform source routing. IP source routing is rarely used for legitimate purposes in networks. Some older IP implementations do not process source-routed packets properly, and it may be possible to crash devices running these implementations by sending them datagrams with source routing options. Disable IP source routing whenever possible. Disabling IP source routing will cause a Cisco router to never forward an IP packet that carries a source routing option.

Note From Cisco IOS XE Release 17.1.1, IP source routing is disabled by default.

ICMP Overview

Originally created for the TCP/IP suite in RFC 792, the Internet Control Message Protocol (ICMP) was designed to report a small set of error conditions. ICMP can also report a wide variety of error conditions and provide feedback and testing capabilities. Each message uses a common format and is sent and received by using the same protocol rules.

ICMP enables IP to perform addressing, datagram packaging, and routing by allowing encapsulated messages to be sent and received between IP devices. These messages are encapsulated in IP datagrams just like any other IP message. When the message is generated, the original IP header is encapsulated in the ICMP message and these two pieces are encapsulated within a new IP header to be returned as an error report to the sending device.

ICMP messages are sent in several situations: when a datagram cannot reach its destination, when the gateway does not have the buffering capacity to forward a datagram, and when the gateway can direct the host to send traffic on a shorter route. To avoid the infinite regress of messages about messages, no ICMP messages are sent about ICMP messages.

ICMP does not make IP reliable or ensure the delivery of datagrams or the return of a control message. Some datagrams may be dropped without any report of their loss. The higher-level protocols that use IP must implement their own reliability procedures if reliable communication is required.

ICMP Unreachable Error Messages

Type 3 error messages are sent when a message cannot be delivered completely to the application at a destination host. Six codes contained in the ICMP header describe the unreachable condition as follows:

- 0—Network unreachable
- 1—Host unreachable
- 2—Protocol unreachable
- 3—Port unreachable
- 4—Fragmentation needed and the “don’t fragment﷯? (DF) bit is set
- 5—Source route failed

Cisco IOS XE software can suppress the generation of ICMP unreachable destination error messages, which is called rate-limiting. The default is no unreachable messages more often than once every half second. Separate intervals can be configured for code 4 and all other unreachable destination error messages. However, there is no method of displaying how many ICMP messages have not been sent.

The ICMP Unreachable Destination Counters feature provides a method to count and display the unsent Type 3 messages. This feature also provides console logging with error messages when there are periods of excessive rate limiting that would indicate a Denial of Service (DoS) attack against the router.

If the Cisco IOS XE software receives a nonbroadcast packet destined for itself that uses an unknown protocol, it sends an ICMP protocol unreachable message back to the source. Similarly, if the software receives a packet that it is unable to deliver to the final destination because it knows of no route to the destination address, it sends an ICMP host unreachable message to the source. This functionality is enabled by default.

Disable ICMP host unreachable messages whenever possible. ICMP supports IP traffic by relaying information about paths, routes, and network conditions. These messages can be used by an attacker to gain network mapping information.

Because the null interface is a packet sink, packets forwarded there will always be discarded and, unless disabled, will generate host unreachable messages. In that case, if the null interface is being used to block a Denial-of-Service attack, these messages flood the local network with these messages. Disabling these messages prevents this situation. In addition, because all blocked packets are forwarded to the null interface, an attacker receiving host unreachable messages could use those messages to determine Access Control List (ACL) configuration. If the "null 0  ? interface is configured on your router, disable ICMP host unreachable messages for discarded packets or packets routed to the null interface.

ICMP Mask Reply Messages

Occasionally, network devices must know the subnet mask for a particular subnetwork in the internetwork. To obtain this information, such devices can send ICMP mask request messages. ICMP mask reply messages are sent in reply from devices that have the requested information. The Cisco IOS XE software can respond to ICMP mask request messages if this function is enabled.

These messages can be used by an attacker to gain network mapping information.

ICMP Redirect Messages

Routes are sometimes less than optimal. For example, it is possible for the router to be forced to resend a packet through the same interface on which it was received. If the router resends a packet through the same interface on which it was received, the Cisco IOS XE software sends an ICMP redirect message to the originator of the packet telling the originator that the router is on a subnet directly connected to the receiving device, and that it must forward the packet to another system on the same subnet. The software sends an ICMP redirect message to the originator of the packet because the originating host presumably could have sent that packet to the next hop without involving this device at all. The redirect message instructs the sender to remove the receiving device from the route and substitute a specified device representing a more direct path. This functionality is enabled by default.

In a properly functioning IP network, a router will send redirects only to hosts on its own local subnets, no end node will ever send a redirect, and no redirect will ever be traversed more than one network hop. However, an attacker may violate these rules; some attacks are based on this. Disabling ICMP redirects will cause no operational impact to the network, and it eliminates this possible method of attack.

Denial of Service Attack

Denial of service has become a growing concern, especially when considering the associated costs of such an attack. DoS attacks can decrease the performance of networked devices, disconnect the devices from the network, and cause system crashes. When network services are unavailable, enterprises and service providers suffer the loss of productivity and sales.

The objective of a DoS attack is to deprive a user or organization access to services or resources. If a Website is compromised by a DoS attack, millions of users could be denied access to the site. DoS attacks do not typically result in intrusion or the illegal theft of information. Instead of providing access to unauthorized users, DoS attacks can cause much aggravation and cost to the target customer by preventing authorized access. Distributed DoS (DDoS) attacks amplify DoS attacks in that a multitude of compromised systems coordinate to flood targets with attack packets, thereby causing denial of service for users of the targeted systems.

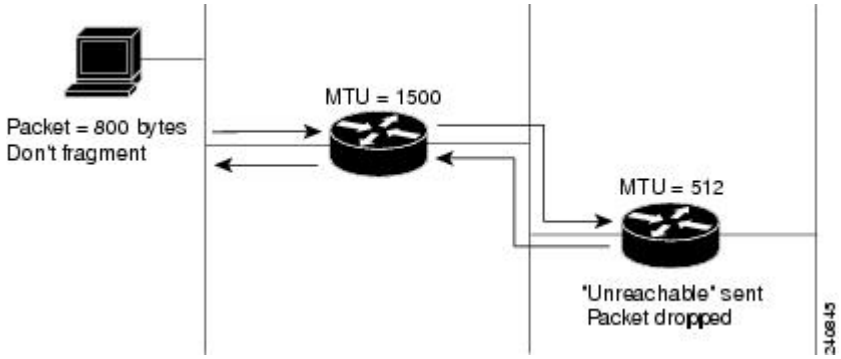
A DoS attack occurs when a stream of ICMP echo requests (pings) are broadcast to a destination subnet. The source addresses of these requests are falsified to be the source address of the target. For each request sent by

the attacker, many hosts on the subnet will respond flooding the target and wasting bandwidth. The most common DoS attack is called a “smurf” attack, named after an executable program and is in the category of network-level attacks against hosts. DoS attacks can be easily detected when error-message logging of the ICMP Unreachable Destination Counters feature is enabled.

Path MTU Discovery

The Cisco IOS XE software supports the IP Path MTU Discovery mechanism, as defined in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a router is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU you set for the interface with the **ip mtu** interface configuration command), but the “don’t fragment?? (DF) bit is set. The Cisco IOS XE software sends a message to the sending host, alerting it to the problem. The host will need to fragment packets for the destination so that they fit the smallest packet size of all the links along the path. This technique is shown in the figure below.

Figure 24: IP Path MTU Discovery



IP Path MTU Discovery is useful when a link in a network goes down, forcing the use of another, different MTU-sized link (and different routers). As shown in the figure above, suppose a router is sending IP packets over a network where the MTU in the first router is set to 1500 bytes, but the second router is set to 512 bytes. If the “don’t fragment?? bit of the datagram is set, the datagram would be dropped because the 512-byte router is unable to forward it. All packets larger than 512 bytes are dropped in this case. The second router returns an ICMP destination unreachable message to the source of the datagram with its Code field indicating “Fragmentation needed and DF set.?? To support IP Path MTU Discovery, it would also include the MTU of the next hop network link in the low-order bits of an unused header field.

IP Path MTU Discovery is also useful when a connection is being established and the sender has no information at all about the intervening links. It is always advisable to use the largest MTU that the links will bear; the larger the MTU, the fewer packets the host must send.

Note IP Path MTU Discovery is a process initiated by end hosts. If an end host does not support IP Path MTU Discovery, the receiving device will have no mechanism to avoid fragmenting datagrams generated by the end host.

If a router that is configured with a small MTU on an outbound interface receives packets from a host that is configured with a large MTU (for example, receiving packets from a Token Ring interface and forwarding them to an outbound Ethernet interface), the router fragments received packets that are larger than the MTU

of the outbound interface. Fragmenting packets slows the performance of the router. To keep routers in your network from fragmenting received packets, run IP Path MTU Discovery on all hosts and routers in your network, and always configure the largest possible MTU for each router interface type.

Show and Clear Commands for IOS Sockets

The Show and Clear Commands for IOS Sockets feature introduces the **show udp**, **show sockets**, and **clear sockets**commands. These new commands are useful for monitoring and managing the Cisco IOS Socket library.

In Cisco IOS software, sockets are a per process entity. This means that the maximum number of sockets is per process and all sockets are managed on a per process basis. For example, each Cisco IOS process could have a socket with file descriptor number 1. This is unlike UNIX or other operating systems that have per system file descriptor allocations.

The **show**and **clear** commands operate on a per process basis to be consistent with the current functionality. Thus, any action taken by the commands will be applicable only to a particular process at a time as selected by the process ID entered on the CLI.

Many applications have a need for **show** and **clear**commands, which primarily aid in debugging. The following scenarios provide examples of when these commands might be useful:

- The application H.323 is using sockets for voice calls. According to the current number of calls, there is still space for more sockets. However, no more sockets can be opened. You can now use the the **show sockets** command to find out if the socket space is indeed exhausted or if there are unused sockets available.
- An application is waiting for a particular socket event to happen. A UDP segment was seen, but the application never became active. You can use the **show udp** command to display the list of events being monitored to determine if a UDP socket event is being monitored or if the socket library failed to activate the application.
- An application wants to forcibly close all the sockets for a particular process. You can use the **clear sockets** command to close both the sockets and the underlying TCP or UDP connection or Stream Control Transmission Protocol (SCTP) association.

How to Configure IP Services

Protecting Your Network from DOS Attacks

ICMP supports IP traffic by relaying information about paths, routes, and network conditions. ICMP messages can be used by an attacker to gain network mapping information. IP source routing allows the source IP host to specify a route through the IP network and is rarely used for legitimate purposes in networks. Some older IP implementations do not process source-routed packets properly, and it may be possible to crash devices running these implementations by sending them datagrams with source routing options.

Whenever possible, ICMP messages and IP source routing should be disabled.

Note From Cisco IOS XE Release 17.1.1, IP source routing is disabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip source-route**
4. **interface** *type/number/slot*
5. **no ip unreachables**
6. **no ip redirects**
7. **no ip mask-reply**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip source-route** **Example:** `Device(config)# no ip source-route`	Disables IP source routing. **Note** From Cisco IOS XE Release 17.1.1, IP source routing is disabled by default.
Step 4	**interface** *type/number/slot* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface to configure and enters interface configuration mode.
Step 5	**no ip unreachables** **Example:** `Device(config-if)# no ip unreachables`	Disables the sending of ICMP protocol unreachable and host unreachable messages. This command is enabled by default. **Note** Disabling the unreachable messages also disables IP Path MTU Discovery because path discovery works by having the Cisco IOS XE software send unreachable messages.
Step 6	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables the sending of ICMP redirect messages to learn routes. This command is enabled by default.
Step 7	**no ip mask-reply** **Example:**	Disables the sending of ICMP mask reply messages.

	Command or Action	Purpose
	`Device(config-if)# no ip mask-reply`	

Configuring ICMP Unreachable Rate Limiting User Feedback

Perform this task to clear all of the unreachable destination packet statistics and to specify an interval number for unreachable destination messages. This task also configures a packet counter (threshold) and interval to trigger a logging message to a console. This task is beneficial to begin a new log after the thresholds have been set.

SUMMARY STEPS

1. **enable**
2. **clear ip icmp rate-limit** [*interface-type interface-number*]
3. **configure terminal**
4. **ip icmp rate-limit unreachable** [**df**] [*ms*] [**log** [*packets*] [*interval-ms*]]
5. **exit**
6. **show ip icmp rate-limit** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip icmp rate-limit** [*interface-type interface-number*] **Example:** `Router# clear ip icmp rate-limit ethernet 2/3`	Clears all current ICMP unreachable statistics for all configured interfaces. The optional *interface-type* and *interface-number* arguments clear the statistics for only one interface.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**ip icmp rate-limit unreachable** [**df**] [*ms*] [**log** [*packets*] [*interval-ms*]] **Example:** `Router(config)# ip icmp rate-limit unreachable df log 1100 12000`	Specifies the rate limitation of ICMP unreachable destination messages and the error message log threshold for generating a message. The default is no unreachable messages are sent more often than once every half second. The arguments and keywords are as follows: • **df** --(Optional) When "don't fragment" (DF) bit is set in the ICMP header, a datagram cannot be fragmented. If the **df** keyword is not specified, all other types of destination unreachable messages are sent.

	Command or Action	Purpose

		• *ms* --(Optional) Interval at which unreachable messages are generated. The valid range is from 1 to 4294967295.
		• **log** --(Optional) List of error messages. The arguments are as follows:
		  • *packets*--(Optional) Number of packets that determine a threshold for generating a log. The default is 1000.
		  • *interval-ms*--(Optional) Time limit for an interval for which a logging message is triggered. The default is 60000, which is 1 minute.
		Note Counting begins as soon as this command is configured.

Step 5	**exit** **Example:** `Router# exit`	Exits to privileged EXEC mode.
Step 6	**show ip icmp rate-limit** [*interface-type interface-number*] **Example:** `Router# show ip icmp rate-limit ethernet 2/3`	(Optional) Displays all current ICMP unreachable statistics for all configured interfaces. The optional *interface-type* and *interface-number* arguments display the statistics for only one interface.

Example

The following output using the**show ip icmp rate-limit** command displays the unreachable destinations by interface:

```
Router# show ip icmp rate-limit
                          DF bit unreachables       All other unreachables
Interval (millisecond)    500                       500
Interface                 # DF bit unreachables     # All other unreachables
---------                 ---------------------     ------------------------
Ethernet0/0               0                         0
Ethernet0/2               0                         0
Serial3/0/3               0                         19
The greatest number of unreachables is on serial interface 3/0/3.
```

Setting the MTU Packet Size

All interfaces have a default MTU packet size. You can adjust the IP MTU size so that the Cisco IOS XE software will fragment any IP packet that exceeds the MTU set for an interface.

Changing the MTU value (with the **mtu** interface configuration command) can affect the IP MTU value. If the current IP MTU value is the same as the MTU value and you change the MTU value, the IP MTU value

will be modified automatically to match the new MTU. However, the reverse is not true; changing the IP MTU value has no effect on the value for the **mtu** interface configuration command.

All devices on a physical medium must have the same protocol MTU in order to operate.

Perform this task to set the MTU packet size for a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/number/slot*
4. **ip mtu** *bytes*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type/number/slot* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**ip mtu** *bytes* **Example:** `Device(config-if)# ip mtu 300`	Sets the IP MTU packet size for an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring IP Accounting With NetFlow

IP Accounting collects the number of bytes and packets processed by the network element based on the source or destination IP address, or the configured IP precedence. The information collected can be used to identify users for network usage billing, monitoring, and troubleshooting.

Cisco ASR 1000 Series Aggregation Services Routers do not support the IP Accounting feature; however, support Flexible Netflow as the recommended method to collect network information. For more information on Flexible NetFlow configuration see the Flexible NetFlow Configuration Guide.

The following steps are performed in this task:

1. Create a flow record based on the IP address and define the counters to be collected.
2. Create a flow record based on IP precedence and define the counters to be collected.
3. Create a flow monitor, define the monitor parameters, and link it with the IP address-based flow record.
4. Create a flow monitor, define the monitor parameters, and link it with IP precedence-based flow record.
5. Attach the IP address-based flow monitor and IP precedence-based flow monitor to an interface where the traffic is monitored.
6. Monitor the flow cache and statistics.
7. Clean the flow cache and statistics.
8. Export the flow cache to external source in .csv format.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **match ipv4 source address**
5. **match ipv4 destination address**
6. **collect counter packets long**
7. **exit**
8. **flow record** *record-name*
9. **match ipv4precedence**
10. **collect counter packets long**
11. **exit**
12. **flow monitor** *flow-monitor-name*
13. **record** *record-name*
14. **cache timeout active** *seconds*
15. **cache entries** *number*
16. **exit**
17. **flow monitor** *flow-monitor-name*
18. **record** *record-name*
19. **cache timeout active** *seconds*
20. **cache entries** *number*
21. **exit**
22. **interface** *type number*
23. **ip flow monitor** *monitor-name* **input**
24. **ip flow monitor** *monitor-name* **input**
25. **exit**
26. **show flow monitor** *monitor-name* **cache**

27. **show flow monitor** *monitor-name* **cache**
28. **clear flow monitor** *monitor-name* **cache**
29. **clear flow monitor** *monitor-name* **statistics**
30. **show flow monitor** *monitor-name* **cache format csv**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record ip-acct`	Creates or modifies an existing Flexible NetFlow flow record, and enters Flexible NetFlow flow record configuration mode.
Step 4	**match ipv4 source address** **Example:** `Device(config-flow-record)# match ipv4 source address`	Configures the IPv4 source address as a key field for a flow record.
Step 5	**match ipv4 destination address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	Configures the IPv4 destination address as a key field for a flow record.
Step 6	**collect counter packets long** **Example:** `Device(config-flow-record)# collect counter packets long`	Configures a 64-bit counter that is incremented for each packet seen in the flow.
Step 7	**exit** **Example:** `Device(config-flow-record)# exit`	Exits Flexible NetFlow flow record configuration mode and returns to global configuration mode.
Step 8	**flow record** *record-name* **Example:** `Device(config)# flow record prec-acct`	Creates or modifies an existing Flexible NetFlow flow record, and enters Flexible NetFlow flow record configuration mode.
Step 9	**match ipv4precedence** **Example:** `Device(config-flow-record) match ipv4 precedence`	Configures the IPv4 precedence (part of type of service) as a key field.

	Command or Action	Purpose
Step 10	**collect counter packets long** **Example:** `Device(config-flow-record)# collect counter packets long`	Configures a 64-bit counter that is incremented for each packet seen in the flow.
Step 11	**exit** **Example:** `Device(config-flow-record)# exit`	Exits Flexible NetFlow flow record configuration mode and returns to global configuration mode.
Step 12	**flow monitor** *flow-monitor-name* **Example:** `Device(config)# flow monitor ip-acct`	Creates or modifies an existing Flexible NetFlow flow monitor and enters Flexible NetFlow flow monitor configuration mode.
Step 13	**record** *record-name* **Example:** `Device(config-flow-monitor)# record ip-acct`	Configures a user-defined flow record that was previously configured for a Flexible NetFlow flow monitor.
Step 14	**cache timeout active** *seconds* **Example:** `Device(config-flow-monitor)# cache timeout active 604800`	Specifies the active flow timeout, in seconds for the flow monitor. **Note** Cisco IOS XE Releases do not support permanent cache, but allow cache timeout up to 7 days by configuring this command.
Step 15	**cache entries** *number* **Example:** `Device(config-flow-monitor)# cache entries 200000`	Specifies the maximum number of entries in the flow monitor cache.
Step 16	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode and returns to global configuration mode.
Step 17	**flow monitor** *flow-monitor-name* **Example:** `Device(config)# flow monitor prec-acct`	Create or modifies an existing Flexible NetFlow flow monitor, and enters Flexible NetFlow flow monitor configuration mode.
Step 18	**record** *record-name* **Example:** `Device(config-flow-monitor)# record prec-acct`	Configures a user-defined flow record that was previously configured for a Flexible NetFlow flow monitor.
Step 19	**cache timeout active** *seconds* **Example:** `Device(config-flow-monitor)# cache timeout active 604800`	Specifies the active flow timeout, in seconds for the flow monitor. **Note** Cisco IOS XE Releases do not support permanent cache, but allow cache timeout up to 7 days by configuring this command.

	Command or Action	Purpose
Step 20	**cache entries** *number* **Example:** `Device(config-flow-monitor)# cache entries 200000`	Specifies the maximum number of entries in the flow monitor cache.
Step 21	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode and returns to global configuration mode.
Step 22	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/4`	Configures an interface and enters interface configuration mode.
Step 23	**ip flow monitor** *monitor-name* **input** **Example:** `Device(config-if)# ip flow monitor ip-acct input`	Enables a Flexible NetFlow flow monitor for IPv4 traffic that the router is transmitting.
Step 24	**ip flow monitor** *monitor-name* **input** **Example:** `Device(config-if)# ip flow monitor prec-acct input`	Enables a Flexible NetFlow flow monitor for IPv4 traffic that the router is transmitting.
Step 25	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 26	**show flow monitor** *monitor-name* **cache** **Example:** `Device# show flow monitor prec-acct cache`	Displays the contents of the cache for the flow monitor record that was previously configured.
Step 27	**show flow monitor** *monitor-name* **cache** **Example:** `Device# show flow monitor ip-acct cache`	Displays the contents of the cache for the flow monitor record that was previously configured.
Step 28	**clear flow monitor** *monitor-name* **cache** **Example:** `Device# clear flow monitor ip-acct cache`	Clears the flow monitor cache information.
Step 29	**clear flow monitor** *monitor-name* **statistics** **Example:** `Device# clear flow monitor ip-acct statistics`	Clears the flow monitor statistics.
Step 30	**show flow monitor** *monitor-name* **cache format csv** **Example:** `Device# show flow monitor ip-acct cache format csv | append bootflash:ip-acct`	Exports the flow monitor cache contents to an external source in comma separated variables (CSV) format.

Configuration Examples for IP Services

Example: Protecting Your Network from DOS Attacks

The following example shows how to change some of the ICMP defaults for Gigabit Ethernet interface 0/0/0 to prevent ICMP from relaying information about paths, routes, and network conditions, which can be used by an attacker to gain network mapping information.

Disabling the unreachable messages will have a secondary effect: it will also disable IP Path MTU Discovery, because path discovery works by having the Cisco IOS XE software send Unreachable messages. If you have a network segment with a small number of devices and an absolutely reliable traffic pattern—which could easily happen on a segment with a small number of rarely used user devices—you would be disabling options that your device would be unlikely to use anyway.

```
Device(config)# no ip source-route
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# no ip unreachables
Device(config-if)# no ip redirects
Device(config-if)# no ip mask-reply
```

Example: Configuring ICMP Unreachable Destination Counters

The following example shows how to clear all of the unreachable destination packet statistics and to specify an interval number for unreachable destination messages. This example also shows how to configure a packet counter threshold and interval to trigger a logging message to a console.

```
Router# clear ip icmp rate-limit ethernet 0/0
Router# configure terminal
Router(config)# ip icmp rate-limit unreachable df log 1100 12000
```

Example: Setting the MTU Packet Size

The following example shows how to change the default MTU packet size for Gigabit Ethernet interface 0/0/0:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip mtu 300
```

Example: Configuring IP Accounting with NetFlow

The following example shows how to use NetFlow for IP Accounting:

```
! Created flow record and flow monitor for IP address accounting
Device# configure terminal
Device(config)# flow record ip-acct
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# exit
```

```
Device(config)# flow monitor ip-acct
Device(config-flow-monitor)# record ip-acct
Device(config-flow-monitor)# cache timeout active 604800
Device(config-flow-monitor)# cache entries 200000
Device(config-flow-monitor)# exit

! Created flow record and flow monitor for precedence accounting
Device(config)# flow record prec-acct
Device(config-flow-record)# match ipv4 precedence
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# exit
Device(config)# flow monitor prec-acct
Device(config-flow-monitor)# record prec-acct
Device(config-flow-monitor)# cache timeout active 604800
Device(config-flow-monitor)# cache entries 200000
Device(config-flow-monitor)# exit

! Apply both ip-acct and prec-acct on an interface
Device(config)# interface GigabitEthernet 0/0/4
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# ip flow monitor ip-acct input
Device(config-if)# ip flow monitor prec-acct input
Device(config-if)# negotiation auto
Device(config-if)# end
```

Verifying IP Accounting with NetFlow

SUMMARY STEPS

1. **show flow monitor** *monitor-name***cache**
2. **show flow monitor** *monitor-name* **cache**
3. **clear flow monitor** *monitor-name* {**cache** | **force-export** | **statistics**}
4. **show flow monitor** *monitor-name* **format csv | append bootflash:***monitor-name*}

DETAILED STEPS

Step 1 **show flow monitor** *monitor-name***cache**

Displays the contents of the cache for the flow monitor.

Example:

```
Device# show flow monitor prec-acct cache

  Cache type:                               Normal (Platform cache)
  Cache size:                               200000
  Current entries:                               3

  Flows added:                                   3
  Flows aged:                                    0

IP PREC             pkts long
=======  ====================
      0               8117679
      1               8118233
```

```
2               8118761
```

Step 2 **show flow monitor** *monitor-name* **cache**

Displays the contents of the cache for the flow monitor.

Example:

```
Device# show flow monitor ip-acct cache

  Cache type:                               Normal (Platform cache)
  Cache size:                               200000
  Current entries:                             10

  Flows added:                                 10
  Flows aged:                                   0

IPV4 SRC ADDR    IPV4 DST ADDR               pkts long
===============  ===============  ====================
192.168.0.1      192.168.2.2                5987314
192.168.0.1      192.168.3.2                5987314
192.168.0.1      192.168.10.2               5987354
192.168.0.1      192.168.1.2                5987363
192.168.0.1      192.168.8.2                5987384
192.168.0.1      192.168.7.2                5987387
192.168.0.1      192.168.6.2                5987420
192.168.0.1      192.168.9.2                5987606
192.168.0.1      192.168.5.2                5987645
192.168.0.1      192.168.2.2                5987659
```

Step 3 **clear flow monitor** *monitor-name* {**cache** | **force-export** | **statistics**}

Clears the flow monitor cache information.

Example:

```
Device# clear flow monitor ip-acct cache
```

Step 4 **show flow monitor** *monitor-name* **format csv | append bootflash:***monitor-name*}

Displays output of statistics from the flows in a flow monitor cache in comma-separated variables (CSV) format.

Example:

```
Device# show flow monitor ip-acct cache format csv | append bootflash:ip-acct
```

Additional References For IP Services

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IP application services commands	Cisco IOS IP Application Services Command Reference

Standards and RFCs

Standard	Title
RFC 1256	ICMP Router Discovery Messages

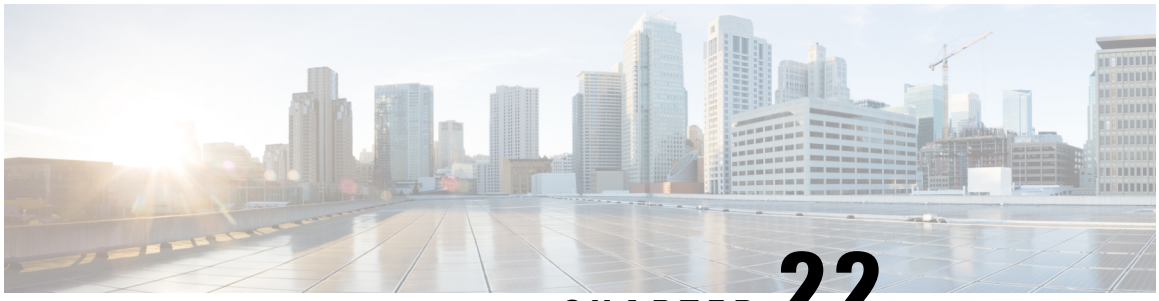
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP Services

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 22

Configuring IPv4 Broadcast Packet Handling

This module explains what IPv4 broadcast packets are, when they are used, and how to customize your router's configuration for situations when the default behavior for handling IPv4 broadcast packets isn't appropriate.

This module also explains some common scenarios that require customizing IPv4 broadcast packet handling by routers. For example, UDP forwarding of Dynamic Host Configuration Protocol (DHCP) traffic to ensure broadcast packets sent by DHCP clients can reach DHCP servers that are not on the same network segment as the client. Configuration tasks and examples are also provided in this module.

Information About IPv4 Broadcast Packet Handling

IP Unicast Address

An IP unicast address is not a broadcast addresses. A packet with an unicast destination IP address is intended for a specific IP host. For example, 172.16.1.1/32. Only the intended host of a unicast packets receives and processes the packet. This term is often used in conjunction with references to types of IP broadcast traffic. For example, a network administrator considering upgrading a router in a network must consider the amount of unicast, multicast, and broadcast traffic because each type of traffic can have a different effect on the performance of the router.

IP Broadcast Address

IP broadcast packets are sent to the destination IP broadcast address 255.255.255.255 (or the older but still occasionally used IP broadcast address of 000.000.000.000). The broadcast destination IP addresses 255.255.255.255 and 000.000.000.000 are used when a packet is intended for every IP-enabled device on a network.

Note Packets that use the broadcast IP address as the destination IP address are known as broadcast packets.

If routers forwarded IP broadcast packets by default, the packets would have to be forwarded out every interface that is enabled for IP because the 255.255.255.255 IP destination address is assumed to be reachable via every IP enabled interface in the router. Forwarding IP broadcast packets out every interface that is enabled for IP would result in what is known as a broadcast storm (network overload due to high levels of broadcast traffic). In order to avoid the IP packet broadcast storm that would be created if a router forwarded packets with a broadcast IP destination address out every IP-enabled interface, the default behavior for a router is to *not* forward broadcast packets. This is a key difference between routing IP traffic at Layer 3 versus bridging it at Layer 2. Layer 2 bridges by default forward IP broadcast traffic out every interface that is in a forwarding state, which can lead to scalability problems.

Some TCP/IP protocols use the IP broadcast address to either communicate with all of the hosts on a network segment or to identify the IP address of a specific host on a network segment. For example:

- Routing Information Protocol (RIP) version 1 sends routing table information using the IP broadcast address so that any other host on the network segment running RIP version 1 can receive and process the updates.
- The Address Resolution Protocol (ARP) is used to determine the Layer 2 MAC address of the host that owns a specific Layer 3 IP address. ARP sends an IP broadcast packet (that is also a Layer 2 broadcast frame) on the local network. All of the hosts on the local network receive the ARP broadcast packet because it is sent to as a Layer 2 broadcast frame. All of the hosts on the local network process the ARP packet because it is sent to the IP broadcast address. Only the host that owns the IP address indicated in the data area of the ARP packet responds to the ARP broadcast packet.

IP Network Broadcast

In Cisco IOS-XE devices, by default, the network prefix-directed broadcast packets are dropped in the ingress interface of a device.

Figure 25: A Simple Network with devices

In the figure, D3 has some network prefix-directed broadcast packets to be delivered through D2 to 192.168.10.x on the network. To achieve this, configure the ingress interface of D2 with the **ip network- broadcast** command. This enables D2 to receive and accept network prefix-directed broadcast packets.

IP Directed Broadcast Address

An IP directed broadcast is intended to reach all hosts on a remote network. A router that needs to send data to a remote IP host when only the IP network address is known uses an IP directed broadcast to reach the remote host. For example, a directed broadcast sent by a host with an IP address of 192.168.100.1 with a destination IP address of 172.16.255.255 is intended only for hosts that are in the 172.16.0.0 address space (hosts that have an IP address that begins with 172.16.0.0).

An IP directed broadcast packet is routed through the network as a unicast packet until it arrives at the target subnet, where it is converted into a Layer 2 broadcast frame (MAC address of FFFF.FFFF.FFFF). Because of the nature of the IP addressing architecture, only the last router in the chain, the one that is connected

directly to the target subnet, can conclusively identify a directed broadcast. For example, only a router with an interface connected to a network using an IP address in the 172.16.0.0/16 address space such as 172.16.1.1/16 can determine that a packet sent to 172.16.255.255 is a directed broadcast and convert it to a Layer 2 broadcast that is received by all hosts on the local network. The other routers in the network that are not connected to the 172.16.0.0/16 network forward packets addressed to 172.16.255.255 as if they were for a specific IP host.

All of the hosts on the remote network receive IP directed broadcasts after they are converted to Layer 2 broadcast frames. Ideally only the intended destination host will fully process the IP directed broadcast and respond to it. However, IP directed broadcasts can be used for malicious purposes. For example, IP directed broadcasts are used in "smurf" Denial of Service (DoS) attack and derivatives thereof. In a "smurf" attack, the attacker sends Internet Control Message Protocol (ICMP) echo requests (pings) to a directed broadcast address using the source IP address of the device that is the target of the attack. The target is usually a host inside a company's network such as a web server. The ICMP echo requests are sent to an IP directed broadcast address in the company's network that causes all the hosts on the target subnet to send ICMP echo replies to the device under attack. By sending a continuous stream of such requests, the attacker can create a much larger stream of replies, which can completely inundate the host that is under attack. For information on how IP directed broadcasts are used in DoS attacks, search the Internet for "IP directed broadcasts," "denial of service," and "smurf attacks."

Due to the security implications of allowing a router to forward directed broadcasts and the reduction in applications that require directed broadcasts, IP directed broadcasts are disabled by default in Cisco IOS Release 12.0 and later releases. If your network requires support for IP directed broadcasts, you can enable it on the interfaces that you want to translate the IP directed broadcasts to Layer 2 broadcasts using the **ip directed-broadcast** command. For example, if your router is receiving IP directed broadcasts on Fast Ethernet interface 0/0 for the network address assigned to Fast Ethernet interface 0/1, and you want the IP directed broadcasts to be translated to Layer 2 broadcasts out interface Fast Ethernet interface 0/1, configure the **ip directed-broadcast** command on Fast Ethernet interface 0/1. You can specify an access list to control which IP directed broadcasts are translated to Layer 2 broadcasts. When an access list is specified, only those IP packets permitted by the access list are eligible to be translated from directed broadcasts to Layer 2 broadcasts. For example, if you know that the only legitimate source IP address of any IP directed broadcasts in your network is 192.168.10.2, create an extended IP access list allowing traffic from 192.168.10.2 and assign the access list with the **ip directed-broadcast***access-list* command.

IP Directed Broadcasts

IP directed broadcasts are dropped by default. Dropping IP directed broadcasts reduces the risk of DoS attacks.

You can enable forwarding of IP directed broadcasts on an interface where the broadcast becomes a physical broadcast. You enable the translation of directed IP broadcast packets to Layer 2 broadcast frames on the interface that is connected to the IP network that the IP directed broadcast is addressed to. For example, if you need to translate IP directed broadcasts with the IP destination address of 172.16.10.255 to Layer 2 broadcast frames, you enable the translation on the interface that is connected to IP network 172.16.10.0/24.

You can specify an access list to control which directed broadcasts are forwarded. When an access list is specified, only those IP packets permitted by the access list are eligible to be translated from directed broadcasts to physical broadcasts.

IP directed broadcasts are disabled by default in Cisco IOS Release 12.0 and newer releases.

IP Multicast Addresses

IP multicast addresses are intended to reach an arbitrary subset of the hosts on a local network. IP broadcast addresses create a problem because every host must receive and process the data in each packet to determine

if it contains information that the host must process further. IP multicast addresses resolve this problem by using well-known IP addresses that a host must be configured to recognize before it will process packets addressed to it. When a host receives an IP multicast packet, the host compares the IP multicast address with the list of multicast addresses it is configured to recognize. If the host is not configured to recognize the IP multicast address, the host ignores the packet instead of processing it further to analyze the data in the packet. Because the host can ignore the packet it spends less time and fewer resources than it would have had to spend if the packet had been an IP broadcast that had to be processed all the way to the data layer before it was discarded.

The range of IP addresses reserved for Class D multicast addresses is 224.0.0.0 to 239.255.255.255/32 (255.255.255.255).

Most of the TCP/IP routing protocols use IP multicast addresses to send routing updates and other information to hosts on the same local network that are running the same routing protocol. Many other applications such as audio/video streaming over the Internet use IP multicast addresses. For a list of the currently assigned IP multicast addresses see Internet Multicast Addresses.

Information on configuring network devices for IP multicast support is available in the following documentation:

- *Cisco IOS IP Multicast Configuration Guide*
- *Cisco IOS IP Multicast Command Reference*

Early IP Implementations

Several early IP implementations do not use the current broadcast address standard of 255.255.255.255. Instead, they use the old standard, which calls for all zeros (000.000.000.000) instead of all ones to indicate broadcast addresses. Many of these implementations do not recognize an all-1s broadcast address and fail to respond to the broadcast correctly. Others forward all-1s broadcasts by default, which causes a serious network overload known as a *broadcast storm*. Implementations that exhibit these problems include systems based on versions of Berkeley Standard Distribution (BSD) UNIX prior to Version 4.3.

DHCP and IPv4 Broadcast Packets

DHCP requires that the client (host requiring information from the DHCP server) send broadcast packets to find a DHCP server to request configuration information from. If the DHCP server is not on the same network segment as the client that is sending the DHCP broadcasts, the router must be configured to forward the DHCP requests to the appropriate network.

For more information on DHCP, see RFC 2131 *Dynamic Host Configuration Protocol,* at http://www.ietf.org/rfc/rfc2131.txt.

UDP Broadcast Packet Forwarding

UDP broadcast packets are used by TCP/IP protocols such as DHCP and applications that need to send the same data to multiple hosts concurrently. Because routers by default do not forward broadcast packets you need to customize your router's configuration if your network has UDP broadcast traffic on it. One option for forwarding UDP broadcast packets is to use the UDP forwarding feature. UDP forwarding rewrites the broadcast IP address of a UDP packet to either a unicast (specific host) IP address or a directed IP broadcast. After the address is rewritten the UDP packet is forwarded by all of the routers in the path to the destination network without requiring additional configuration changes on the other routers.

You can enable forwarding of UDP broadcast packets, such as DHCP requests, to a host, or to multiple hosts on the same target network. When a UDP broadcast packet is forwarded, the destination IP address is rewritten to match the address that you configure. For example, the **ip helper-address 172.16.10.2** command rewrites the IP destination address from 255.255.255.255 to 172.16.10.2.

To enable UDP broadcast packet forwarding to specific host, use a specific host IP address as the helper address when you configure the **ip helper-address** *address* command. To enable UDP broadcast packet forwarding to a range of hosts to allow for load sharing and redundancy, use an IP directed broadcast address as the helper address when you configure the **ip helper-address** *address* command.

UDP Broadcast Packet Flooding

You can allow IP broadcasts to be flooded throughout your network in a controlled fashion using the database created by theLayer 2 bridging Spanning Tree Protocol (STP). Enabling this feature also prevents flooding loops. In order to support this capability, the Cisco IOS software on your router must include support for transparent bridging, and transparent bridging must be configured on each interface that is to participate in the flooding. If bridging is not configured on an interface, the interface is still able to receive broadcasts. However, the interface will never forward broadcasts it receives, and the router will never use that interface to send broadcasts received on a different interface.

Packets that are forwarded to a single network address using the IP helper address mechanism can be flooded. Only one copy of the packet is sent on each network segment.

In order to be considered for flooding, packets must meet the following criteria. (These are the same conditions used to consider packet forwarding using IP helper addresses.)

- The packet must be a MAC-level broadcast (FFFF.FFFF.FFFF).
- The packet must be an IP-level broadcast (255.255.255.255).
- The packet must be a Trivial File Transfer Protocol (TFTP), Domain Name System (DNS), Time, NetBIOS, Neighbor Discovery (ND), or BOOTP packet, or a UDP protocol specified by the **ip forward-protocol udp** global configuration command.
- The time-to-live (TTL) value of the packet must be at least two.

If you want to send the flooded UDP packets to a specific host, you can change the Layer 3 IP broadcast address of the flooded UDP packets with the **ip broadcast-address** command in interface configuration mode. The address of the flooded UDP packets can be set to any desired IP address. The source address of the flooded UDP packet is never changed. The TTL value of the flooded UDP packet is decremented.

After a decision has been made to send the datagram out on an interface (and the destination IP address possibly changed), the datagram is handed to the normal IP output routines and is, therefore, subject to access lists if they are present on the output interface.

If no actual bridging is desired, you can configure a type-code bridging filter that will deny all packet types from being bridged. Refer to the "Configuring Transparent Bridging" module of the *Cisco IOS Bridging and IBM Networking Configuration Guide* for more information about using access lists to filter bridged traffic. The Spanning-Tree database is still available to the IP forwarding code to use for the flooding.

IP Broadcast Flooding Acceleration

You can accelerate flooding of UDP datagrams using the spanning-tree algorithm. Used in conjunction with the **ip forward-protocol spanning-tree** command in global configuration mode, this feature boosts the

performance of spanning-tree-based UDP flooding by a factor of about four to five times. The feature, called *turbo flooding*, is supported over Ethernet interfaces configured for Advanced Research Projects Agency (ARPA) encapsulated, FDDI, and high-level data link control (HDLC)-encapsulated serial interfaces. However, it is not supported on Token Ring interfaces. As long as the Token Rings and the non-HDLC serial interfaces are not part of the bridge group being used for UDP flooding, turbo flooding will behave normally.

Default UDP Port Numbers

If a helper address is specified and UDP forwarding is enabled, broadcast packets destined to the following port numbers are forwarded by default:

- Time service (port 37)
- IEN-116 Name Service (port 42)
- TACACS service (port 49)
- Domain Naming System (port 53)
- BOOTP client and server packets (ports 67 and 68)
- TFTP (port 69)
- NetBIOS Name Server (port 137)
- NetBIOS Datagram Server (port 138)

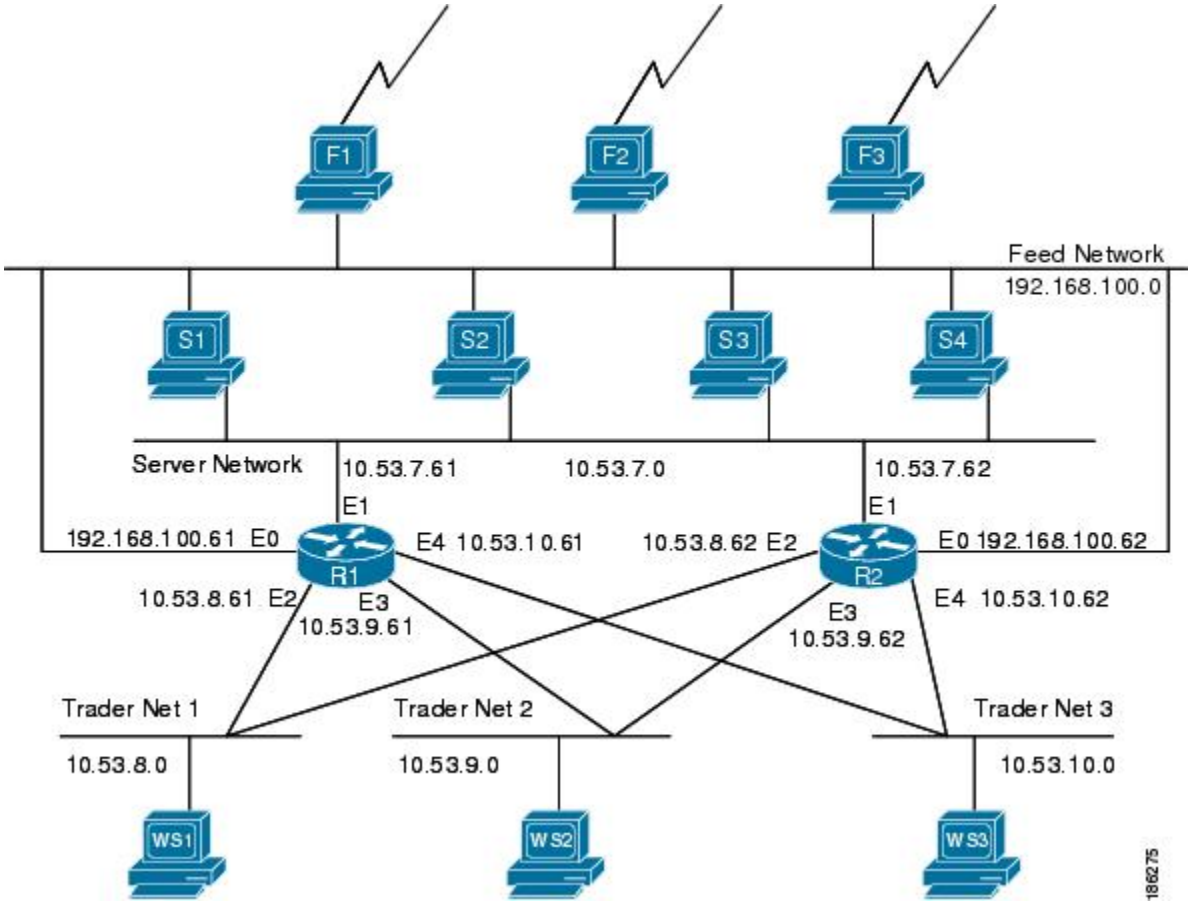
Default IP Broadcast Address

The Cisco IOS software supports sending IP broadcasts on both LANs and WANs. There are several ways to indicate an IP broadcast address. The default is an address consisting of all ones (255.255.255.255), although the software can be configured to generate any form of IP broadcast address such as all zeros (0.0.0.0), and directed broadcasts such as 172.16.255.255. Cisco IOS software can receive and process most IP broadcast addresses.

UDP Broadcast Packet Case Study

This case study is from a trading floor application in a financial company. The workstations (WS1, WS2, and WS3) in the following figure receive financial data from the feed network. The financial data is sent using UDP broadcasts.

Figure 26: Topology that Requires UDP Broadcast Forwarding

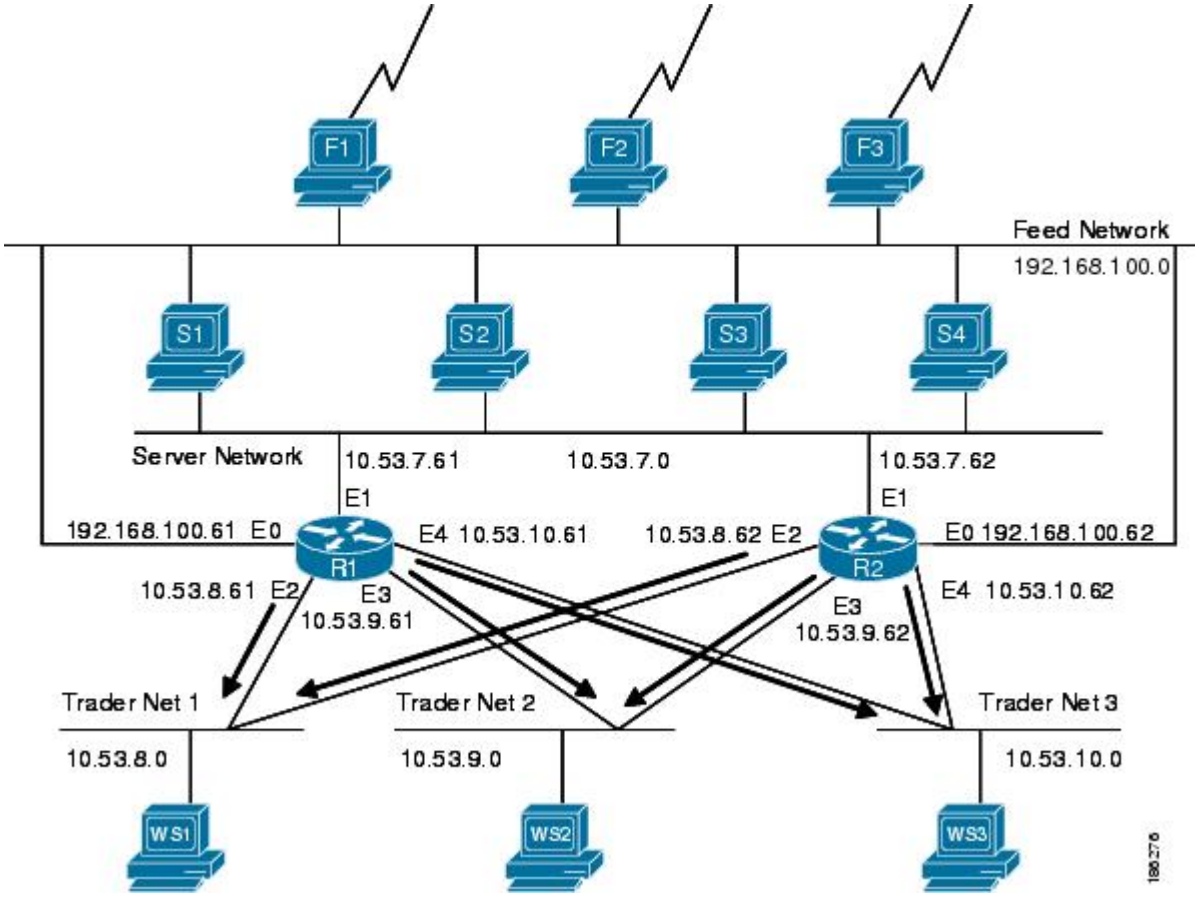


The following sections explain the possible solutions for this application:

UDP Broadcast Packet Forwarding

The first option is UDP broadcast packet using helper addresses. To configure helper addressing, you must specify the **ip helper-address** command on every interface on every router that receives a UDP broadcast that needs to be forwarded. On router 1 and router 2 in the figure below, IP helper addresses can be configured to move data from the server network to the trader networks. However IP helper addressing was determined not to be an optimal solution for this type of topology because each router receives unnecessary broadcasts from the other router, as shown in the figure below.

Figure 27: Flow of UDP Packets



In this case, router 1 receives each broadcast sent by router 2 three times, one for each segment, and router 2 receives each broadcast sent by router 1 three times, one for each segment. When each broadcast is received, the router must analyze it and determine that the broadcast does not need to be forwarded. As more segments are added to the network, the routers become overloaded with unnecessary traffic, which must be analyzed and discarded.

When IP helper addressing is used in this type of topology, no more than one router can be configured to forward UDP broadcasts (unless the receiving applications can handle duplicate broadcasts). This is because duplicate packets arrive on the trader network. This restriction limits redundancy in the design and can be undesirable in some implementations.

To configure routers to send UDP broadcasts bidirectionally in this type of topology, a second **ip helper address** command must be applied to every router interface that receives UDP broadcasts. As more segments and devices are added to the network, more **ip helper address** commands are required to reach them, so the administration of these routers becomes more complex over time.

Note Bidirectional traffic in this topology significantly impacts router performance.

Although IP helper addressing is well-suited to nonredundant, nonparallel topologies that do not require a mechanism for controlling broadcast loops, IP helper addressing does not work well in this topology. To improve performance, the network designers considered four other alternatives:

- Setting the broadcast address on the servers to all ones (255.255.255.255)—This alternative was dismissed because the servers have more than one interface, causing server broadcasts to be sent back onto the feed network. In addition, some workstation implementations do not allow all 1s broadcasts when multiple interfaces are present.
- Setting the broadcast address of the servers to the major network broadcast IP address--This alternative was dismissed because the TCP/IP implementation on the servers does not allow the use of major network IP broadcast addresses when the network is subnetted.
- Eliminating the subnets and letting the workstations use Address Resolution Protocol (ARP) to learn addresses—This alternative was dismissed because the servers cannot quickly learn an alternative route in the event of a primary router failure.
- UDP broadcast packet flooding—This alternative uses the spanning-tree topology created with transparent bridging to forward UDP broadcast packets in a redundant topology while avoiding loops and duplicate broadcast traffic.

UDP Broadcast Packet Flooding

UDP flooding uses the spanning-tree algorithm to forward packets in a controlled manner. Bridging is enabled on each router interface for the sole purpose of building the spanning tree. The spanning tree prevents loops by stopping a broadcast from being forwarded out an interface on which the broadcast was received. The spanning tree also prevents packet duplication by placing certain interfaces in the blocked state (so that no packets are forwarded) and other interfaces in the forwarding state (so that packets that need to be forwarded are forwarded).

Before you can enable UDP flooding, the router must be running software that supports transparent bridging and bridging must be configured on each interface that is to participate in the flooding. If bridging is not configured for an interface, the interface will receive broadcasts, but the router will not forward those broadcasts and will not use that interface as a destination for sending broadcasts received on a different interface.

When configured for UDP flooding, the router uses the destination address specified by the **ip broadcast-address** command on the output interface to assign a destination address to a flooded UDP datagram. Thus, the destination address might change as the datagram propagates through the network. The source address, however, does not change.

With UDP flooding, both routers shown in the figure below use a spanning-tree to control the network topology for the purpose of forwarding broadcasts. The **bridge protocol** command can specify either the **dec** keyword (for the Digital Equipment Corporation (DEC) spanning-tree protocol) or the **ieee** keyword (for the IEEE Ethernet protocol). All routers in the network must enable the same spanning-tree protocol. The **ip forward-protocol spanning-tree** command uses the database created by the **bridge protocol** command. Only one broadcast packet arrives at each segment, and UDP broadcasts can traverse the network in both directions.

Because bridging is enabled only to build the spanning-tree database, use access lists to prevent the spanning-tree from forwarding non-UDP traffic.

The router configuration specifies a path cost for each interface to determine which interface forwards or blocks packets. The default path cost for Ethernet is 100. Setting the path cost for each interface on router 2 to 50 causes the spanning-tree algorithm to place the interfaces in router 2 in forwarding state. Given the higher path cost (100) for the interfaces in router 1, the interfaces in router 1 are in the blocked state and do

not forward the broadcasts. With these interface states, broadcast traffic flows through router 2. If router 2 fails, the spanning-tree algorithm will place the interfaces in router 1 in the forwarding state, and router 1 will forward broadcast traffic.

With one router forwarding broadcast traffic from the server network to the trader networks, you should configure the other router to forward unicast traffic. For that reason, each router enables the ICMP Router Discovery Protocol (IRDP), and each workstation on the trader networks runs the IRDP daemon. On router 1, the **preference** keyword of the **ip irdp** command sets a higher IRDP preference than does the configuration for router 2, which causes each IRDP daemon to use router 1 as its preferred default gateway for unicast traffic forwarding. Users of those workstations can use the **netstat -rn** command to see how the routers are being used.

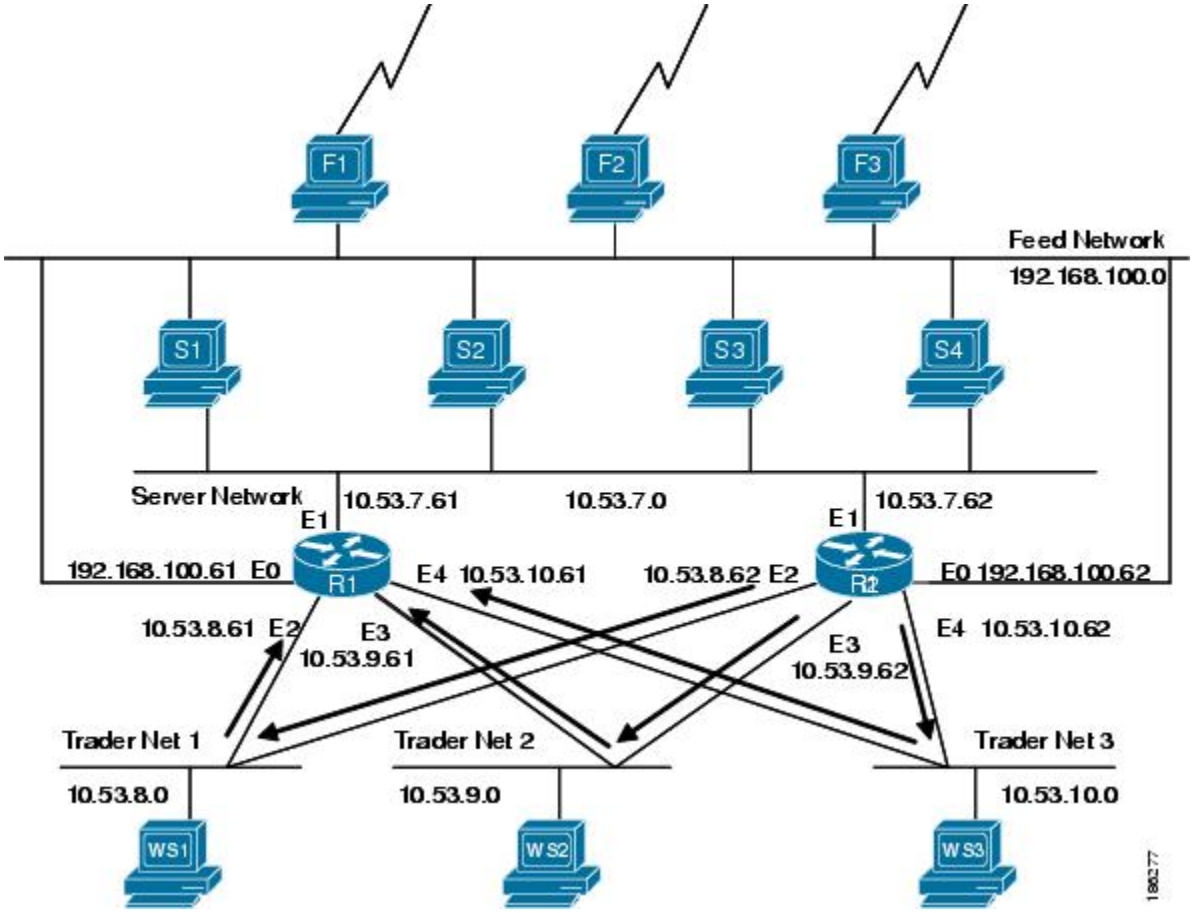
On the routers, the **holdtime**, **maxadvertinterval**, and **minadvertinterval** keywords of the **ip irdp** command reduce the advertising interval from the default so that the IRDP daemons running on the hosts expect to see advertisements more frequently. With the advertising interval reduced, the workstations will adopt router 2 more quickly if router 1 becomes unavailable. With this configuration, when a router becomes unavailable, IRDP offers a convergence time of less than one minute.

IRDP is preferred over the Routing Information Protocol (RIP) and default gateways for the following reasons:

- RIP takes longer to converge.
- Configuration of router 1 as the default gateway on each Sun workstation on the trader networks would allow those Sun workstations to send unicast traffic to router 1, but would not provide an alternative route if router 1 becomes unavailable.

The figure below shows how data flows when the network is configured for UDP flooding.

Figure 28: Data Flow with UDP Flooding and IRDP



Note This topology is broadcast intensive--broadcasts sometimes consume 20 percent of the 10-MB Ethernet bandwidth. However, this is a favorable percentage when compared to the configuration of IP helper addressing, which, in the same network, causes broadcasts to consume up to 50 percent of the 10-MB Ethernet bandwidth.

If the hosts on the trader networks do not support IRDP, Hot Standby Routing Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) can be used to select which router will handle unicast traffic. These protocols allow the standby router to take over quickly if the primary router becomes unavailable.

Enable turbo flooding on the routers to increase the performance of UDP flooding.

Note Turbo flooding increases the amount of processing that is done at interrupt level, which increases the CPU load on the router. Turbo flooding may not be appropriate on routers that are already under high CPU load or that must also perform other CPU-intensive activities.

Feature Information for IP Broadcast Packet Handling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 36: Feature Information for IP Broadcast Packet Handling

Feature Name	Releases	Feature Information
IP Directed Broadcasts	CISCO IOS XE 16	Enables the translation of a directed broadcast to physical broadcasts. The following command was introduced or modified by this feature: **ip directed-broadcast**.
IP Network Broadcasts	CISCO IOS XE Amsterdam 17.3.1	Discards traffic on receipt, unless explicitly configured to allow. The following command was introduced or modified by this feature: **ip network-broadcast**.

How to Configure IP Broadcast Packet Handling

Enable IP Network Broadcast

Perform this task to receive and accept the IP network broadcast from any source on the receiving interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *address mask*
5. **ip network-broadcast**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Specifies an interface and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 163.168.10.2 255.255.255.0`	Assigns an IP address to the interface.
Step 5	**ip network-broadcast** **Example:** `Device(config-if)# ip network-broadcast`	Enables IP network directed broadcasts on the interface. Configure this command on the receiving interface to receive and accept the network-prefix-directed broadcast packets. **Note** Default is disabled, therefore, the 'network-prefix-directed broadcast' packets are discarded.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Enabling IP Directed Broadcasts Without an Access List

Perform this task to permit the forwarding of IP directed broadcasts from any source.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Specifies an interface and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 172.16.10.1 255.255.255.0`	Assigns an IP address to the interface.
Step 5	**ip directed-broadcast** **Example:** `Device(config-if)# ip directed-broadcast`	Enables IP directed broadcasts on the interface. • Configure this command on the interface that is connected to the IP network address of the directed broadcast packets. • In this example the directed broadcast packets are addressed to 172.16.10.255. **Note** Default is disabled hence the 'network-prefix-directed broadcast' packets will be silently discarded.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Enabling IP Directed Broadcasts with an Access List

Perform this task to limit the forwarding of IP directed broadcasts by applying an access list to the **ip directed-broadcast** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *100-199* **permit ip** *source-address mask destination-address mask*
4. **interface** *type number*
5. **ip address** *address mask*
6. **ip directed-broadcast** *access-list*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *100-199* **permit ip** *source-address mask destination-address mask* **Example:** `Device(config)# access-list 100 permit ip 10.4.9.167 0.0.0.0 172.16.10.0 0.0.0.255`	Creates an access list to limit the IP directed broadcasts that are forwarded. • In this example the IP directed broadcasts are sent by the host with the IP address of 10.4.9.167 to the IP directed broadcast address 172.16.10.255.
Step 4	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *address mask* **Example:** `Device(config-if)# ip address 172.16.10.1 255.255.255.0`	Assigns an IP address to the interface.
Step 6	**ip directed-broadcast** *access-list* **Example:** `Device(config-if)# ip directed-broadcast 100`	Enables IP directed broadcasts on the interface for broadcast packets that are allowed by the access list you assigned. Configure this command on the interface that is connected to the IP network address of the directed broadcast packets. • In this example the directed broadcast packets are addressed to 172.16.10.255.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Enabling Forwarding of UDP Broadcast Packets to a Specific Host

Perform this task to enable UDP broadcast packet forwarding to a single host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip forward-protocol udp**
4. **interface** *type number*
5. **ip address** *address mask*
6. **ip helper-address** *address*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip forward-protocol udp** **Example:** `Device(config)# ip forward-protocol udp`	Enables forwarding of UDP broadcast packets.
Step 4	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *address mask* **Example:** `Device(config-if)# ip address 172.16.10.1 255.255.255.0`	Assigns an IP address to the interface.
Step 6	**ip helper-address** *address* **Example:** `Device(config-if)# ip helper-address 172.16.10.2`	Enables an IP helper address for the interface that is receiving the UDP broadcast packets. • In this example the IP destination address of the IP UDP broadcast packets is rewritten to 172.16.10.2.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Enabling Forwarding of UDP Broadcast Packets to a Range of Hosts

Perform this task to enable UDP broadcast packet forwarding to a range of hosts to allow for load sharing between the destination hosts and to provide redundancy if one or more of the destination hosts fail.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip forward-protocol udp**
4. **interface** *type number*
5. **ip address** *address mask*
6. **ip helper-address** *address*
7. **exit**
8. **interface** *type number*
9. **ip address** *address mask*
10. **ip directed-broadcast**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip forward-protocol udp** **Example:** `Device(config)# ip forward-protocol udp`	Enables forwarding of UDP broadcast packets.
Step 4	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.168.10.1 255.255.255.0`	Assigns an IP address to the interface.

	Command or Action	Purpose
Step 6	**ip helper-address** *address* **Example:** `Device(config-if)# ip helper-address 172.16.10.255`	Enables an IP helper address for the interface that is receiving the UDP broadcast packets. • In this example an IP directed broadcast address is used. The IP destination address of the IP UDP broadcast packets is rewritten to 172.16.10.255. • All of the hosts on the 172.16.10.0/24 network that support the application or service that the UDP broadcast packets are intended for will respond to the UDP broadcast packets. **Note** This often results in the source of the UDP broadcast packets receiving responses from two or more hosts. In most circumstances the source of the UDP broadcast packets accepts the first response and ignores any subsequent responses. In some situations the source of the UDP broadcast packets cannot handle duplicate responses and reacts by reloading, or other unexpected behavior.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Specifies an interface and enters interface configuration mode.
Step 9	**ip address** *address mask* **Example:** `Device(config-if)# ip address 172.16.10.1 255.255.255.0`	Assigns an IP address to the interface.
Step 10	**ip directed-broadcast** **Example:** `Device(config-if)# ip directed-broadcast`	Enables IP directed broadcasts on the interface that is transmitting the UDP broadcasts.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Changing the Default IP Broadcast Address for All Interfaces to 0.0.0.0 on Routers Without Nonvolatile Memory

If you router does not have NVRAM, and you need to change the IP broadcast address to 0.0.0.0, you must change the IP broadcast address manually by setting jumpers in the processor configuration register. Setting bit 10 causes the device to use all 0s. Bit 10 interacts with bit 14, which controls the network and host portions of the broadcast address. Setting bit 14 causes the device to include the network and host portions of its address in the broadcast address. The table below shows the combined effect of setting bits 10 and 14.

Table 37: Configuration Register Settings for Broadcast Address Destination

Bit 14	Bit 10	Address (<net><host>)
Out	Out	<ones><ones>
Out	In	<zeros><zeros>
In	In	<net><zeros>
In	Out	<net><ones>

For additional information on setting the hardware jumpers on your router, see the hardware documentation that was supplied with you router.

Changing the Default IP Broadcast Address for All Interfaces to 0.0.0.0 on Routers with Nonvolatile Memory

Cisco IOS-based routers with NVRAM have software configuration registers that allow you to modify several behaviors of the router such as where it looks for images to load, what IP broadcast address it uses, and the console line speed. The factory default value for the configuration register is 0x2102 where *0X* indicates this a hexadecimal number. The **config-register** command is used to modify the settings of the software configuration registers.

Information on configuring other behaviors with the software configuration registers using the **config-register** command is available in the following documentation:

- "Loading and Managing System Images" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*
- *Cisco IOS Configuration Fundamentals Command Reference*

Caution You need to be very careful when you change the software configuration registers on your router because if you inadvertently alter the console port line speed, you will not be able to configure the router with a terminal server on the console port unless you know the speed that you set for the console port, and you know how to change the line speed for your terminal application. If your router is configured for alternate access to the CLI such as using Telnet or a web browser, you can use this method to log in to the router and change the software configuration register back to 0x2102.

Perform this task to set the IP broadcast address on every interface to 0.0.0.0 while maintaining the remainder of the default values for the software configuration register settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **config-register** *value*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**config-register** *value* **Example:** `Device(config)# config-register 0x2502`	Sets the IP broadcast address to 0.0.0.0 on every interface while maintaining the remainder of the default values for the other software configuration register settings.
Step 4	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Changing the IP Broadcast Address to Any IP Address on One or More Interfaces in a Router

Perform this task if you network requires an IP broadcast address other than 255.255.255.255 or 0.0.0.0, or you want to change the IP broadcast address to 0.0.0.0 on a subset of the interfaces on the router instead of on all of the interfaces on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *address mask*
5. **ip broadcast-address** *address*

6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Specifies an interface and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 172.16.10.1 255.255.255.0`	Assigns an IP address to the interface.
Step 5	**ip broadcast-address** *address* **Example:** `Device(config-if)# ip broadcast-address 172.16.10.255`	Specifies the IP broadcast address • In this example IP broadcasts are sent to 172.16.10.255.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuring UDP Broadcast Packet Flooding

Before you begin

The version of Cisco IOS software on your router must support transparent bridging.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *number* **protocol ieee**

4. **ip forward-protocol spanning-tree**
5. **ip forward-protocol turbo-flood**
6. **ip forward-protocol udp**
7. **interface** *type number*
8. **ip address** *address mask*
9. **bridge-group** *number*
10. **interface** *type number*
11. **ip address** *address mask*
12. **bridge-group** *number*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *number* **protocol ieee** **Example:** `Device(config)# bridge 1 protocol ieee`	Enables spanning-tree bridging and specifies the bridging protocol.
Step 4	**ip forward-protocol spanning-tree** **Example:** `Device(config)# ip forward-protocol spanning-tree`	Enables using the spanning-tree forwarding table to flood broadcast packets.
Step 5	**ip forward-protocol turbo-flood** **Example:** `Device(config)# ip forward-protocol turbo-flood`	(Optional) Enables fast forwarding of broadcast packets using the spanning-tree forwarding table.
Step 6	**ip forward-protocol udp** **Example:** `Device(config)# ip forward-protocol udp`	Enables forwarding of UDP broadcasts.
Step 7	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.168.10.1 255.255.255.0`	Assigns an IP address to the interface.
Step 9	**bridge-group** *number* **Example:** `Device(config-if)# bridge-group 1`	Places the interface in the spanning-tree bridge group specified.
Step 10	**interface** *type number* **Example:** `Device(config-if)# interface fastethernet 0/1`	Specifies an interface and enters interface configuration mode.
Step 11	**ip address** *address mask* **Example:** `Device(config-if)# ip address 172.16.10.1 255.255.255.0`	Assigns an IP address to the interface.
Step 12	**bridge-group** *number* **Example:** `Device(config-if)# bridge-group 1`	Places the interface in the spanning-tree bridge group specified.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for IP Broadcast Packet Handling

Example: Enabling IP Directed Broadcasts with an Access List

The following example shows how to enable IP directed broadcasts with an access list to control the directed broadcasts that are forwarded.

```
Device(config)# access-list 100 permit ip 10.4.9.167 0.0.0.0 172.16.10.0 0.0.0.255
Device(config)# interface fastethernet 0/0
Device(config-if)# ip address 172.16.10.1 255.255.255.0
Device(config-if)# ip directed-broadcast 100
```

Example: Configuring UDP Broadcast Packet Flooding

```
Device(config)# bridge 1 protocol ieee
Device(config)# ip forward-protocol spanning-tree
Device(config)# ip forward-protocol turbo-flood
Device(config)# ip forward-protocol udp
Device(config)# interface fastethernet 0/0
Device(config-if)# ip address 192.168.10.1 255.255.255.0
Device(config-if)# bridge-group 1
Device(config)# interface fastethernet 0/1
Device(config-if)# ip address 172.16.10.1 255.255.255.0
Device(config-if)# bridge-group 1
```

Additional References for WCCP—Configurable Router ID

Related Documents

Related Topic	Document Title
WCCP commands	Cisco IOS IP Application Services Command Reference
Currently assigned IP multicast addresses	*Internet Multicast Addresses* http://www.iana.org/assignments/multicast-addresses
Configuration fundamentals configuration tasks	*Cisco IOS Configuration Fundamentals Configuration Guide*
Configuration fundamentals commands	*Cisco IOS Configuration Fundamentals Command Reference*
Cisco IOS bridging and IBM networking configuration tasks	*Cisco IOS Bridging and IBM Networking Configuration Guide*
Cisco IOS bridging and IBM networking commands	*Cisco IOS Bridging and IBM Networking Command Reference*
Cisco IOS IP multicast configuration tasks	*Cisco IOS IP Multicast Configuration Guide*
Cisco IOS IP Multicast commands	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
IEEE Spanning-Tree Bridging	802.1D MAC Bridges http://www.ieee802.org/1/pages/802.1D-2003.html

MIBs

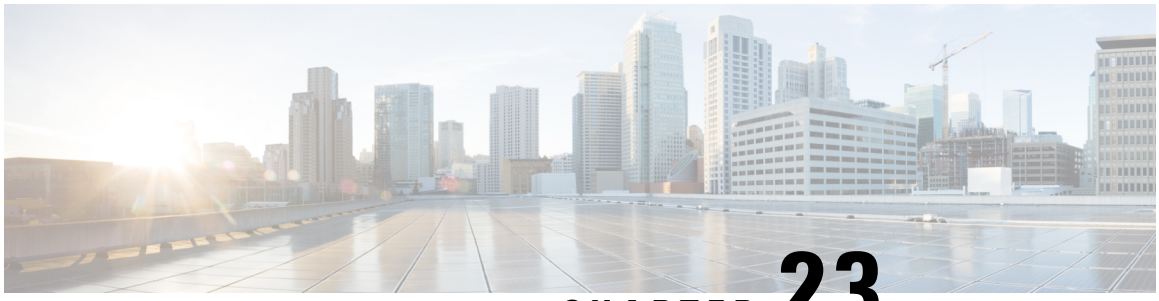
MIB	MIBs Link
—	No new or modified MIBs are supported, and support for existing MIBs has not been modified.

RFCs

RFC	Title
RFC 1812	*Requirements for IP Version 4 Routers* http://www.ietf.org/rfc/rfc1812.txt
RFC 2131	*Dynamic Host Configuration Protocol* http://www.ietf.org/rfc/rfc2131.txt .

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 23

Object Tracking: IPv6 Route Tracking

The Object Tracking: IPv6 Route Tracking feature expands the Enhanced Object Tracking (EOT) functionality to allow the tracking of IPv6 routes.

Restrictions for Object Tracking: IPv6 Route Tracking

Object Tracking: IPv6 Route Tracking is not Stateful Switchover (SSO)-aware and cannot be used with Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) in SSO mode.

Information About Object Tracking: IPv6 Route Tracking

Enhanced Object Tracking and IPv6 Route Tracking

Enhanced Object Tracking (EOT) provides complete separation between the objects to be tracked and the action to be taken by a client when a tracked object changes. Thus, several clients such as Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) can register interest with a tracking process, track the same object, and each take different a action when the object changes.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes use this number to track a specific object.

A tracking process periodically polls tracked objects and notes any change in value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

The Object Tracking: IPv6 Route Tracking feature expands EOT functionality to allow the tracking of IPv6 routes.

How to Configure Object Tracking: IPv6 Route Tracking

Tracking the IPv6-Routing State of an Interface

SUMMARY STEPS

1. **track timer interface** {*seconds* | **msec** *milliseconds*}
2. **track** *object-number* **interface** *type number* **ipv6 routing**
3. **carrier-delay**
4. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
5. **end**
6. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**track timer interface** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config)# track timer interface 5`	(Optional) Specifies the interval that a tracking process polls the tracked interface. • The default interval that the tracking process polls interface objects is 1 second. **Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the **msec** keyword and *milliseconds* argument.
Step 2	**track** *object-number* **interface** *type number* **ipv6 routing** **Example:** `Device(config)# track 1 interface GigabitEthernet 0/0/1 ipv6 routing`	Tracks the IPv6-routing state of an interface and enters tracking configuration mode. • IPv6-route tracking tracks an IPv6 route in the routing table and the ability of an interface to route IPv6 packets.
Step 3	**carrier-delay** **Example:** `Device(config-track)# carrier-delay`	(Optional) Enables enhanced object tracking to consider the carrier-delay timer when tracking the status of an interface.
Step 4	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. **Note** The **up** keyword specifies the time to delay the notification of an up event. The **down** keyword specifies the time to delay the notification of a down event.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.
Step 6	**show track** *object-number* **Example:** `Device# show track 1`	Displays tracking information. • Use this command to verify the configuration.

Tracking the Threshold of IPv6-Route Metrics

SUMMARY STEPS

1. **track timer ipv6 route** {*seconds* | **msec** *milliseconds*}
2. **track resolution ipv6 route** {**bgp** | **eigrp** | **isis** | **ospf** | **static** } *resolution-value*
3. **track** *object-number* **ipv6 route** *ipv6-address/prefix-length* **metric threshold**
4. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
5. **ipv6 vrf** *vrf-name*
6. **threshold metric** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]}
7. **end**
8. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**track timer ipv6 route** {*seconds*	**msec** *milliseconds*} **Example:** `Device(config)# track timer ipv6 route 20`	(Optional) Specifies the interval that a tracking process polls the tracked object. • The default interval that the tracking process polls IPv6-route objects is 15 seconds. **Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the **msec** keyword and *milliseconds* argument.
Step 2	**track resolution ipv6 route** {**bgp**	**eigrp**	**isis**	**ospf**	**static** } *resolution-value* **Example:** `Device(config)# track resolution ipv6 route eigrp 300`	(Optional) Specifies resolution parameters for a tracked object. • Use this command to change the default metric resolution values.

	Command or Action	Purpose
Step 3	**track** *object-number* **ipv6 route** *ipv6-address/prefix-length* **metric threshold** **Example:** `Device(config)# track 6 ipv6 route 2001:DB8:0:ABCD::1/10 metric threshold`	Tracks the scaled metric value of an IPv6 route to determine if it is above or below a threshold and enters tracking configuration mode. • The default down value is 255, which equates to an inaccessible route. • The default up value is 254.
Step 4	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. **Note** The **up** keyword specifies the time to delay the notification of an up event. The **down** keyword specifies the time to delay the notification of a down event.
Step 5	**ipv6 vrf** *vrf-name* **Example:** `Device(config-track)# ipv6 vrf VRF1`	(Optional) Tracks an IPv6 route in a specific VPN virtual routing and forwarding (VRF) table.
Step 6	**threshold metric** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]} **Example:** `Device(config-track)# threshold metric up 254 down 255`	(Optional) Sets a metric threshold other than the default value. **Note** The **up** keyword specifies the up threshold. The state is up if the scaled metric for that route is less than or equal to the up threshold. The default up threshold is 254. The **down** keyword specifies the down threshold. The state is down if the scaled metric for that route is greater than or equal to the down threshold. The default down threshold is 255.
Step 7	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.
Step 8	**show track** *object-number* **Example:** `Device# show track 6`	(Optional) Displays tracking information. • Use this command to verify the configuration.

Tracking IPv6-Route Reachability

Perform this task to track the reachability of an IPv6 route. A tracked object is considered up when a routing table entry exists for the route and the route is accessible.

SUMMARY STEPS

1. **track timer ipv6 route** {*seconds* | **msec** *milliseconds*}
2. **track** *object-number* **ip route** *ip-address*/*prefix-length* **reachability**
3. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
4. **ipv6 vrf** *vrf-name*
5. **end**
6. **show track** *object-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**track timer ipv6 route** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config)# track timer ipv6 route 20`	(Optional) Specifies the interval that a tracking process polls the tracked object. • The default interval that the tracking process polls IPv6-route objects is 15 seconds. **Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1-second interval configured using the **msec** keyword and *milliseconds* argument.
Step 2	**track** *object-number* **ip route** *ip-address*/*prefix-length* **reachability** **Example:** `Device(config)# track 4 ipv6 route 2001:DB8:0:AB82::1/10 reachability`	Tracks the reachability of an IPv6 route and enters tracking configuration mode.
Step 3	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:** `Device(config-track)# delay up 30`	(Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. **Note** The **up** keyword specifies the time to delay the notification of an up event. The **down** keyword specifies the time to delay the notification of a down event.
Step 4	**ipv6 vrf** *vrf-name* **Example:** `Device(config-track)# ipv6 vrf VRF2`	(Optional) Configures a VPN virtual routing and forwarding (VRF) table.
Step 5	**end** **Example:** `Device(config-track)# end`	Returns to privileged EXEC mode.
Step 6	**show track** *object-number* **Example:**	(Optional) Displays tracking information. • Use this command to verify the configuration.

	Command or Action	Purpose
	`Device# show track 4`	

Configuration Examples for Object Tracking: IPv6 Route Tracking

Example: Tracking the IPv6-Routing State of an Interface

The following example shows how to configure tracking for IPv6 routing on the GigabitEthernet 0/0/1 interface:

```
Device(config)# track timer interface 5
Device(config)# track 1 interface GigabitEthernet 0/0/1 ipv6 routing
Device(config-track)# carrier-delay
Device(config-track)# delay up 30
Device(config-track)# end
```

Example: Tracking the Threshold of IPv6-Route Metrics

The following example shows how to configure tracking for IPv6 metric thresholds:

```
Device(config)# track timer ipv6 route 20
Device(config)# track resolution ipv6 route eigrp 300
Device(config)# track 6 ipv6 route 2001:DB8:0:ABCD::1/10 metric threshold
Device(config-track)# delay up 30
Device(config-track)# ipv6 vrf VRF1
Device(config-track)# threshold metric up 254 down 255
Device(config-track)# end
```

Example: Tracking IPv6-Route Reachability

The following example shows how to configure tracking for IPv6-route reachability:

```
Device(config)# track timer ipv6 route 20
Device(config)# track 4 ipv6 route 2001:DB8:0:AB82::1/10 reachability
Device(config-track)# delay up 30
Device(config-track)# ipv6 vrf VRF2
Device(config-track)# end
```

Additional References for Object Tracking: IPv6 Route Tracking

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Object tracking	*Configuring Enhanced Object Tracking*
IP Application Services commands	*Cisco IOS IP Application Services Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

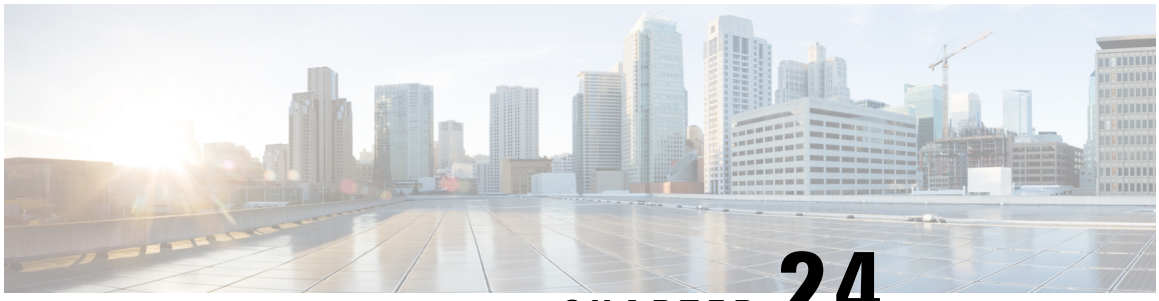
Feature Information for Object Tracking: IPv6 Route Tracking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 38: Feature Information for Object Tracking: IPv6 Route Tracking

Feature Name	Releases	Feature Information
Object Tracking: IPv6 Route Tracking		This feature expands Enhanced Object Tracking (EOT) functionality to allow the tracking of IPv6 routes.



CHAPTER 24

IPv6 Static Route Support for Object Tracking

The IPv6 Static Route Support for Object Tracking feature allows an IPv6 static route to be associated with a tracked-object. A static route is only inserted into the routing information base (RIB) when the tracked object is reachable.

This module provides an overview of the feature and explains how to configure it.

Information About IPv6 Static Route Support for Object Tracking

IPv6 Static Route Support for Object Tracking Overview

Object tracking allows you to track specific objects on a device, such as the interface line protocol state, IP routing, and route reachability, and to take action when the tracked object's state changes. Tracking allows software clients to register interest in the behavior of an object, and receive notifications of changes. This object represents the state of the system functionality such as the status of an interface (up or down), the existence of an IP prefix in the Routing Information Base (RIB) and so on.

An IPv6 static route creates a tracked object-context for each tracked object. Tracked object contexts are stored in an AVL list that is maintained by the IPv6 static route and indexed by the object number. A tracked-object context is removed from the AVL list when the object is no longer associated with any IPv6 static routes. All IPv6 static routes associated with a tracked object is linked to the tracked object context by an indirect list. An IPv6 static route becomes a client of the tracked objects, and this allows the IPv6 static route to track the state of a tracked object. The **ipv6 route** command allows an IPv6 static route to be associated with a tracked object.

Routing Table Insertion

An IPv6 static route associated with a tracked-object is inserted into the IPv6 routing table if the state of the tracked-object is up and all other routing-table-insertion criteria are met.

The IPv6 Static Route Object Tracking feature uses the IPv6 static deferred state check mechanism to insert or delete a static route into or from the Routing Information Base (RIB). A change in the state of the tracked object is signaled from tracked objects and this causes IPv6 static to insert all IPv6 static routes associated with the tracked object into the state check queue (unless they are already in it). A separate process removes IPv6 static routes from the state check queue and determines whether these routes should be inserted into the RIB or removed from the RIB using the RIB insertion criteria.

Routing Table Insertion Criteria

The following insertion criteria must be met for an IPv6 static route to be inserted into the IPv6 routing table:

1. Interface is up.
2. Next-hop address is not the device's own address.
3. Next-hop address .
4. Next-hop address is resolved.
5. Bidirectional Forwarding Detection (BFD) session is up, if BFD tracking is configured.

Note An IPv6 static route can be associated with a tracked object and a BFD session. Both tracked object and BFD session state must be up before the IPv6 static route is inserted in the routing table.

6. Tracked object state is up.

An IPv6 static route in the routing table is removed if any of the insertion criteria becomes false.

How to Configure IPv6 Static Route Support for Object Tracking

Configuring the IPv6 Static Routing Support for Object Tracking

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route vrf** *table-name-id ipv6-prefix* {*interface-type interface-number* [*next-hop-ipv6-address*] | *next-hop-ipv6-address*} [*admin-distance* [*multicast-vrf-distance*]] [**multicast**] [**nexthop-vrf** *table-name-id*] [**unicast**] [**tag** *tag-value*] [**track** *object-number*] **name***static-route*]}
4. **end**
5. **show track** *object-number*
6. **show ipv6 static vrf** *id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route vrf** *table-name-id ipv6-prefix* {*interface-type interface-number* [*next-hop-ipv6-address*] | *next-hop-ipv6-address*} [*admin-distance* [*multicast-vrf-distance*]] [**multicast**] [**nexthop-vrf** *table-name-id*] [**unicast**] [**tag** *tag-value*] [**track** *object-number*] **name***static-route*]} **Example:** `Device(config)# ipv6 route vrf 3 2001:DB8:1:2::/64 GigabitEthernet0/0 2001:DB8:3:4::1 track 42`	Establishes static IPv6 routes for all VPN routing and forwarding (VRF) instance tables or a specific VRF table for IPv4 or IPv6 address. • Configure the IPv6 static route object tracking to the static route configuration by using the **track** *object-number* command.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show track** *object-number* **Example:** `Device# show track 42`	Displays information about objects that are tracked by the tracking process.
Step 6	**show ipv6 static vrf** *id* **Example:** `Device(config)# show ipv6 static vrf 3`	Displays static routes that are added to the routing-table, and the reasons if a static route is not added.

Example

The following is sample output from the **show track** command:

```
Device# show track 42

Track 42
  IP route 10.21.12.0 255.255.255.0 reachability
  Reachability is Down (no ip route), delayed Up (1 sec remaining) (connected)
    1 change, last change 00:00:24
  Delay up 20 secs, down 10 secs
  First-hop interface is unknown (was GigabitEthernet1/0)
  Tracked by:
    HSRP GigabitEthernet0/0 3
```

Configuration Examples for IPv6 Static Route Support for Object Tracking

Example: IPv6 Static Route Object Tracking

The following example associates the static route 2001:DB8:1:2::/64 with the state of tracked-object number 42. The static route is inserted in the IPv6 routing table if the state of tracked-object number 42 is up.

```
Device> enable
Device# configure terminal
Device(config)# ipv6 route vrf 3 2001:DB8:1:2::/64 GigabitEthernet0/0 2001:DB8:3:4::1 track
 42
Device(config)# end
```

Additional References for IPv6 Static Route Support for Object Tracking

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
IP Application Services commands	*Cisco IOS IP Application Services Command Reference*
Object tracking	*Configuring Enhanced Object Tracking*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

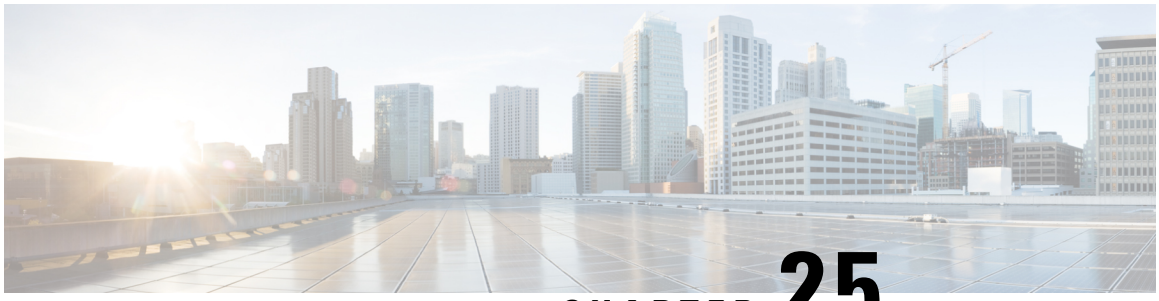
Feature Information for IPv6 Static Route Support for Object Tracking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 39: Feature Information for IPv6 Static Route Support for Object Tracking

Feature Name	Releases	Feature Information
IPv6 Static Route Support for Object Tracking		This feature expands Enhanced Object Tracking (EOT) functionality to allow the object tracking for IPv6 static routes.



CHAPTER 25

Configuring TCP

TCP is a protocol that specifies the format of data and acknowledgments used in data transfer. TCP is a connection-oriented protocol because participants must establish a connection before data can be transferred. By performing flow control and error correction, TCP guarantees reliable, in-sequence delivery of packets. TCP is considered a reliable protocol because it will continue to request an IP packet that is dropped or received out of order until it is received. This module explains concepts related to TCP and how to configure TCP in a network.

Prerequisites for TCP

TCP Time Stamp, TCP Selective Acknowledgment, and TCP Header Compression

Because TCP time stamps are always sent and echoed in both directions and the time-stamp value in the header is always changing, TCP header compression will not compress the outgoing packet. To allow TCP header compression over a serial link, the TCP time-stamp option is disabled. If you want to use TCP header compression over a serial line, TCP time stamp and TCP selective acknowledgment must be disabled. Both features are disabled by default. Use the **no ip tcp selective-ack** command to disable the TCP selective acknowledgment once it is enabled.

Information About TCP

TCP Services

TCP provides reliable transmission of data in an IP environment. TCP corresponds to the transport layer (Layer 4) of the Open Systems Interconnection (OSI) reference model. Among the services that TCP provides are stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing.

With stream data transfer, TCP delivers an unstructured stream of bytes that are identified by sequence numbers. This service benefits applications because they do not have to divide data into blocks before handing it off to TCP. Instead, TCP groups bytes into segments and passes them to IP for delivery.

TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery through an internetwork. It does this by sequencing bytes with a forwarding acknowledgment number that indicates to the destination the next byte that the source expects to receive. Bytes that are not acknowledged within a specified time period are retransmitted. The reliability mechanism of TCP allows devices to handle lost, delayed, duplicate, or misread packets. A timeout mechanism allows devices to detect lost packets and request retransmission.

TCP offers efficient flow control, which means that the receiving TCP process indicates the highest sequence number that it can receive without overflowing its internal buffers when sending acknowledgments back to the source.

TCP offers full-duplex operation, and TCP processes can both send and receive data at the same time.

TCP multiplexing allows numerous simultaneous upper-layer conversations to be multiplexed over a single connection.

TCP Connection Establishment

To use reliable transport services, TCP hosts must establish a connection-oriented session with one another. Connection establishment is performed by using a “three-way handshake” mechanism.

A three-way handshake synchronizes both ends of a connection by allowing both sides to agree upon the initial sequence numbers. This mechanism guarantees that both sides are ready to transmit data. The three-way handshake is necessary so that packets are not transmitted or retransmitted during session establishment or after session termination.

Each host randomly chooses a sequence number, which is used to track bytes within the stream that the host is sending. The three-way handshake proceeds in the following manner:

- The first host (Host A) initiates a connection by sending a packet with the initial sequence number (X) and the synchronize/start (SYN) bit set to indicate a connection request.
- The second host (Host B) receives the SYN, records the sequence number X, and replies by acknowledging (ACK) the SYN (with an ACK = X + 1). Host B includes its own initial sequence number (SEQ = Y). An ACK = 20 means that the host has received bytes 0 through 19 and expects byte 20 next. This technique is called forward acknowledgment.
- Host A acknowledges all bytes that Host B has sent with a forward acknowledgment indicating the next byte Host A expects to receive (ACK = Y + 1). Data transfer can then begin.

TCP Connection Attempt Time

You can set the amount of time the software will wait before attempting to establish a TCP connection. The connection attempt time is a host parameter and pertains to traffic that originated at the device and not to traffic going through the device. To set the TCP connection attempt time, use the **ip tcp synwait-time** command in global configuration mode. The default is 30 seconds.

TCP Selective Acknowledgment

The TCP Selective Acknowledgment feature improves performance if multiple packets are lost from one TCP window of data.

Prior to this feature, because of limited information available from cumulative acknowledgments, a TCP sender could learn about only one lost packet per-round-trip time. An aggressive sender could choose to resend packets early, but such re-sent segments might have already been successfully received.

The TCP selective acknowledgment mechanism helps improve performance. The receiving TCP host returns selective acknowledgment packets to the sender, informing the sender of data that has been received. In other words, the receiver can acknowledge packets received out of order. The sender can then resend only missing data segments (instead of everything since the first missing packet).

Prior to selective acknowledgment, if TCP lost packets 4 and 7 out of an 8-packet window, TCP would receive acknowledgment of only packets 1, 2, and 3. Packets 4 through 8 would need to be re-sent. With selective acknowledgment, TCP receives acknowledgment of packets 1, 2, 3, 5, 6, and 8. Only packets 4 and 7 must be re-sent.

TCP selective acknowledgment is used only when multiple packets are dropped within one TCP window. There is no performance impact when the feature is enabled but not used. Use the **ip tcp selective-ack** command in global configuration mode to enable TCP selective acknowledgment.

Refer to RFC 2018 for more details about TCP selective acknowledgment.

TCP Time Stamp

The TCP time-stamp option provides improved TCP round-trip time measurements. Because the time stamps are always sent and echoed in both directions and the time-stamp value in the header is always changing, TCP header compression will not compress the outgoing packet. To allow TCP header compression over a serial link, the TCP time-stamp option is disabled. Use the **ip tcp timestamp** command to enable the TCP time-stamp option.

Refer to RFC 1323 for more details on TCP time stamps.

TCP Maximum Read Size

The maximum number of characters that TCP reads from the input queue for Telnet and relogin at one time is very large (the largest possible 32-bit positive number) by default. To change the TCP maximum read size value, use the **ip tcp chunk-size** command in global configuration mode.

Note We do not recommend that you change this value.

TCP Path MTU Discovery

Path MTU Discovery is a method for maximizing the use of the available bandwidth in the network between endpoints of a TCP connection, which is described in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a device is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU that you set for the interface with the **interface** configuration

command), but the "do not fragment" (DF) bit is set. The intermediate gateway sends a "Fragmentation needed and DF bit set" Internet Control Message Protocol (ICMP) message to the sending host, alerting the host to the problem. On receiving this message, the host reduces its assumed path MTU and consequently sends a smaller packet that will fit the smallest packet size of all links along the path.

By default, TCP Path MTU Discovery is disabled. Existing connections are not affected irrespective of whether this feature is enabled or disabled.

Customers using TCP connections to move bulk data between systems on distinct subnets would benefit most by enabling this feature. Customers using remote source-route bridging (RSRB) with TCP encapsulation, serial tunnel (STUN), X.25 Remote Switching (also known as XOT or X.25 over TCP), and some protocol translation configurations might also benefit from enabling this feature.

Use the **ip tcp path-mtu-discovery** global configuration command to enable Path MTU Discovery for connections initiated by the device when the device is acting as a host.

For more information about Path MTU Discovery, refer to the "Configuring IP Services" module of the *IP Application Services Configuration Guide*.

TCP Window Scaling

The TCP Window Scaling feature adds support for the Window Scaling option in RFC 1323,*TCP Extensions for High Performance*. A larger window size is recommended to improve TCP performance in network paths with large bandwidth-delay product characteristics that are called Long Fat Networks (LFNs). The TCP Window Scaling enhancement provides LFN support.

The window scaling extension expands the definition of the TCP window to 32 bits and then uses a scale factor to carry this 32-bit value in the 16-bit window field of the TCP header. The window size can increase to a scale factor of 14. Typical applications use a scale factor of 3 when deployed in LFNs.

The TCP Window Scaling feature complies with RFC 1323. The maximum window size was increased to 1,073,741,823 bytes. The larger scalable window size will allow TCP to perform better over LFNs. Use the **ip tcp window-size** command in global configuration mode to configure the TCP window size.

TCP Sliding Window

A TCP sliding window provides an efficient use of network bandwidth because it enables hosts to send multiple bytes or packets before waiting for an acknowledgment.

In TCP, the receiver specifies the current window size in every packet. Because TCP provides a byte-stream connection, window sizes are expressed in bytes. A window is the number of data bytes that the sender is allowed to send before waiting for an acknowledgment. Initial window sizes are indicated at connection setup, but might vary throughout the data transfer to provide flow control. A window size of zero means "Send no data." The default TCP window size is 4128 bytes. From Cisco IOS XE 17.14.1a, the default TCP window size is 131072 bytes. Use the ip tcp window-size command to change the default window size.

Note In the presence of optimizations (**ip tcp ack-tuning** and **ip tcp winupdate-opt**) which are enabled by default from 17.10.1, the value 131072 will be applied unless it is configured otherwise. Although, when default keyword is used for the **ip tcp window-size** command prior to version 17.14.1a, the value 4128 will take effect.

In a TCP sliding-window operation, for example, the sender might have a sequence of bytes to send (numbered 1 to 10) to a receiver who has a window size of five. The sender then places a window around the first five bytes and transmits them together. The sender then waits for an acknowledgment.

The receiver responds with an ACK = 6, indicating that it has received bytes 1 to 5 and is expecting byte 6 next. In the same packet, the receiver indicates that its window size is 5. The sender then moves the sliding window five bytes to the right and transmits bytes 6 to 10. The receiver responds with an ACK = 11, indicating that it is expecting sequenced byte 11 next. In this packet, if the receiver indicates that its window size is 0, the sender cannot send any more bytes until the receiver sends another packet with a window size greater than 0.

TCP Outgoing Queue Size

The default TCP outgoing queue size per connection is five segments if the connection has a TTY associated with it (such as a Telnet connection). If no TTY connection is associated with a connection, the default queue size is 20 segments. Use the **ip tcp queuemax** command to change the five-segment default value.

TCP MSS Adjustment

The TCP MSS Adjustment feature enables the configuration of the maximum segment size (MSS) for transient packets that traverse a device, specifically TCP segments with the SYN bit set. Use the **ip tcp adjust-mss** command in interface configuration mode to specify the MSS value on the intermediate device of the SYN packets to avoid truncation.

When a host (usually a PC) initiates a TCP session with a server, the host negotiates the IP segment size by using the MSS option field in the TCP SYN packet. The value of the MSS field is determined by the MTU configuration on the host. The default MSS value for a PC is 1500 bytes.

The PPP over Ethernet (PPPoE) standard supports a Maximum Transmission Unit (MTU) of only 1492 bytes. The disparity between the host and PPPoE MTU size can cause the device in between the host and the server to drop 1500-byte packets and terminate TCP sessions over the PPPoE network. Even if the path MTU (which detects the correct MTU across the path) is enabled on the host, sessions may be dropped because system administrators sometimes disable ICMP error messages that must be relayed from the host for path MTU to work.

The **ip tcp adjust-mss** command helps prevent TCP sessions from being dropped by adjusting the MSS value of the TCP SYN packets.

The **ip tcp adjust-mss** command is effective only for TCP connections passing through the device.

In most cases, the optimum value for the *max-segment-size* argument of the **ip tcp adjust-mss** command is 1452 bytes. This value plus the 20-byte IP header, the 20-byte TCP header, and the 8-byte PPPoE header add up to a 1500-byte packet that matches the MTU size for the Ethernet link.

See the “Configuring the MSS Value and MTU for Transient TCP SYN Packets” section for configuration instructions.

TCP Applications Flags Enhancement

The TCP Applications Flags Enhancement feature enables the user to display additional flags with reference to TCP applications. There are two types of flags: status and option. The status flags indicate the status of TCP connections such as passive open, active open, retransmission timeout, and app closed for listening. The additional flags indicate the state of set options such as whether a VPN routing and forwarding instance (VRF)

is set, whether a user is idle, and whether a keepalive timer is running. Use the **show tcp** command to display TCP application flags.

TCP Show Extension

The TCP Show Extension feature introduces the capability to display addresses in IP format instead of the hostname format and to display the VRF table associated with the connection. To display the status for all endpoints with addresses in IP format, use the **show tcp brief numeric** command.

TCP MIB for RFC 4022 Support

The TCP MIB for RFC 4022 Support feature introduces support for RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*. RFC 4022 is an incremental change of the TCP MIB to improve the manageability of TCP.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

Zero-Field TCP Packets

Prior to Cisco IOS XE Release 2.5, when a zero-field TCP packet is received on the router, the TCP packet counter is incremented.

In Cisco IOS XE Release 2.5 and later releases, when a zero-field TCP packet is received on the router, the TCP packet counter is not incremented.

When a zero-field TCP packet is received, it is displayed as 0 under the TCP statistics field when the **show ip traffic** command is configured. When the debug **ip tcp packet** command is configured, and a zero-field TCP packet is received, a debug message similar to the following is displayed:

```
Jan 19 21:57:28.487: TCP: Alert! Received a segment with cleared flags
10.4.14.49
```

How to Configure TCP

Configuring TCP Performance Parameters

Before you begin

Both sides of the network link must be configured to support window scaling or the default of 65,535 bytes will be applied as the maximum window size. To support Explicit Congestion Notification (ECN), the remote peer must be ECN-enabled because the ECN capability is negotiated during a three-way handshake with the remote peer.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip tcp synwait-time** *seconds*
4. **ip tcp path-mtu-discovery** [**age-timer** {*minutes* | **infinite**}]
5. **ip tcp selective-ack**
6. **ip tcp timestamp**
7. **ip tcp chunk-size** *characters*
8. **ip tcp window-size** *bytes*
9. **ip tcp ecn**
10. **ip tcp queuemax** *packets*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip tcp synwait-time** *seconds* **Example:** `Device(config)# ip tcp synwait-time 60`	(Optional) Sets the amount of time the Cisco software will wait before attempting to establish a TCP connection. • The default is 30 seconds.
Step 4	**ip tcp path-mtu-discovery** [**age-timer** {*minutes* | **infinite**}] **Example:** `Device(config)# ip tcp path-mtu-discovery age-timer 11`	(Optional) Enables Path MTU Discovery. • **age-timer** —Time interval, in minutes, TCP reestimates the Maximum Transmission Unit (MTU) with a larger Maximum Segment Size (MSS). The default is 10 minutes. The maximum is 30 minutes. • **infinite**—Disables the age timer.
Step 5	**ip tcp selective-ack** **Example:** `Device(config)# ip tcp selective-ack`	(Optional) Enables TCP selective acknowledgment.
Step 6	**ip tcp timestamp** **Example:** `Device(config)# ip tcp timestamp`	(Optional) Enables the TCP time stamp.
Step 7	**ip tcp chunk-size** *characters* **Example:**	(Optional) Sets the TCP maximum read size for Telnet or rlogin.

	Command or Action	Purpose
	`Device(config)# ip tcp chunk-size 64000`	**Note** We do not recommend that you change this value.
Step 8	**ip tcp window-size** *bytes* **Example:** `Device(config)# ip tcp window-size 75000`	(Optional) Sets the TCP window size. • The bytes argument can be set to an integer from 68 to 1073741823. To enable window scaling to support Long Flat Networks (LFNs), the TCP window size must be more than 65535. The default window size from Cisco IOS XE 17.14.1a is 131072 if window scaling is not configured. Although, when default keyword is used for the **ip tcp window-size** command prior to version 17.14.1a, the value 4128 will take effect if window scaling is not configured. **Note** With CSCsw45317, the *bytes* argument can be set to an integer from 68 to 1073741823.
Step 9	**ip tcp ecn** **Example:** `Device(config)# ip tcp ecn`	(Optional) Enables ECN for TCP.
Step 10	**ip tcp queuemax** *packets* **Example:** `Device(config)# ip tcp queuemax 10`	(Optional) Sets the TCP outgoing queue size.
Step 11	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring the MSS Value and MTU for Transient TCP SYN Packets

Perform this task to configure the maximum size segment (MSS) for transient packets that traverse a device, specifically TCP segments with the SYN bit set, and to configure the MTU size of IP packets.

If you are configuring the **ip mtu** command on the same interface as the **ip tcp adjust-mss** command, we recommend that you use the following commands and values:

- **ip tcp adjust-mss 1452**
- **ip mtu 1492**

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip tcp adjust-mss** *max-segment-size*

5. **ip mtu** *bytes*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip tcp adjust-mss** *max-segment-size* **Example:** `Device(config-if)# ip tcp adjust-mss 1452`	Adjusts the MSS value of TCP SYN packets going through a device. • The *max-segment-size* argument is the maximum segment size, in bytes. The range is from 500 to 1460.
Step 5	**ip mtu** *bytes* **Example:** `Device(config-if)# ip mtu 1492`	Sets the MTU size of IP packets, in bytes, sent on an interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits to global configuration mode.

Configuring the MSS Value for IPv6 Traffic

Perform this task to configure the maximum size segment (MSS) for transient packets that traverse a device, specifically TCP segments with the DF bit set in IPv6 network layer (IP) header.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 tcp adjust-mss** *max-segment-size*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ipv6 tcp adjust-mss** *max-segment-size* **Example:** `Device(config-if)# ipv6 tcp adjust-mss 1452`	Adjusts the MSS value of TCP DF packets going through a device. • The *max-segment-size* argument is the maximum segment size, in bytes. The range is from 40 to 1940.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying TCP Performance Parameters

SUMMARY STEPS

1. **show tcp** [*line-number*] [**tcb** *address*]
2. **show tcp brief** [**all** | **numeric**]
3. **debug ip tcp transactions**
4. **debug ip tcp congestion**

DETAILED STEPS

Step 1 **show tcp** [*line-number*] [**tcb** *address*]

Displays the status of TCP connections. The arguments and keyword are as follows:

- *line-number*—(Optional) Absolute line number of the Telnet connection status.
- **tcb**—(Optional) Transmission control block (TCB) of the Explicit Congestion Notification (ECN)-enabled connection.
- *address*—(Optional) TCB hexadecimal address. The valid range is from 0x0 to 0xFFFFFFFF.

The following sample output from the **show tcp tcb** command displays detailed information about an ECN-enabled connection that uses a hexadecimal address format:

Example:

```
Device# show tcp tcb 0x62CD2BB8

Connection state is LISTEN, I/O status: 1, unread input bytes: 0
Connection is ECN enabled
Local host: 10.10.10.1, Local port: 179
Foreign host: 10.10.10.2, Foreign port: 12000
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x4F31940):
Timer          Starts    Wakeups            Next
Retrans             0          0             0x0
TimeWait            0          0             0x0
AckHold             0          0             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss:           0 snduna:           0 sndnxt:           0     sndwnd:      0
irs:           0 rcvnxt:           0 rcvwnd:       131072  delrcvwnd:      0
SRTT: 0 ms, RTTO: 2000 ms, RTV: 2000 ms, KRTT: 0 ms
minRTT: 60000 ms, maxRTT: 0 ms, ACK hold: 200 ms
Flags: passive open, higher precedence, retransmission timeout
TCB is waiting for TCP Process (67)
Datagrams (max data segment is 516 bytes):
Rcvd: 6 (out of order: 0), with data: 0, total data bytes: 0
Sent: 0 (retransmit: 0, fastretransmit: 0), with data: 0, total data
bytes: 0
```

Cisco Software Modularity

The following sample output from the **show tcp tcb** command displays a Software Modularity image:

Example:

```
Device# show tcp tcb 0x1059C10

Connection state is ESTAB, I/O status: 0, unread input bytes: 0
Local host: 10.4.2.32, Local port: 23
Foreign host: 10.4.2.39, Foreign port: 11000
VRF table id is: 0
Current send queue size: 0 (max 65536)
Current receive queue size: 0 (max 32768)  mis-ordered: 0 bytes
Event Timers (current time is 0xB9ACB9):
Timer          Starts    Wakeups            Next(msec)
Retrans             6          0                 0
SendWnd             0          0                 0
TimeWait            0          0                 0
AckHold             8          4                 0
KeepAlive          11          0           7199992
PmtuAger            0          0                 0
GiveUp              0          0                 0
Throttle            0          0                 0
irs:    1633857851  rcvnxt: 1633857890  rcvadv: 1633890620  rcvwnd:  32730
iss:    4231531315  snduna: 4231531392  sndnxt: 4231531392  sndwnd:   4052
sndmax: 4231531392  sndcwnd:     10220
SRTT: 84 ms,  RTTO: 650 ms,  RTV: 69 ms,  KRTT: 0 ms
minRTT: 0 ms,  maxRTT: 200 ms, ACK hold: 200 ms
Keepalive time: 7200 sec, SYN wait time: 75 sec
Giveup time: 0 ms, Retransmission retries: 0, Retransmit forever: FALSE
State flags: none
Feature flags: Nagle
Request flags: none
Window scales: rcv 0, snd 0, request rcv 0, request snd 0
Timestamp option: recent 0, recent age 0, last ACK sent          0
```

```
Datagrams (in bytes): MSS 1460, peer MSS 1460, min MSS 1460, max MSS 1460
Rcvd: 14 (out of order: 0), with data: 10, total data bytes: 38
Sent: 10 (retransmit: 0, fastretransmit: 0), with data: 5, total data bytes: 76
Header prediction hit rate: 72 %
Socket states: SS_ISCONNECTED, SS_PRIV
Read buffer flags: SB_WAIT, SB_SEL, SB_DEL_WAKEUP
Read notifications: 4
Write buffer flags: SB_DEL_WAKEUP
Write notifications: 0
Socket status: 0
```

Step 2 **show tcp brief** [**all** | **numeric**]

(Optional) Displays addresses in IP format.

Use the **show tcp brief** command to display a concise description of TCP connection endpoints. Use the optional **all** keyword to display the status for all endpoints with addresses in a Domain Name System (DNS) hostname format. If this keyword is not used, endpoints in the LISTEN state are not shown. Use the optional **numeric** keyword to display the status for all endpoints with addresses in IP format.

Note If the **ip domain-lookup** command is enabled on the device, and you execute the **show tcp brief** command, the response time of the device to display the output will be very slow. To get a faster response, you should disable the **ip domain-lookup** command.

The following is sample output from the **show tcp brief** command while a user is connected to the system by using Telnet:

Example:

```
Device# show tcp brief

TCB       Local Address           Foreign Address        (state)
609789AC  Device.cisco.com.23     cider.cisco.com.3733   ESTAB
```

The following example shows the IP activity after the **numeric** keyword is used to display addresses in IP format:

Example:

```
Device# show tcp brief numeric

TCB           Local Address          Foreign Address     (state)
6523A4FC      10.1.25.3.11000        10.1.25.3.23         ESTAB
65239A84      10.1.25.3.23           10.1.25.3.11000      ESTAB
653FCBBC      *.1723 *.* LISTEN
```

Step 3 **debug ip tcp transactions**

Use the **debug ip tcp transactions** command to display information about significant TCP transactions such as state changes, retransmissions, and duplicate packets. The TCP/IP network isolated above the data link layer might encounter performance issues. The **debug ip tcp transactions** command can be useful in debugging these performance issues.

The following is sample output from the **debug ip tcp transactions** command:

Example:

```
Device# debug ip tcp transactions

TCP: sending SYN, seq 168108, ack 88655553
TCP0: Connection to 10.9.0.13:22530, advertising MSS 966
TCP0: state was LISTEN -> SYNRCVD [23 -> 10.9.0.13(22530)]
TCP0: state was SYNSENT -> SYNRCVD [23 -> 10.9.0.13(22530)]
TCP0: Connection to 10.9.0.13:22530, received MSS 956
TCP0: restart retransmission in 5996
TCP0: state was SYNRCVD -> ESTAB [23 -> 10.9.0.13(22530)]
```

```
TCP2: restart retransmission in 10689
TCP2: restart retransmission in 10641
TCP2: restart retransmission in 10633
TCP2: restart retransmission in 13384 -> 10.0.0.13(16151)]
TCP0: restart retransmission in 5996 [23 -> 10.0.0.13(16151)]
```

The following line from the **debug ip tcp transactions** command sample output shows that TCP has entered Fast Recovery mode:

Example:

```
fast re-transmit - sndcwnd - 512, snd_last - 33884268765
```

The following lines from the **debug ip tcp transactions** command sample output show that a duplicate acknowledgment is received when TCP is in Fast Recovery mode (first line) and a partial acknowledgment has been received (second line):

Example:

```
TCP0:ignoring second congestion in same window sndcwn - 512, snd_1st - 33884268765
TCP0:partial ACK received sndcwnd:338842495
```

Step 4 **debug ip tcp congestion**

Use the **debug ip tcp congestion** command to display information about TCP congestion events. The TCP/IP network isolated above the data link layer might encounter performance issues. The **debug ip tcp congestion** command can be used to debug these performance issues. The command also displays information related to variations in the TCP send window, congestion window, and congestion threshold window.

The following is sample output from the **debug ip tcp congestion** command:

Example:

```
Device# debug ip tcp congestion

*May 20 22:49:49.091: Setting New Reno as congestion control algorithm
*May 22 05:21:47.281: Advance cwnd by 12
*May 22 05:21:47.281: TCP85FD0C10: sndcwnd: 1472
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1475
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1478
*May 22 05:21:47.285: Advance cwnd by 9
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1487
*May 20 22:50:32.559: [New Reno] sndcwnd: 8388480 ssthresh: 65535 snd_mark: 232322
*May 20 22:50:32.559: 10.168.10.10:42416 <---> 10.168.30.11:49100 congestion window changes
*May 20 22:50:32.559: cwnd from 8388480 to 2514841, ssthresh from 65535 to 2514841
```

For Cisco TCP, New Reno is the default congestion control algorithm. However, an application can also use Binary Increase Congestion Control (BIC) as the congestion control algorithm. The following is sample output from the **debug ip tcp congestion** command using BIC:

Example:

```
Device# debug ip tcp congestion

*May 22 05:21:42.281: Setting BIC as congestion control algorithm
*May 22 05:21:47.281: Advance cwnd by 12
*May 22 05:21:47.281: TCP85FD0C10: sndcwnd: 1472
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1475
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1478
```

```
*May 22 05:21:47.285: Advance cwnd by 9
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1487
*May 20 22:50:32.559: [BIC] sndcwnd: 8388480 ssthresh: 65535 bic_last_max_cwnd: 0 last_cwnd: 8388480
*May 20 22:50:32.559: 10.168.10.10:42416 <---> 10.168.30.11:49100 congestion window changes
*May 20 22:50:32.559: cwnd from 8388480 to 2514841, ssthresh from 65535 to 2514841
*May 20 22:50:32.559: bic_last_max_cwnd changes from 0 to 8388480
```

Configuration Examples for TCP

Example: Verifying the Configuration of TCP ECN

The following example shows how to verify whether TCP ECN is configured:

```
Device# show running-config

Building configuration...
.
.
.
ip tcp ecn ! ECN is configured.
.
.
.
```

The following example shows how to verify whether TCP is ECN-enabled on a specific connection (local host):

```
Device# show tcp tcb 123456A

!Local host
!
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Enabled
Local host: 10.1.25.31, Local port: 11002
Foreign host: 10.1.25.34, Foreign port: 23
```

The following example shows how to display concise information about one address:

```
Device# show tcp brief

!
TCB          Local address            Foreign Address        (state)
609789C      Router.example.com.23      cider.example.com.3733    ESTAB
```

The following example shows how to enable IP TCP ECN debugging:

```
Device# debug ip tcp ecn
!
TCP ECN debugging is on
!
Device# telnet 10.1.25.31

Trying 10.1.25.31 ...
!
01:43:19: 10.1.25.35:11000 <---> 10.1.25.31:23   out ECN-setup SYN
```

```
01:43:21: 10.1.25.35:11000 <---> 10.1.25.31:23   congestion window changes
01:43:21: cwnd from 1460 to 1460, ssthresh from 65535 to 2920
01:43:21: 10.1.25.35:11000 <---> 10.1.25.31:23   in non-ECN-setup SYN-ACK
```

Before a TCP connection can use ECN, a host sends an ECN-setup SYN (synchronization) packet to a remote end that contains an Echo Congestion Experience (ECE) and Congestion window reduced (CWR) bit set in the header. Setting the ECE and CWR bits indicates to the remote end that the sending TCP is ECN capable, rather than an indication of congestion. The remote end sends an ECN-setup SYN-ACK (acknowledgment) packet to the sending host.

In this example the “out ECN-setup SYN” text means that a SYN packet with the ECE and CWR bit set was sent to the remote end. The “in non-ECN-setup SYN-ACK” text means that the remote end did not favorably acknowledge the ECN request and, therefore, the session is not ECN capable.

The following output shows that ECN capabilities are enabled at both ends. In response to the ECN-setup SYN, the other end favorably replied with an ECN-setup SYN-ACK message. This connection is now ECN capable for the rest of the session.

```
Device# telnet 10.10.10.10

Trying 10.10.10.10 ... Open
Password required, but none set
!
1d20h: 10.1.25.34:11003 <---> 10.1.25.35:23   out ECN-setup SYN
1d20h: 10.1.25.34:11003 <---> 10.1.25.35:23   in ECN-setup SYN-ACK
```

The following example shows how to verify that the hosts are connected:

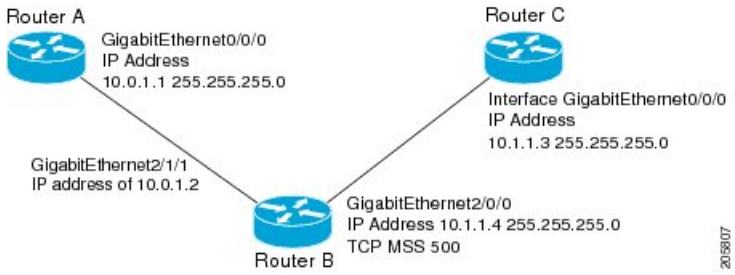
```
Device# show debugging
!
TCP:
  TCP Packet debugging is on
  TCP ECN debugging is on
!
Device# telnet 10.1.25.234
!
Trying 10.1.25.234 ...
!
00:02:48: 10.1.25.31:11001 <---> 10.1.25.234:23   out ECN-setup SYN
00:02:48: tcp0: O CLOSED 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 131072
00:02:50: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:02:50: cwnd from 1460 to 1460, ssthresh from 65535 to 2920
00:02:50: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 131072
00:02:54: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:02:54: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:02:54: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 131072
00:03:02: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:02: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:02: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 131072
00:03:18: 10.1.25.31:11001 <---> 10.1.25.234:23   SYN with ECN disabled
00:03:18: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:18: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:18: tcp0: O SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 SYN  WIN 131072
00:03:20: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:20: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:20: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
```

```
        OPTS 4 SYN  WIN 131072
00:03:24: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:24: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:24: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 SYN  WIN 131072
00:03:32: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:32: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:32: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 SYN  WIN 131072
!Connection timed out; remote host not responding
```

Example: Configuring the TCP MSS Adjustment

The following example shows how to configure and verify the interface adjustment value for the example topology displayed in the figure below:

Figure 29: Example Topology for TCP MSS Adjustment



Configure the interface adjustment value on router B:

```
Router_B(config)# interface GigabitEthernet 2/0/0
Router_B(config-if)# ip tcp adjust-mss 500
```

Telnet from router A to router C with B having the Maximum Segment Size (MSS) adjustment configured:

```
Router_A# telnet 192.168.1.1

Trying 192.168.1.1... Open
```

Observe the debug output from router C:

```
Router_C# debug ip tcp transactions

Sep 5 18:42:46.247: TCP0: state was LISTEN -> SYNRCVD [23 -> 10.0.1.1(38437)]
Sep 5 18:42:46.247: TCP: tcb 32290C0 connection to 10.0.1.1:38437, peer MSS 500, MSS is 500
Sep 5 18:42:46.247: TCP: sending SYN, seq 580539401, ack 6015751
Sep 5 18:42:46.247: TCP0: Connection to 10.0.1.1:38437, advertising MSS 500
Sep 5 18:42:46.251: TCP0: state was SYNRCVD -> ESTAB [23 -> 10.0.1.1(38437)]
```

The MSS gets adjusted to 500 on Router B as configured.

The following example shows the configuration of a Point-to-Point Protocol over Ethernet (PPPoE) client with the MSS value set to 1452:

```
Device(config)# vpdn enable
Device(config)# no vpdn logging
Device(config)# vpdn-group 1
Device(config-vpdn)# request-dialin
```

```
Device(config-vpdn-req-in)# protocol pppoe
Device(config-vpdn-req-in)# exit
Device(config-vpdn)# exit
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 192.168.100.1.255.255.255.0
Device(config-if)# ip tcp adjust-mss 1452
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface ATM 0
Device(config-if)# no ip address
Device(config-if)# no atm ilmi-keepalive
Device(config-if)# pvc 8/35
Device(config-if)# pppoe client dial-pool-number 1
Device(config-if)# dsl equipment-type CPE
Device(config-if)# dsl operating-mode GSHDSL symmetric annex B
Device(config-if)# dsl linerate AUTO
Device(config-if)# exit
Device(config)# interface Dialer 1
Device(config-if)3 ip address negotiated
Device(config-if)# ip mtu 1492
Device(config-if)# ip nat outside
Device(config-if)# encapsulation ppp
Device(config-if)# dialer pool 1
Device(config-if)# dialer-group 1
Device(config-if)# ppp authentication pap callin
Device(config-if)# ppp pap sent-username sohodyn password 7 141B1309000528
Device(config-if)# ip nat inside source list 101 Dialer1 overload
Device(config-if)# exit
Device(config)# ip route 0.0.0.0.0.0.0.0 Dialer1
Device(config)# access-list permit ip 192.168.100.0.0.0.0.255 any
```

The following example shows the configuration of interface adjustment value for IPv6 traffic:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/0
Device(config)# ipv6 tcp adjust-mss 1452
Device(config)# end
```

Example: Configuring the TCP Application Flags Enhancement

The following output shows the flags (status and option) displayed using the **show tcp** command:

```
Device# show tcp
.
.
.
Status Flags: passive open, active open, retransmission timeout
 App closed
Option Flags: vrf id set
IP Precedence value: 6
.
.
.
SRTT: 273 ms, RTTO: 490 ms, RTV: 217 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
```

Example: Displaying Addresses in IP Format

The following example shows the IP activity by using the **numeric** keyword to display the addresses in IP format:

```
Device# show tcp brief numeric

TCB           Local Address          Foreign Address     (state)
6523A4FC      10.1.25.3.11000        10.1.25.3.23         ESTAB
65239A84      10.1.25.3.23           10.1.25.3.11000      ESTAB
653FCBBC      *.1723 *.* LISTEN
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP Application Services commands	IP Application Services Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 793	Transmission Control Protocol
RFC 1191	Path MTU discovery
RFC 1323	TCP Extensions for High Performance
RFC 2018	TCP Selective Acknowledgment Options
RFC 2581	TCP Congestion Control
RFC 3168	The Addition of Explicit Congestion Notification (ECN) to IP
RFC 3782	The NewReno Modification to TCP's Fast Recovery Algorithm
RFC 4022	Management Information Base for the Transmission Control Protocol (TCP)

MIBs

MIB	MIBs Link
CISCO-TCP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for TCP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

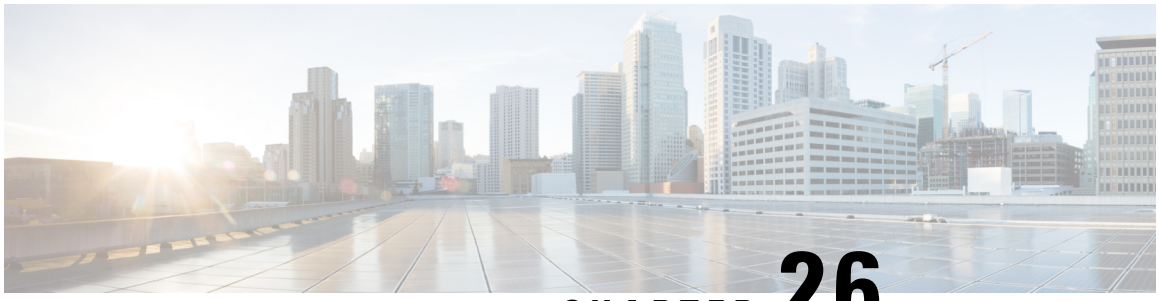
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 40: Feature Information for TCP

Feature Name	Releases	Feature Information
TCP Application Flags Enhancement	12.2(31)SB2 12.4(2)T	The TCP Applications Flags Enhancement feature enables the user to display additional flags with reference to TCP applications. There are two types of flags: status and option. The status flags indicate the status of TCP connections such as retransmission timeouts, application closed, and synchronized (SYNC) handshakes for listening. The additional flags indicate the state of set options such as whether a VPN routing and forwarding instance (VRF) is set, whether a user is idle, and whether a keepalive timer is running. The following command was modified by this feature: **show tcp**.

Feature Name	Releases	Feature Information
TCP Congestion Avoidance	12.3(7)T	The TCP Congestion Avoidance feature enables the monitoring of acknowledgment packets to the TCP sender when multiple packets are lost in a single window of data. Before this feature was introduced, the sender would exit Fast-Recovery mode, wait for three or more duplicate acknowledgment packets before retransmitting the next unacknowledged packet, or wait for the retransmission timer to start slowly. This delay could lead to performance issues. Implementation of RFC 2581 and RFC 3782 addresses the modifications to the Fast-Recovery algorithm that incorporates a response to partial acknowledgments received during Fast Recovery, improving performance in situations where multiple packets are lost in a single window of data. This feature is an enhancement to the existing Fast Recovery algorithm. No commands are used to enable or disable this feature. The output of the **debug ip tcp transactions** command monitors acknowledgment packets by displaying the following conditions: • TCP entering Fast Recovery mode. • Duplicate acknowledgments being received during Fast Recovery mode. • Partial acknowledgments being received. The following command was modified by this feature: **debug ip tcp transactions**.
TCP Explicit Congestion Notification	12.3(7)T	The TCP Explicit Congestion Notification (ECN) feature allows an intermediate router to notify end hosts of impending network congestion. It also provides enhanced support for TCP sessions associated with applications such as Telnet, web browsing, and transfer of audio and video data, that are sensitive to delay or packet loss. The benefit of this is the reduction of delay and packet loss in data transmissions. The following commands were introduced or modified by this feature: **debug ip tcp ecn**, **ip tcp ecn**, **show debugging**, **show tcp**.
TCP MIB for RFC4022 Support	12.2(33)XN	The TCP MIB for RFC 4022 Support feature introduces support for RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*. RFC 4022 is an incremental change of the TCP MIB to improve the manageability of TCP. There are no new or modified commands for this feature.

Feature Name	Releases	Feature Information
TCP MSS Adjust	12.2(4)T 12.2(8)T 12.2(18)ZU2 12.2(28)SB 12.2(33)SRA 12.2(33)SXH 15.0(1)S	The TCP MSS Adjust feature enables the configuration of the maximum segment size (MSS) for transient packets that traverse a device, specifically TCP segments in the SYN bit set. In 12.2(4)T, this feature was introduced. In 12.2(8)T, the command that was introduced by this feature was changed from **ip adjust-mss** to **ip tcp adjust-mss**. In 12.2(28)SB and 12.2(33)SRA, this feature was enhanced to be configurable on subinterfaces. The following command was introduced by this feature: **ip tcp adjust-mss**.
TCP Show Extension	12.2(31)SB2 12.4(2)T	The TCP Show Extension feature introduces the capability to display addresses in IP format instead of hostname format and to display the VRF table associated with the connection. The following command was modified by this feature: **show tcp brief**.
TCP Window Scaling	12.2(8)T 12.2(31)SB2	The TCP Window Scaling feature adds support for the Window Scaling option in RFC 1323. A larger window size is recommended to improve TCP performance in network paths with large bandwidth, long-delay characteristics that are called Long Fat Networks (LFNs). This TCP Window Scaling enhancement provides that support. The following command was introduced or modified by this feature: **ip tcp window-size**.
TCP Keepalive Timer	15.2(4)M	The TCP Keepalive Timer feature introduces the capability to identify dead connections between multiple routing devices. The following command was introduced or modified by this feature: **ip tcp keepalive**.



CHAPTER 26

Configuring WCCP

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that intercepts IP packets and redirects those packets to a destination other than that specified in the IP packet. Typically the packets are redirected from their destination web server on the Internet to a content engine that is local to the client. In some WCCP deployment scenarios, redirection of traffic may also be required from the web server to the client. WCCP enables you to integrate content engines into your network infrastructure.

Cisco IOS Release 12.1 and later releases allow the use of either WCCP Version 1 (WCCPv1) or Version 2 (WCCPv2).

The tasks in this document assume that you have already configured content engines on your network. For specific information on hardware and network planning associated with Cisco Content Engines and WCCP, see the Cisco Content Engines documentation at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/webscale/content/index.htm

Prerequisites for WCCP

- To use WCCP, IP must be configured on the interface connected to the Internet and another interface must be connected to the content engine.
- The interface connected to the content engine must be a Fast Ethernet or Gigabit Ethernet interface.

Restrictions for WCCP

General

The following limitations apply to Web Cache Communication Protocol Version1 (WCCPv1) and WCCP Version 2 (WCCPv2):

- WCCP works only with IPv4 networks.
- WCCP bypasses Network Address Translation (NAT) when Cisco Express Forwarding is enabled.

WCCPv1

- WCCPv1 supports the redirection of HTTP (TCP port 80) traffic only.
- WCCPv1 does not allow multiple routers to be attached to a cluster of content engines.

WCCPv2

- WCCP works only with IPv4 networks.
- For routers servicing a multicast cluster, the Time To Live (TTL) value must be set at 15 or fewer.
- Service groups can comprise up to 32 content engines and 32 routers.
- All content engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Multicast addresses must be from 224.0.0.0 to 239.255.255.255.

WCCP VRF Support

- In Cisco IOS Release 12.2(33)SRE, this feature is supported only on Cisco 7200 NPE-G2 and Cisco 7304-NPE-G100 routers.

WCCP Layer 2 Forwarding and Return

In a Cisco Wide Area Application Services (WAAS) and Cisco IOS XE firewall configuration, all packets processed by a Wide Area Application Engine (WAE) device must go over the Cisco IOS XE firewall in both directions to support the Web Cache Coordination Protocol (WCCP) generic routing encapsulation (GRE) redirect. This situation occurs when a Layer 2 redirect is not available. If a Layer 2 redirect is configured on the WAE, the system defaults to the GRE redirect to continue to function.

The client device and a WAE device or a cache engine cannot be connected to a Cisco device with the same interface and WCCP redirect configured on the interface.

The following two configurations are supported:

- For WCCP Layer 2 return, the client and WAE are connected to a Cisco device with same interface and WCCP output is configured on the interface.
- For WCCP Layer 2 return, the client and WAE are connected to a Cisco device with same physical interface but in different VLANs and sub-interfaces.

Cisco 7600 Series Routers Access Control Lists

When WCCP is using the mask assignment, any redirect list is merged with the mask information from the appliance and the resulting merged ACL is passed down to the Cisco 7600 series router hardware. Only Permit or Deny ACL entries from the redirect list in which the protocol is IP or exactly matches the service group protocol are merged with the mask information from the appliance.

The following restrictions apply to the redirect-list ACL:

- The ACL must be an IPv4 simple or extended ACL.
- Only individual source or destination port numbers may be specified; port ranges cannot be specified.
- The only valid matching criteria in addition to individual source or destination port numbers are **dscp** or **tos**.
- The use of **fragments**, **time-range**, or **options** keywords, or any TCP flags is not permitted.
- If the redirect ACL does not meet the restrictions shown, the system will log the following error message:

```
WCCP-3-BADACE: Service <service group>, invalid access-list entry (seq:<sequence>,
reason:<reason>)
```

- WCCP continues to redirect packets, but the redirection is carried out in software until the access list is adjusted.

Information About WCCP

WCCP Overview

WCCP uses Cisco Content Engines (or other content engines running WCCP) to localize traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables Cisco IOS XE routing platforms to transparently redirect content requests. With transparent redirection, users can fulfill content requests locally without configuring their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a content engine. The word "transparent" in this case means that the end user does not know that a requested file (such as a web page) came from the content engine instead of from the originally specified server.

A content engine receiving a request attempts to service it from its own local cache. If the requested information is not present, the content engine issues its own request to the originally targeted server to get the required information. A content engine retrieving the requested information forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of content engines, called a content engine cluster, to provide content to a router or multiple routers. Network administrators can easily scale their content engines to manage heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each cluster member to work in parallel, resulting in linear scalability. Clustering content engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 content engines to scale to your desired capacity.

Layer 2 Forwarding Redirection and Return

WCCP uses either generic routing encapsulation (GRE) or Layer 2 (L2) to redirect or return IP traffic. When WCCP forwards traffic via GRE, the redirected packets are encapsulated within a GRE header. The packets also have a WCCP redirect header. When WCCP forwards traffic using L2, the original MAC header of the IP packet is overwritten and replaced with the MAC header for the WCCP client.

Using L2 as a forwarding method allows direct forwarding to the content engine without further lookup. Layer 2 redirection requires that the router and content engines are directly connected, that is, on the same IP subnetwork.

When WCCP returns traffic via GRE, the returned packets are encapsulated within a GRE header. The destination IP address is the address of the router and the source address is the address of the WCCP client. When WCCP returns traffic via L2, the original IP packet is returned without any added header information. The router to which the packet is returned will recognize the source of the packet and prevent redirection.

The WCCP redirection method does not have to match the return method.

L2 forwarding, return, or redirection are typically used for hardware-accelerated platforms. Depending on your release, L2 forwarding, return, and redirection can also be used for software-switching platforms.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **l2-redirect** keyword to configure L2 redirection. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **l2-redirect** keyword to configure L2 redirection.

Note Before configuring a GRE tunnel, configure a loopback interface (that is not attached to a VRF) with an IP address so that the internally created tunnel interface is enabled for IPv4 forwarding by unnumbering itself to this dummy loopback interface. You do not need to configure a loopback interface if the system has at least one interface that is not attached to a VRF and that is configured with an IPv4 address.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

For more information about WAAS commands used to configure Cisco Content Engines, see the Cisco Wide Area Application Services Command Reference.

WCCP Mask Assignment

The WCCP Mask Assignment feature enables mask assignment as the load-balancing method (instead of the default hash assignment method) for a WCCP service.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **mask-assign** keyword to configure mask assignment. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **mask-assign** keyword to configure mask assignment.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

For more information about WAAS commands used to configure Cisco Content Engines, see the Cisco Wide Area Application Services Command Reference.

Hardware Acceleration

Cisco 7600 series routers provide WCCP Layer 2 Policy Feature Card (PFC) redirection hardware acceleration. Hardware acceleration allows Cisco Content Engines to perform a L2 MAC address rewrite redirection method when directly connected to a compatible router.

Redirection processing is accelerated in the routing hardware, which is more efficient than L3 redirection with Generic Routing Encapsulation (GRE). L2 redirection takes place on the router, and is not visible to the Multilayer Switch Feature Card (MSFC). The WCCP L2 PFC redirection feature requires no configuration on the MSFC. The **show ip wccp** {*service-number* | **web-cache**} **detail** command displays which redirection method is in use for each content engine.

In order for the router to make complete use of hardware redirection, the content engine must be configured with L2 redirection and mask assignment.

Use the **ip wccp web-cache accelerated** command on hardware-based platforms to enforce the use of L2 redirection and mask assignment. Using this command configures the router to form a service group and redirect packets with an appliance only if the appliance is configured for L2 and mask assignment.

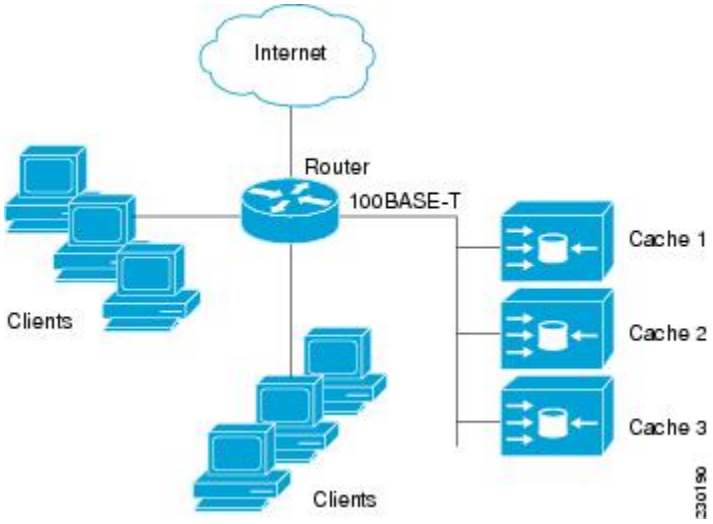
The following guidelines apply to WCCP Layer 2 PFC redirection:

- The WCCP Layer 2 PFC redirection feature sets the IP flow mask to full-flow mode.
- You can configure the Cisco Cache Engine software Release 2.2 or later releases to use the WCCP Layer 2 PFC redirection feature.
- L2 redirection takes place on the PFC and is not visible to the MSFC. The **show ip wccp** {*service-number* | **web-cache**} **detail** command on the MSFC displays statistics for only the first packet of an L2 redirected flow, which provides an indication of how many flows, rather than packets, are using L2 redirection. You can view information about L2 redirected flows by entering the **show platform flow ip** command. The PFC3 provides hardware acceleration for GRE. If you use WCCP Layer 3 redirection with GRE, there is hardware support for encapsulation, but the PFC3 does not provide hardware support for decapsulation of WCCP GRE traffic.

WCCPv1 Configuration

With WCCPv1, only a single router services a cluster. In this scenario, this router is the device that performs all the IP packet redirection. The figure below illustrates the WCCPv1 configuration.

Figure 30: WCCPv1 Configuration



Content is not duplicated on the content engines. The benefit of using multiple content engines is that you can scale a caching solution by clustering multiple physical content engines to appear as one logical cache.

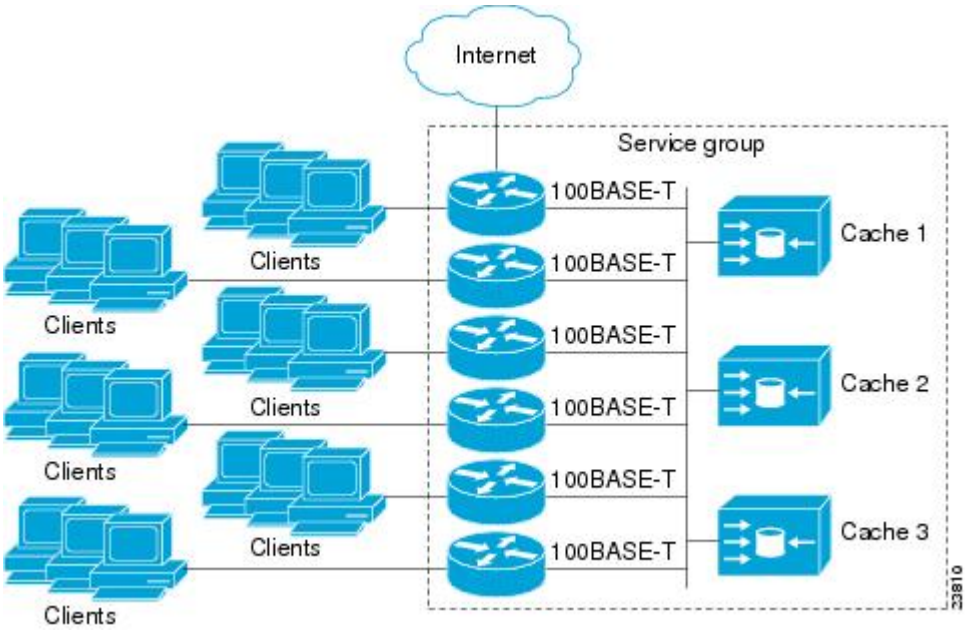
The following sequence of events details how WCCPv1 configuration works:

1. Each content engine is configured by the system administrator with the IP address of the control router. Up to 32 content engines can connect to a single control router.
2. The content engines send their IP addresses to the control router using WCCP, indicating their presence. Routers and content engines communicate to each other via a control channel; this channel is based on UDP port 2048.
3. This information is used by the control router to create a cluster view (a list of caches in the cluster). This view is sent to each content engine in the cluster, essentially making all the content engines aware of each other. A stable view is established after the membership of the cluster remains the same for a certain amount of time.
4. When a stable view has been established, one content engine is elected as the lead content engine. (The lead is defined as the content engine seen by all the content engines in the cluster with the lowest IP address). This lead content engine uses WCCP to indicate to the control router how IP packet redirection should be performed. Specifically, the lead content engine designates how redirected traffic should be distributed across the content engines in the cluster.

WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router could redirect content requests to a cluster. The figure below illustrates a sample configuration using multiple routers.

Figure 31: Cisco Content Engine Network Configuration Using WCCPv2



The subset of content engines within a cluster and routers connected to the cluster that are running the same service is known as a service group. Available services include TCP and UDP redirection.

In WCCPv1, the content engines were configured with the address of the single router. WCCPv2 requires that each content engine be aware of all the routers in the service group. To specify the addresses of all the routers in a service group, you must choose one of the following methods:

- Unicast—A list of router addresses for each of the routers in the group is configured on each content engine. In this case the address of each router in the group must be explicitly specified for each content engine during configuration.
- Multicast—A single multicast address is configured on each content engine. In the multicast address method, the content engine sends a single-address notification that provides coverage for all routers in the service group. For example, a content engine could indicate that packets should be sent to a multicast address of 224.0.0.100, which would send a multicast packet to all routers in the service group configured for group listening using WCCP (see the **ip wccp group-listen** or the **ipv6 wccp group-listen** interface configuration command for details).

The multicast option is easier to configure because you need only specify a single address on each content engine. This option also allows you to add and remove routers from a service group dynamically, without needing to reconfigure the content engines with a different list of addresses each time.

The following sequence of events details how WCCPv2 configuration works:

1. Each content engine is configured with a list of routers.
2. Each content engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of content engines in the group.
3. When the view is consistent across all content engines in the cluster, one content engine is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

WCCPv2 Support for Services Other Than HTTP

WCCPv2 allows redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. WCCPv2 supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and Real Audio, video, and telephony applications.

To accommodate the various types of services available, WCCPv2 introduced the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as 98) or a predefined service keyword (such as **web-cache**). This information is used to validate that service group members are all using or providing the same service.

The content engines in a service group specify traffic to be redirected by protocol (TCP or UDP) and up to eight source or destination ports. Each service group has a priority status assigned to it. The priority of a dynamic service is assigned by the content engine. The priority value is in the range of 0 to 255 where 0 is the lowest priority. The predefined web-cache service has an assigned priority of 240.

WCCPv2 Support for Multiple Routers

WCCPv2 allows multiple routers to be attached to a cluster of cache engines. The use of multiple routers in a service group allows for redundancy, interface aggregation, and distribution of the redirection load. WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the Hashed Message Authentication Code—Message

Digest (HMAC MD5) standard. Shared-secret MD5 one-time authentication (set using the **ip wccp** [**password** [**0** | **7**] *password*] global configuration command) enables messages to be protected against interception, inspection, and replay.

WCCPv2 Web Cache Packet Return

If a content engine is unable to provide a requested object it has cached due to error or overload, the content engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the content engine unserviced. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the content engine cluster). This process provides error handling transparency to clients.

Typical reasons why a content engine would reject packets and initiate the packet return feature include the following:

- Instances when the content engine is overloaded and has no room to service the packets
- Instances when the content engine is filtering for certain conditions that make caching packets counterproductive (for example, when IP authentication has been turned on)

WCCPv2 Load Distribution

WCCPv2 can be used to adjust the load being offered to individual content engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. WCCPv2 allows the designated content engine to adjust the load on a particular content engine and balance the load across the content engines in a cluster. WCCPv2 uses three techniques to perform load distribution:

- Hot spot handling—Allows an individual hash bucket to be distributed across all the content engines. Prior to WCCPv2, information from one hash bucket could go to only one content engine.
- Load balancing—Allows the set of hash buckets assigned to a content engine to be adjusted so that the load can be shifted from an overwhelmed content engine to other members that have available capacity.
- Load shedding—Enables the router to selectively redirect the load to avoid exceeding the capacity of a content engine.

The use of these hashing parameters prevents one content engine from being overloaded and reduces the potential for bottlenecking.

WCCP VRF Support

The WCCP VRF Support feature enhances the WCCPv2 protocol by implementing support for virtual routing and forwarding (VRF).

The WCCP VRF Support feature allows service groups to be configured on a per-VRF basis in addition to those defined globally.

Along with the service identifier, the VRF of WCCP protocol packets arriving at the router is used to associate cache-engines with a configured service group.

The same VRF must have the interface on which redirection is applied, the interface which is connected to cache engine, and the interface on which the packet would have left if it had not been redirected.

WCCP VRF Tunnel Interfaces

In Cisco IOS XE releases that support the WCCP VRF Support feature, the use of GRE redirection results in the creation of new tunnel interfaces. You can display these tunnel interfaces by entering the **show ip interface brief | include tunnel** command:

```
Device# show ip interface brief | include tunnel

Tunnel0                172.16.0.1      YES unset  up                    up
Tunnel1                172.16.0.1      YES unset  up                    up
Tunnel2                172.16.0.1      YES unset  up                    up
Tunnel3                172.16.0.1      YES unset  up                    up
Device#
```

The tunnel interfaces are automatically created in order to process outgoing GRE-encapsulated traffic for WCCP. The tunnel interfaces appear when a content engine connects and requests GRE redirection. The tunnel interfaces are not created directly by WCCP, but are created indirectly via a tunnel application programming interface (API). WCCP does not have direct knowledge of the tunnel interfaces, but can redirect packets to them, resulting in the appropriate encapsulation being applied to the packets. After the appropriate encapsulation is applied, the packet is then sent to the content engine.

Note The tunnel interfaces are not used to connect with incoming WCCP GRE return packets.

One tunnel is created for each service group that is using GRE redirection. One additional tunnel is created to provide an IP address that allows the other tunnel group interfaces to be unnumbered but still enabled for IPv4.

You can confirm the connection between the tunnels and WCCP by entering the **show tunnel groups wccp** command:

```
Device# show tunnel groups wccp

WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
   intf: Tunnel0, locally sourced
 WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
   intf: Tunnel3, locally sourced
 WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
   intf: Tunnel2, locally sourced
```

You can display additional information about each tunnel interface by entering the **show tunnel interface** *interface-number* command:

```
Device# show tunnel interface t0

Tunnel0
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 10.1.1.80
   Application ID 2: WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up

Device# show tunnel interface t1

Tunnel1
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 172.16.0.1
   Application ID 2: unspecified
   Linestate - current up
```

```
   Internal linestate - current up, evaluated up

Device# show tunnel interface t2

Tunnel2
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 10.1.1.80
   Application ID 2: WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up

Device# show tunnel interface t3

Tunnel3
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 10.1.1.80
   Application ID 2: WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up
Device#
```

Note that the service group number shown in the examples is the internal tunnel representation of the WCCP service group number. Group 0 is the web-cache service. To determine the dynamic services, subtract 256 from the displayed service group number to convert to the WCCP service group number. For interfaces that are used for redirection, the source address shown is the WCCP router ID.

You can display information about the connected content engines and encapsulation, including software packet counters, by entering the **show adjacency** [*tunnel-interface*] [**encapsulation**] [**detail**] [**internal**] command:

```
Device# show adjacency t0

Protocol Interface                 Address
IP       Tunnel0                   10.1.1.82(3)

Device# show adjacency t0 encapsulation

Protocol Interface                 Address
IP       Tunnel0                   10.1.1.82(3)
  Encap length 28
  45000000000000000FF2F7D2B1E010150
  1E0101520000883E00000000
  Provider: TUNNEL
  Protocol header count in macstring: 3
    HDR 0: ipv4
       dst: static, 10.1.1.82
       src: static, 10.1.1.80
      prot: static, 47
       ttl: static, 255
        df: static, cleared
      per packet fields: tos ident tl chksm
    HDR 1: gre
      prot: static, 0x883E
      per packet fields: none
    HDR 2: wccpv2
       dyn: static, cleared
      sgID: static, 0
      per packet fields: alt altB priB

Device# show adjacency t0 detail

Protocol Interface                 Address
IP       Tunnel0                   10.1.1.82(3)
                                   connectionid 1
                                   0 packets, 0 bytes
                                   epoch 0
```

```
                                    sourced in sev-epoch 1
                                    Encap length 28
                                    4500000000000000FF2F7D2B1E010150
                                    1E0101520000883E00000000
                                    Tun endpt
                                    Next chain element:
                                     IP adj out of Ethernet0/0, addr 10.1.1.82
Device# show adjacency t0 internal

Protocol Interface                  Address
IP       Tunnel0                    10.1.1.82(3)
                                    connectionid 1
                                    0 packets, 0 bytes
                                    epoch 0
                                    sourced in sev-epoch 1
                                    Encap length 28
                                    4500000000000000FF2F7D2B1E010150
                                    1E0101520000883E00000000
                                    Tun endpt
                                    Next chain element:
                                     IP adj out of Ethernet0/0, addr 10.1.1.82
                                     parent oce 0x4BC76A8
                                     frame originated locally (Null0)
                                    L3 mtu 17856
                                    Flags (0x2808C4)
                                    Fixup enabled (0x40000000)
                                          GRE WCCP redirection
                                    HWIDB/IDB pointers 0x55A13E0/0x35F5A80
                                    IP redirect disabled
                                    Switching vector: IPv4 midchain adj oce
                                    IP Tunnel stack to 10.1.1.82 in Default (0x0)
                                     nh tracking enabled: 10.1.1.82/32
                                     IP adj out of Ethernet0/0, addr 10.1.1.82
                                    Adjacency pointer 0x4BC74D8
                                    Next-hop 10.1.1.82
Device#
```

WCCP Bypass Packets

WCCP intercepts IP packets and redirects those packets to a destination other than the destination that is specified in the IP header. Typically the packets are redirected from a web server on the Internet to a web cache that is local to the destination.

Occasionally a web cache cannot manage the redirected packets appropriately and returns the packets unchanged to the originating router. These packets are called bypass packets and are returned to the originating router using either Layer 2 forwarding without encapsulation (L2) or encapsulated in generic routing encapsulation (GRE). The router decapsulates and forwards the packets normally. The VRF associated with the ingress interface (or the global table if there is no VRF associated) is used to route the packet to the destination.

GRE is a tunneling protocol developed by Cisco that encapsulates packet types from a variety of protocols inside IP tunnels, creating a virtual point-to-point link over an IP network.

WCCP Closed Services and Open Services

In applications where packets are intercepted and redirected by a Cisco IOS router to external WCCP client devices, it may be necessary to block the packets for the application when a WCCP client device is not available. This blocking is achieved by configuring a WCCP closed service. When a WCCP service is configured as closed, the packets that fulfill the services, but do not have an active client device, are discarded.

By default, WCCP operates as an open service, wherein communication between clients and servers proceeds normally in the absence of an intermediary device.

The **ip wccp service-list** or the **ipv6 wccp service-list** command can be used for both closed-mode and open-mode services. Use the **service-list** keyword and *service-access-list* argument to register an application protocol type or port number. Use the **mode** keyword to select an open or closed service.

WCCP Outbound ACL Check

When WCCP is enabled for redirection on an ingress interface, the packets are redirected by WCCP and instead egress on an interface other than the destination that is specified in the IP header. The packets are still subject to ACLs configured on the ingress interface. However, redirection can cause the packets to bypass the ACL configured on the original egress interface. Packets that would have been dropped because of the ACL configured on the original egress interface can be sent out on the redirect egress interface, which poses a possible security problem. Enabling the WCCP Outbound ACL check feature ensures that redirected packets are subject to any ACL conditions configured on the original egress interface.

WCCP Service Groups

WCCP is a component of Cisco IOS software that redirects traffic with defined characteristics from its original destination to an alternative destination. The typical application of WCCP is to redirect traffic bound for a remote web server to a local web cache to improve response time and optimize network resource usage.

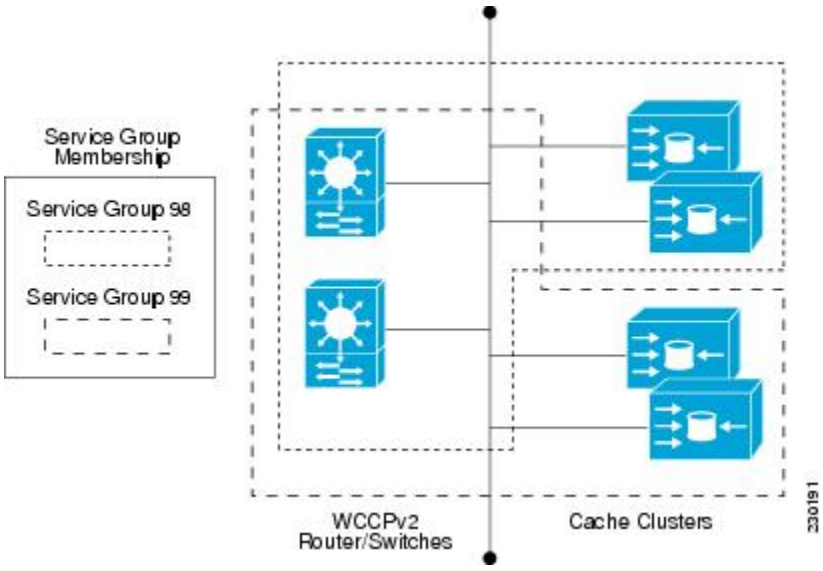
The nature of the selected traffic for redirection is defined by service groups (see figure below) specified on content engines and communicated to routers by using WCCP. The maximum number of service groups allowed across all VRFs is 256.

WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is web cache, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the content engines. This service is referred to as a *well-known service*, because the characteristics of the web cache service are known by both the router and content engines. A description of a well-known service is not required beyond a service identification. To specify the standard web cache service, use the **ip wccp** or the **ipv6 wccp** command with the **web-cache** keyword.

Note More than one service can run on a router at the same time, and routers and content engines can be part of multiple service groups at the same time.

Figure 32: WCCP Service Groups



The dynamic services are defined by the content engines; the content engine instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group's traffic, because this information is provided by the first content engine to join the group. In a dynamic service, up to eight ports can be specified within a single protocol.

Cisco Content Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other content engine devices may use this service number for some other service.

WCCP—Check All Services

An interface may be configured with more than one WCCP service. When more than one WCCP service is configured on an interface, the precedence of a service depends on the relative priority of the service compared to the priority of the other configured services. Each WCCP service has a priority value as part of its definition. When an interface is configured with more than one WCCP service, the precedence of the packets is matched against service groups in priority order.

Note The priority of a WCCP service group cannot be configured via Cisco IOS software.

With the **ip wccp check services all** or the **ipv6 wccp check services all** command, WCCP can be configured to check all configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by a redirect ACL and by the service priority. The **ip wccp check services all** commands must be configured at global level to support multiple WCCP services.

If no WCCP services are configured with a redirect ACL, the services are considered in priority order until a service is found that matches the IP packet. If no services match the packet, the packet is not redirected. If a service matches the packet and the service has a redirect ACL configured, then the IP packet will be checked against the ACL. If the packet is rejected by the ACL, the packet will not be passed down to lower priority services unless the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured. When the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured, WCCP

will continue to attempt to match the packet against any remaining lower priority services configured on the interface.

WCCP Interoperability with NAT

To redirect traffic using WCCP to a router running WAAS software that is also configured with NAT, enable the **ip nat inside** or the **ipv6 nat inside** command on the WAAS interface. If you are not able to configure the **ip nat inside** or the**ipv6 nat inside** command on the WAAS interface, disable Cisco Express Forwarding. You must also update the WCCP redirect ACL to include a private address to ensure that pretranslated traffic is redirected.

WCCP Troubleshooting Tips

CPU usage may be very high when WCCP is enabled. The WCCP counters enable a determination of the bypass traffic directly on the router and can indicate whether the cause is high CPU usage due to enablement of WCCP. In some situations, 10 percent bypass traffic may be normal; in other situations, 10 percent may be high. However, any figure above 25 percent should prompt a closer investigation of what is occurring in the web cache.

If the counters suggest that the level of bypass traffic is high, the next step is to examine the bypass counters in the content engine and determine why the content engine is choosing to bypass the traffic. You can log in to the content engine console and use the CLI to investigate further. The counters allow you to determine the percent of traffic being bypassed.

You can use the **clear ipv6 wccp***service-id* command to remove the IPv6 WCCP statistics (counts) maintained on the router for a particular service.

You can use the **clear wccp** command to remove all (IPv4 and IPv6) WCCP statistics (counts) maintained on the router for a particular service.

You can use the **show ipv6 wccp** command to display the IPv6 WCCP global statistics (counts).

You can use the **show wccp** command to display all (IPv4 and IPv6) WCCP global statistics (counts).

How to Configure WCCP

The following configuration tasks assume that you have already installed and configured the content engines you want to include in your network. You must configure the content engines in the cluster before configuring WCCP functionality on your routers or switches. Refer to the Cisco Cache Engine User Guide for content engine configuration and setup tasks.

Configuring WCCP

Perform this task to configure WCCP.

Until you configure a WCCP service using the **ip wccp** {**web-cache** | *service-number*} global configuration command, WCCP is disabled on the device. The first use of a form of the **ip wccp** command enables WCCP. By default WCCPv2 is used for services, but you can use WCCPv1 functionality instead. To change the running version of WCCP from Version 2 to Version 1, or to return to WCCPv2 after an initial change, use the **ip wccp version** command in global configuration mode.

If a function is not allowed in WCCPv1, an error prompt will be printed to the screen. For example, if WCCPv1 is running on the device and you try to configure a dynamic service, the following message will be displayed: "WCCP V1 only supports the web-cache service." The **show ip wccp** EXEC command will display the WCCP protocol version number that is running on your device.

Use the **ip wccp web-cache password** command to set a password for a device and the content engines in a service group. MD5 password security requires that each device and content engine that wants to join a service group be configured with the service group password. The password must be up to eight characters in length. Each content engine or device in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp version** {**1** | **2**}
4. **ip wccp** {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password* [**0** | **7**]]
5. **interface** *type number*
6. **ip wccp** {**web-cache** | *service-number*} **redirect** {**in** | **out**}
7. **exit**
8. **interface** *type number*
9. **ip wccp redirect exclude in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip wccp version** {**1**	**2**} **Example:** `Device(config)# ip wccp version 2`	Specifies which version of WCCP to configure on a device. • WCCPv2 is the default running version.
Step 4	**ip wccp** {**web-cache**	*service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password* [**0**	**7**]] **Example:** `Device(config)# ip wccp web-cache password pwd`	Specifies a web-cache or dynamic service to enable on a device, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service. • **Note** The password length must not exceed 8 characters.

	Command or Action	Purpose
Step 5	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 0/0`	Targets an interface number for which the web cache service will run, and enters interface configuration mode.
Step 6	**ip wccp** {**web-cache** | *service-number*} **redirect** {**in** | **out**} **Example:** `Device(config-if)# ip wccp web-cache redirect in`	Enables packet redirection on an outbound or inbound interface using WCCP. • As indicated by the **out** and **in** keyword options, redirection can be specified for outbound interfaces or inbound interfaces.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/2/0`	Targets an interface number on which to exclude traffic for redirection, and enters interface configuration mode.
Step 9	**ip wccp redirect exclude in** **Example:** `Device(config-if)# ip wccp redirect exclude in`	(Optional) Excludes traffic on the specified interface from redirection.

Configuring Closed Services

Perform this task to specify the number of service groups for WCCP, to configure a service group as a closed or open service, and to optionally specify a check of all services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip wccp** [**vrf** *vrf-name*] *service-number* [**service-list** *service-access-list* **mode** {**open** | **closed**}]
   - or
   - **ip wccp** [**vrf** *vrf-name*] **web-cache mode** {**open** | **closed**}
4. **ip wccp check services all**
5. **ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip wccp** [**vrf** *vrf-name*] *service-number* [**service-list** *service-access-list* **mode** {**open** | **closed**}]
	• or
	• **ip wccp** [**vrf** *vrf-name*] **web-cache mode** {**open** | **closed**}
	Example:
	`Device(config)# ip wccp 90 service-list 120 mode closed`
	or
	`Device(config)# ip wccp web-cache mode closed`
		Configures a dynamic WCCP service as closed or open.
		or
		Configures a web-cache service as closed or open.
		Note When configuring the web-cache service as a closed service, you cannot specify a service access list.
		Note When configuring a dynamic WCCP service as a closed service, you must specify a service access list.

Step 4	**ip wccp check services all** **Example:** `Device(config)# ip wccp check services all`	(Optional) Enables a check of all WCCP services. • Use this command to configure WCCP to check the other configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by the redirect ACL and not just the service description. **Note** The **ip wccp check services all** command is a global WCCP command that applies to all services and is not associated with a single service.
Step 5	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **Example:** `Device(config)# ip wccp 201`	Specifies the WCCP service identifier. • You can specify the standard web-cache service or a dynamic service number from 0 to 255. • The maximum number of services that can be specified is 256.
Step 6	**exit** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# exit`	

Registering a Router to a Multicast Address

If you decide to use the multicast address option for your service group, you must configure the router to listen for the multicast broadcasts on an interface.

For network configurations where redirected traffic needs to traverse an intervening router, the router being traversed must be configured to perform IP multicast routing. You must configure the following two components to enable traversal over an intervening router:

- Enable IP multicast routing using the **ip multicast-routing** global configuration command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ip wccp group-listen** interface configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**vrf** *vrf-name*] [**distributed**]
4. **ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-address** *multicast-address*
5. **interface** *type number*
6. **ip pim** {**sparse-mode** | **sparse-dense-mode** | **dense-mode** [**proxy-register** {**list** *access-list* | **route-map** *map-name*}]}
7. **ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-listen**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**vrf** *vrf-name*] [**distributed**] **Example:** `Device(config)# ip multicast-routing`	Enables IP multicast routing.
Step 4	**ip wccp** [**vrf** *vrf-name*] {**web-cache**	*service-number*} **group-address** *multicast-address*	Specifies the multicast address for the service group.

	Command or Action	Purpose
	Example: `Device(config)# ip wccp 99 group-address 239.1.1.1`	
Step 5	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Enables the interfaces to which the content engines will connect to receive multicast transmissions for which the web cache service will run, and enters interface configuration mode.
Step 6	**ip pim** {**sparse-mode** | **sparse-dense-mode** | **dense-mode** [**proxy-register** {**list** *access-list* | **route-map** *map-name*}]} **Example:** `Device(config-if)# ip pim dense-mode`	(Optional) Enables Protocol Independent Multicast (PIM) on an interface. **Note** To ensure correct operation of the **ip wccp group-listen** command on Cisco 7600 series routers, you must enter the **ip pim** command in addition to the **ip wccp group-listen** command.
Step 7	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-listen** **Example:** `Device(config-if)# ip wccp 99 group-listen`	Configures an interface to enable or disable the reception of IP multicast packets for WCCP.

Using Access Lists for a WCCP Service Group

Perform this task to configure the device to use an access list to determine which traffic should be directed to which content engines.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**]
5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} | [**log**]
7. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
8. **ip wccp** [**vrf** *vrf-name*] **web-cache group-list** *access-list*
9. **ip wccp** [**vrf** *vrf-name*] **web-cache redirect-list** *access-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 1 remark Give access to user1`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0`	Creates an access list that enables or disables traffic redirection to the cache engine and permits the specified source based on a source address and wildcard mask. • Every access list needs at least one permit statement; it does not need to be the first entry. • Standard IP access lists are numbered 1 to 99 or 1300 to 1999. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.5.22 is allowed to pass the access list.
Step 5	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 1 remark Give access to user1`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 6	**access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} | [**log**] **Example:** `Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0`	Denies the specified source based on a source address and wildcard mask. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the abbreviation any as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255.

	Command or Action	Purpose
		• In this example, host 172.16.7.34 is denied passing the access list.
Step 7	Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**ip wccp** [**vrf** *vrf-name*] **web-cache group-list** *access-list* **Example:** `Device(config) ip wccp web-cache group-list 1`	Indicates to the device from which IP addresses of content engines to accept packets.
Step 9	**ip wccp** [**vrf** *vrf-name*] **web-cache redirect-list** *access-list* **Example:** `Device(config)# ip wccp web-cache redirect-list 1`	(Optional) Disables caching for certain clients.

Enabling the WCCP Outbound ACL Check

Note When all redirection is performed in the hardware, the mode of redirection will change when outbound ACL checking is enabled. The first packet is switched in software to allow the extra ACL check to be performed before a shortcut is installed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*]
4. **ip wccp check acl outbound**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*] **Example:** `Device(config)# ip wccp web-cache`	Enables the support for a Cisco content engine service group or any content engine service group and configures a redirect ACL list or group ACL. **Note** The **web-cache** keyword is for WCCP version 1 and version 2 and the *service-number* argument is for WCCP version 2 only.
Step 4	**ip wccp check acl outbound** **Example:** `Device(config)# ip wccp check acl outbound`	Checks the access control list (ACL) for egress interfaces for packets redirected by WCCP.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration.

Enabling WCCP Interoperability with NAT

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **ip wccp** *service-number* **redirect in**
6. **exit**
7. **interface** *type number*
8. **ip nat outside**
9. **ip wccp** *service-number* **redirect in**
10. **exit**
11. **interface** *type number*
12. **ip nat inside**
13. **ip wccp redirect exclude in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface on which to enable NAT and enters interface configuration mode. • This is the LAN-facing interface.
Step 4	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation).
Step 5	**ip wccp** *service-number* **redirect in** **Example:** `Router(config-if)# ip wccp 61 redirect in`	Enables packet redirection on an inbound interface using WCCP.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Router(config)# interface ethernet 2`	Specifies an interface on which to enable NAT and enters interface configuration mode. • This is the WAN-facing interface.
Step 8	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the outside network.
Step 9	**ip wccp** *service-number* **redirect in** **Example:** `Router(config-if)# ip wccp 62 redirect in`	Enables packet redirection on an inbound interface using WCCP.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:**	Specifies an interface on which to enable NAT and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface ethernet 3`	• This is the WAAS-facing interface.
Step 12	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Designates that traffic originating from or destined for the interface is subject to NAT and indicates that the interface is connected to the inside network (the network subject to NAT translation).
Step 13	**ip wccp redirect exclude in** **Example:** `Router(config-if)# ip wccp redirect exclude in`	Configures an interface to exclude packets received on an interface from being checked for redirection..

Verifying and Monitoring WCCP Configuration Settings

SUMMARY STEPS

1. **enable**
2. **show ip wccp** [**web-cache** |*service-number*] [**detail view**]
3. **show ip interface**
4. **more system:running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip wccp** [**web-cache** |*service-number*] [**detail view**] **Example:** `Device# show ip wccp 24 detail`	Displays global information related to WCCP, including the protocol version running, the number of content engines in the router service group, which content engine group is allowed to connect to the router, and which access list is being used. • *service-number*—(Optional) Dynamic number of the web-cache service group being controlled by the content engine. The range is from 0 to 99. For web caches that use Cisco Content Engines, the reverse proxy service is indicated by a value of 99. • **web-cache**—(Optional) statistics for the web-cache service. • **detail**—(Optional) other members of a particular service group or web cache that have or have not been detected.

	Command or Action	Purpose
		• **view**—(Optional) information about a router or all web caches.
Step 3	**show ip interface** **Example:** `Device# show ip interface`	Displays status about whether any **ip wccp redirection** commands are configured on an interface; for example, "Web Cache Redirect is enabled / disabled."
Step 4	**more system:running-config** **Example:** `Device# more system:running-config`	(Optional) Displays contents of the running configuration file (equivalent to the **show running-config** command).

Configuration Examples for WCCP

Example: Changing the Version of WCCP on a Router

The following example shows how to change the WCCP version from the default of WCCPv2 to WCCPv1, and enabling the web-cache service in WCCPv1:

```
Device# show ip wccp

% WCCP version 2 is not enabled
Device# configure terminal

Device(config)# ip wccp version 1

Device(config)# end
Device# show ip wccp

% WCCP version 1 is not enabled
Device# configure terminal

Device(config)# ip wccp web-cache
Device(config)# end
Device# show ip wccp

Global WCCP information:
    Router information:
        Router Identifier:                  10.4.9.8
        Protocol Version:                   1.0
.
.
.
```

Example: Configuring a General WCCPv2 Session

```
Device# configure terminal
Device(config)# ip wccp web-cache group-address 224.1.1.100 password password
```

```
Device(config)# ip wccp source-interface GigabitEthernet 0/1/0
Device(config)# ip wccp check services all
! Configures a check of all WCCP services.
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# ip wccp redirect exclude in
Device(config-if)# exit
```

Example: Setting a Password for a Router and Content Engines

```
Router# configure terminal
Router(config)# ip wccp web-cache password password1
```

Example: Configuring a Web Cache Service

```
Device# configure terminal
Device(config)# ip wccp web-cache
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
Device(config-if)# exit
Device# copy running-config startup-config
```

The following example shows how to configure a session in which redirection of HTTP traffic arriving on Gigabit Ethernet interface 0/1/0 is enabled:

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
Device(config-if)# exit
Device# show ip interface GigabitEthernet 0/1/0
.
.
.
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
.
.
.
```

Example: Running a Reverse Proxy Service

The following example assumes that you are configuring a service group using Cisco cache engines, which use dynamic service 99 to run a reverse proxy service:

```
Router# configure terminal
Router(config)# ip wccp 99
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# ip wccp 99 redirect out
```

Example: Registering a Router to a Multicast Address

```
Device# configure terminal
Device(config)# ip wccp web-cache group-address 224.1.1.100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip wccp web cache group-listen
```

The following example shows a router configured to run a reverse proxy service, using the multicast address of 224.1.1.1. Redirection applies to packets outgoing via Gigabit Ethernet interface 0/1/0:

```
Device# configure terminal
Device(config)# ip wccp 99 group-address 224.1.1.1
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip wccp 99 redirect out
```

Example: Using Access Lists

To achieve better security, you can use a standard access list to notify the device which IP addresses are valid addresses for a content engine attempting to register with the current device. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```
Device(config)# access-list 10 permit host 10.1.1.1
Device(config)# access-list 10 permit host 10.1.1.2
Device(config)# access-list 10 permit host 10.1.1.3
Device(config)# ip wccp web-cache group-list 10
```

To disable caching for certain clients, servers, or client/server pairs, you can use WCCP access lists. The following example shows that any requests coming from 10.1.1.1 to 10.3.1.1 will bypass the cache, and that all other requests will be serviced normally:

```
Device(config)# ip wccp web-cache redirect-list 120
Device(config)# access-list 120 deny tcp host 10.1.1.1 any
Device(config)# access-list 120 deny tcp any host 10.3.1.1
Device(config)# access-list 120 permit ip any any
```

The following example configures a device to redirect web-related packets received via Gigabit Ethernet interface 0/1/0, destined to any host except 209.165.200.224:

```
Device(config)# access-list 100 deny ip any host 209.165.200.224
Device(config)# access-list 100 permit ip any any
Device(config)# ip wccp web-cache redirect-list 100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip wccp web-cache redirect in
```

Example: WCCP Outbound ACL Check Configuration

The following configuration example shows that the access list prevents traffic from network 10.0.0.0 leaving Gigabit Ethernet interface 0/1/0. Because the outbound ACL check is enabled, WCCP does not redirect that traffic. WCCP checks packets against the ACL before they are redirected.

```
Device(config)# ip wccp web-cache
Device(config)# ip wccp check acl outbound
Device(config)# interface gigabitethernet 0/1/0
```

```
Device(config-if)# ip access-group 10 out
Device(config-if)# exit
Device(config)# ip wccp web-cache redirect-list redirect-out
Device(config)# access-list 10 deny 10.0.0.0 0.255.255.255
Device(config)# access-list 10 permit any
```

If the outbound ACL check is disabled, the HTTP packets from network 10.0.0.0 would be redirected to a web cache. Users with that network address could retrieve web pages even though the network administrator wanted to prevent it.

Example: Verifying WCCP Settings

The following example shows how to verify your configuration changes by using the **more system:running-config** command in privileged EXEC mode. The following example shows that both the web cache service and dynamic service 99 are enabled on the device:

```
Device# more system:running-config

 Building configuration...
 Current configuration:
 !
 version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 service udp-small-servers
 service tcp-small-servers
 !
 hostname router4
 !
 enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTh1
 enable password password1
 !
 ip subnet-zero
 ip wccp web-cache
 ip wccp 99
 ip domain-name cisco.com
 ip name-server 10.1.1.1
 ip name-server 10.1.1.2
 ip name-server 10.1.1.3
 !
 !
 !
 interface GigabitEthernet0/1/1
 ip address 10.3.1.2 255.255.255.0
 no ip directed-broadcast
 ip wccp web-cache redirect in
 ip wccp 99 redirect in
 no ip route-cache
 no ip mroute-cache
 !
 interface GigabitEthernet0/1/0
 ip address 10.4.1.1 255.255.255.0
 no ip directed-broadcast
 ip wccp 99 redirect in
 no ip route-cache
 no ip mroute-cache
 !
 interface Serial0
 no ip address
 no ip directed-broadcast
```

```
 no ip route-cache
 no ip mroute-cache
 shutdown
 !
 interface Serial1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 !
 ip default-gateway 10.3.1.1
 ip classless
 ip route 0.0.0.0 0.0.0.0 10.3.1.1
 no ip http server
 !
 !
 !
 line con 0
 transport input none
 line aux 0
 transport input all
 line vty 0 4
 password password1
 login
 !
 end
```

The following example shows how to display global statistics related to WCCP:

```
Device# show ip wccp web-cache detail

WCCP Client information:
WCCP Client ID:      10.1.1.2
Protocol Version:    2.0
State:               Usable
Redirection:         L2
Packet Return:       L2
Packets Redirected:  0
Connect Time:        00:20:34
Assignment:          MASK
Mask  SrcAddr    DstAddr    SrcPort DstPort
----  -------    -------    ------- -------
0000: 0x00000000 0x00001741 0x0000  0x0000
Value SrcAddr    DstAddr   SrcPort DstPort CE-IP
----- -------    -------   ------- ------- -----
0000: 0x00000000 0x00000000 0x0000 0x0000 0x3C010102 (10.1.1.2)
0001: 0x00000000 0x00000001 0x0000 0x0000 0x3C010102 (10.1.1.2)
0002: 0x00000000 0x00000040 0x0000 0x0000 0x3C010102 (10.1.1.2)
0003: 0x00000000 0x00000041 0x0000 0x0000 0x3C010102 (10.1.1.2)
0004: 0x00000000 0x00000100 0x0000 0x0000 0x3C010102 (10.1.1.2)
0005: 0x00000000 0x00000101 0x0000 0x0000 0x3C010102 (10.1.1.2)
0006: 0x00000000 0x00000140 0x0000 0x0000 0x3C010102 (10.1.1.2)
```

For more information about the **show ip wccp web-cache** command, see the *Cisco IOS IP Application Services Command Reference*.

Example: Enabling WCCP Interoperability with NAT

```
Router(config)# interface ethernet1 ! This is the LAN-facing interface
```

```
Router(config-if)# ip nat inside
Router(config-if)# ip wccp 61 redirect in
Router(config-if)# exit
Router(config)# interface ethernet2 ! This is the WAN-facing interface
Router(config-if)# ip nat outside
Router(config-if)# ip wccp 62 redirect in
Router(config-if)# exit
Router(config)# interface ethernet3 ! This is the WAAS-facing interface
Router(config-if)# ip nat inside
Router(config-if)# ip wccp redirect exclude in
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco ACNS software configuration information	• *Cisco ACNS Software Caching Configuration Guide, Release 4.2* • Cisco ACNS Software listing page on Cisco.com
IP access list overview, configuration tasks, and commands	*Cisco IOS Security Command Reference*
IP addressing and services commands and configuration tasks	• *Cisco IOS IP Addressing Services Configuration Guide* • *Cisco IOS IP Addressing Services Command Reference*
WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for WCCP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 41: Feature Information for WCCP

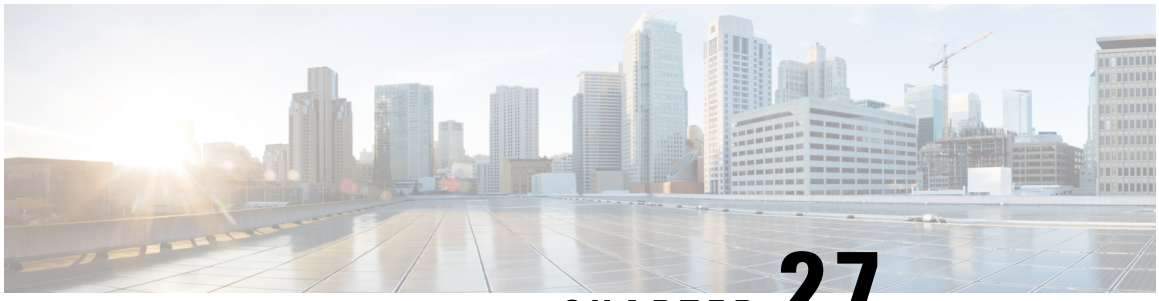
Feature Name	Releases	Feature Information
WCCP Bypass Counters	Cisco IOS XE Release 2.2	The WCCP Bypass Counters feature allows you to display a count of packets that have been bypassed by a web cache and returned to the originating router to be forwarded normally. The **show ip wccp** command was modified by this feature.

Feature Name	Releases	Feature Information
WCCP: Check Services All	Cisco IOS XE Release 3.1S	The WCCP: Check Services All feature enables you to configure WCCP to search all service groups and redirect ACLs in priority order for a match. The following command was modified by this feature: **ip wccp check services all**
WCCP Closed Services	Cisco IOS XE Release 3.1S	The WCCP Closed Services feature permits WCCP services to be configured so that WCCP always intercepts traffic for such services but, if no WCCP client (such as a content engine) has registered to receive this traffic, packets are discarded. This behavior supports Application-Oriented Network Services (AONS) applications, which require traffic to be transparently intercepted using WCCP but do not want the packets to be forwarded to their destination if the WCCP client is unavailable to perform its processing. (This is contrary to the traditional use of WCCP to assist caches where the absence of a cache does not change the behavior as observed by the user.) The following command was modified by this feature: **ip wccp**.
WCCP—Configurable Router ID	Cisco IOS XE Release 3.1S	The WCCP--Configurable Router ID feature permits the router ID which WCCP uses to be configurable, rather than relying on the router's selection mechanism. The following command was modified by this feature:**ip wccp source-interface** .
WCCP Egress Redirection Support	Cisco IOS XE Release 3.1S	The WCCP Egress Redirection Support feature enables WCCP based redirection applied to the outbound traffic on the outbound interface. The following command was modified by this feature: **ip wccp redirect**.
WCCP Exclude Interface	Cisco IOS XE Release 3.1S	The WCCP Exclude Interface feature enables you to configure an interface to exclude packets received on an interface from being checked for redirection by configuring. The following command was introduced by this feature: **ip wccp redirect exclude in**
WCCP Fast Timers	Cisco IOS XE Release 3.1S	The WCCP Fast Timers feature enables WCCP to establish redirection more quickly when a WCCP client is added to a service group or when a WCCP client fails. The following command was modified by this feature: **show ip wccp**.

Feature Name	Releases	Feature Information
WCCP Group List	Cisco IOS XE Release 3.1S	The WCCP Group List feature enables you to configure the IP addresses of cache engines from which a router accepts packets. Configuring a group list is used to validate the protocol packets received from the cache engine. Packets matching the address in a configured group-list are processed, others are discarded. The following command was modified by this feature: **ip wccp**.
WCCP—Group Listen and Multicast Service Support	Cisco IOS XE Release 3.1S	The WCCP--Group Listen and Multicast Service Support feature adds the ability to configure a multicast address per service group for sending and receiving protocol messages. In the multicast address method, the cache engine sends a single-address notification that provides coverage for all routers in the service group. The following command was modified by this feature: **ip wccp group-listen**.
WCCP Increased Services	Cisco IOS XE Release 3.1S	The WCCP Increased Services feature increases the number of services supported by WCCP to a maximum of 256 across all VRFs. The following commands were modified by this feature: **ip wccp**, **ip wccp check services all**, **ip wccp outbound-acl-check**, **show ip wccp**.
WCCP Layer 2 Redirection/Forwarding	Cisco IOS XE Release 2.2	The WCCP Layer 2 Redirection/Forwarding feature allows directly connected Cisco content engines to use Layer 2 redirection, which is more efficient than Layer 3 redirection via GRE encapsulation. You can configure a directly connected Cache Engine to negotiate use of the WCCP Layer 2 Redirection/Forwarding feature. The WCCP Layer 2 Redirection/Forwarding feature requires no configuration on the router or switch. There are no new or modified commands associated with this feature.
WCCP L2 Return	Cisco IOS XE Release 2.2	The WCCP L2 Return feature allows content engines to return packets to WCCP routers directly connected at Layer 2 by swapping the source and destination MAC addresses rather than tunneling packets back to the router inside a Layer 3 GRE tunnel. There are no new or modified commands associated with this feature.
WCCP Mask Assignment	Cisco IOS XE Release 2.2	The WCCP Mask Assignment feature introduces support for ACNS/WAAS devices using mask assignment as a cache engine assignment method. There are no new or modified commands associated with this feature.

Feature Name	Releases	Feature Information
WCCP Outbound ACL Check	Cisco IOS XE Release 3.1S	The WCCP Outbound ACL Check feature enables you to ensure that traffic redirected by WCCP at an input interface is subjected to the outbound ACL checks that may be configured on the output interface prior to redirection. This feature is supported by Web Cache Communication Protocol (WCCP) Version 1 and Version 2. The following commands were introduced or modified by this feature: **ip wccp**, **ip wccp check acl outbound**.
WCCP Redirection on Inbound Interfaces	Cisco IOS XE Release 2.2 Cisco IOS XE Release 3.0S	The WCCP Redirection on Inbound Interfaces feature enables interfaces to be configured for input redirection for a particular WCCP service. When this feature is enabled on an interface, all packets arriving at that interface are compared against the specified WCCP service. If the packets match, they will be redirected. The following commands were introduced or modified by this feature: **ip wccp redirect-list**.
WCCP Version 2	Cisco IOS XE Release 2.2	The WCCP Version 2 feature provides several enhancements and features to the WCCP protocol, including: • The ability of multiple routers to service a content engine cluster. • Redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. • Optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the HMAC MD5 standard. • A check on packets that determines which requests have been returned from the content engine unserviced. • Load adjustments for individual content engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. The following commands were introduced or modified by this feature: **clear ip wccp, ip wccp, ip wccp group-listen, ip wccp redirect, ip wccp redirect exclude in, ip wccp version, show ip wccp**.

Feature Name	Releases	Feature Information
WCCP VRF Support	Cisco IOS XE Release 3.1S	The WCCP VRF Support feature provides enhancements to the existing WCCPv2 protocol which support VRF awareness. In Cisco IOS Release 12.2(33)SRE, this feature is supported only on Cisco 7200 NPE-G2 and Cisco 7304-NPE-G100 routers. The following commands were introduced or modified by this feature: **clear ip wccp**, **debug ip wccp**, **ip wccp**, **ip wccp group-listen**, **ip wccp redirect**, **show ip wccp**.



CHAPTER 27

WCCP—Configurable Router ID

The WCCP—Configurable Router ID feature enables the configuration of a Web Cache Communication Protocol (WCCP) source interface. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, router IDs are no longer automatically generated when the current router ID is no longer valid and the router ID does not change when another IP address is added to the system.

Restrictions for WCCP—Configurable Router ID

The following restriction apply to this feature:

- Do not configure the Web Cache Control Protocol (WCCP) router ID as the tunnel source, if multipoint generic routing encapsulation (GRE) tunnels are configured on a router, because this configuration may cause the traffic over this tunnel to fail.

Information About WCCP—Configurable Router ID

WCCP—Configurable Router ID Overview

WCCP uses a router ID in its control messages that a WCCP client can use to uniquely identify a particular WCCP server. The router ID is an IP address and is used as the source address of any WCCP-generated Generic Routing Encapsulation (GRE) frames. Prior to the WCCP—Configurable Router ID feature, WCCP selected a router ID using an automatic mechanism; the highest reachable IP address on the system (or the highest loopback IP address, if there is one) was used as the WCCP router ID. The highest IP address on the system is not always the best choice as the router ID or as the source address of GRE frames. A change in addressing information on the system may cause the WCCP router ID to change unexpectedly. During this changeover period, WCCP clients briefly advertise the existence of two routers (the old router ID and the new router ID) and GRE frames are sourced from a different address.

The WCCP—Configurable Router ID feature enables you to define a WCCP source interface from which the router ID will be obtained. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, the router ID does not change when another IP address is added to the system. The router ID changes only when a new router ID is manually configured using the **ip wccp source- interface** or the **ipv6 wccp source- interface** command, or when the address on the manually configured interface is no longer valid.

How to Configure WCCP—Configurable Router ID

Configuring a Preferred WCCP Router ID

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp** [**vrf** *vrf-name*] **source-interface** *source-interface*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip wccp** [**vrf** *vrf-name*] **source-interface** *source-interface* **Example:** `Device(config)# ip wccp source-interface GigabitEthernet 0/0/0`	Configures a preferred WCCP router ID.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for WCCP—Configurable Router ID

Example: Configuring a Preferred WCCP Router ID

The following example displays the configuration for a preferred WCCP router ID:

```
! Configure a preferred WCCP router ID
ip wccp source-interface GigabitEthernet 0/0/0
```

Additional References for WCCP—Configurable Router ID

Related Documents

Related Topic	Document Title
WCCP commands	Cisco IOS IP Application Services Command Reference
Currently assigned IP multicast addresses	*Internet Multicast Addresses* http://www.iana.org/assignments/multicast-addresses
Configuration fundamentals configuration tasks	*Cisco IOS Configuration Fundamentals Configuration Guide*
Configuration fundamentals commands	*Cisco IOS Configuration Fundamentals Command Reference*
Cisco IOS bridging and IBM networking configuration tasks	*Cisco IOS Bridging and IBM Networking Configuration Guide*
Cisco IOS bridging and IBM networking commands	*Cisco IOS Bridging and IBM Networking Command Reference*
Cisco IOS IP multicast configuration tasks	*Cisco IOS IP Multicast Configuration Guide*
Cisco IOS IP Multicast commands	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
IEEE Spanning-Tree Bridging	802.1D MAC Bridges http://www.ieee802.org/1/pages/802.1D-2003.html

MIBs

MIB	MIBs Link
—	No new or modified MIBs are supported, and support for existing MIBs has not been modified.

RFCs

RFC	Title
RFC 1812	*Requirements for IP Version 4 Routers* http://www.ietf.org/rfc/rfc1812.txt
RFC 2131	*Dynamic Host Configuration Protocol* http://www.ietf.org/rfc/rfc2131.txt .

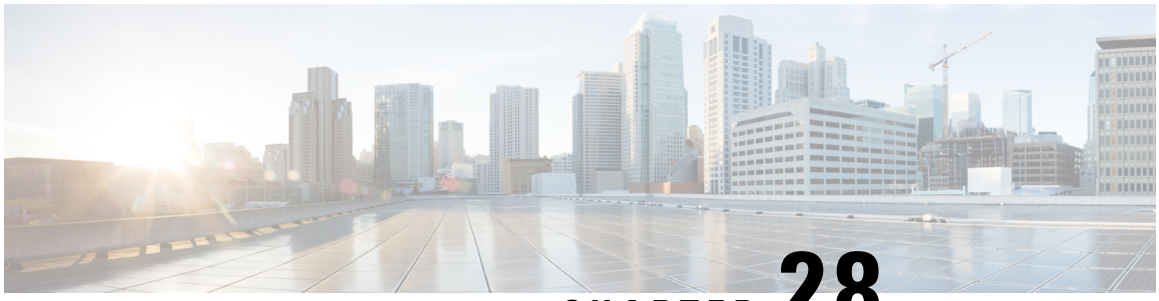
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for WCCP—Configurable Router ID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 28

WCCPv2—IPv6 Support

This feature introduces support for Web Cache Communication Protocol version 2 (WCCPv2) in an IPv6 environment.

WCCP is a Cisco-developed content-routing technology that intercepts IP packets and redirects those packets to a destination other than that specified in the IP packet. Typically the packets are redirected from their destination web server on the Internet to a content engine that is local to the client. In some WCCP deployment scenarios, redirection of traffic may also be required from the web server to the client. WCCP enables you to integrate content engines into your network infrastructure.

Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router can redirect content requests to a cluster.

Prerequisites for WCCPv2—IPv6 Support

- IPv6 must be configured on the interface used for redirection and on the interface facing the content engine.
- The interface connected to the content engine must be a Fast Ethernet or Gigabit Ethernet interface.

Restrictions for WCCPv2—IPv6 Support

WCCPv2

- For routers servicing a multicast cluster, the Time To Live (TTL) value must be set at 15 or lower.
- Service groups can comprise up to 32 content engines and 32 routers.
- All content engines in a cluster must be configured to communicate with all routers servicing the cluster.

- Multicast addresses must be in the range from 224.0.0.0 to 239.255.255.255.

Layer 2 Forwarding and Return

- Layer 2 redirection requires that content engines be directly connected to an interface on each WCCP router. Unless multicast IP addresses are used, WCCP configuration of the content engine must reference the directly connected interface IP address of the WCCP router and not a loopback IP address or any other IP address configured on the WCCP router.

Information About WCCPv2—IPv6 Support

WCCP Overview

WCCP uses Cisco Content Engines (or other content engines running WCCP) to localize traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables Cisco IOS XE routing platforms to transparently redirect content requests. With transparent redirection, users can fulfill content requests locally without configuring their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a content engine. The word "transparent" in this case means that the end user does not know that a requested file (such as a web page) came from the content engine instead of from the originally specified server.

A content engine receiving a request attempts to service it from its own local cache. If the requested information is not present, the content engine issues its own request to the originally targeted server to get the required information. A content engine retrieving the requested information forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of content engines, called a content engine cluster, to provide content to a router or multiple routers. Network administrators can easily scale their content engines to manage heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each cluster member to work in parallel, resulting in linear scalability. Clustering content engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 content engines to scale to your desired capacity.

Layer 2 Forwarding Redirection and Return

WCCP uses either generic routing encapsulation (GRE) or Layer 2 (L2) to redirect or return IP traffic. When WCCP forwards traffic via GRE, the redirected packets are encapsulated within a GRE header. The packets also have a WCCP redirect header. When WCCP forwards traffic using L2, the original MAC header of the IP packet is overwritten and replaced with the MAC header for the WCCP client.

Using L2 as a forwarding method allows direct forwarding to the content engine without further lookup. Layer 2 redirection requires that the router and content engines are directly connected, that is, on the same IP subnetwork.

When WCCP returns traffic via GRE, the returned packets are encapsulated within a GRE header. The destination IP address is the address of the router and the source address is the address of the WCCP client.

When WCCP returns traffic via L2, the original IP packet is returned without any added header information. The router to which the packet is returned will recognize the source of the packet and prevent redirection.

The WCCP redirection method does not have to match the return method.

L2 forwarding, return, or redirection are typically used for hardware-accelerated platforms. Depending on your release, L2 forwarding, return, and redirection can also be used for software-switching platforms.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **l2-redirect** keyword to configure L2 redirection. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **l2-redirect** keyword to configure L2 redirection.

Note Before configuring a GRE tunnel, configure a loopback interface (that is not attached to a VRF) with an IP address so that the internally created tunnel interface is enabled for IPv4 forwarding by unnumbering itself to this dummy loopback interface. You do not need to configure a loopback interface if the system has at least one interface that is not attached to a VRF and that is configured with an IPv4 address.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

For more information about WAAS commands used to configure Cisco Content Engines, see the Cisco Wide Area Application Services Command Reference.

WCCP Mask Assignment

The WCCP Mask Assignment feature enables mask assignment as the load-balancing method (instead of the default hash assignment method) for a WCCP service.

For content engines running Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **mask-assign** keyword to configure mask assignment. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **mask-assign** keyword to configure mask assignment.

For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

For more information about WAAS commands used to configure Cisco Content Engines, see the Cisco Wide Area Application Services Command Reference.

WCCP Hash Assignment

The Cisco ASR 1000 Series Aggregation Services Routers support hash assignment for IPv6 load balance across different content engines, but does not support mask assignment. However, it supports both hash assignment and mask assignment for IPv4.

For content engines running the Cisco Application and Content Networking System (ACNS) software, use the **wccp custom-web-cache** command with the **hash-assign** keyword to configure hash assignment. For content engines running Cisco Wide Area Application Services (WAAS) software, use the **wccp tcp-promiscuous** command with the **hash-assign** keyword to configure hash assignment.

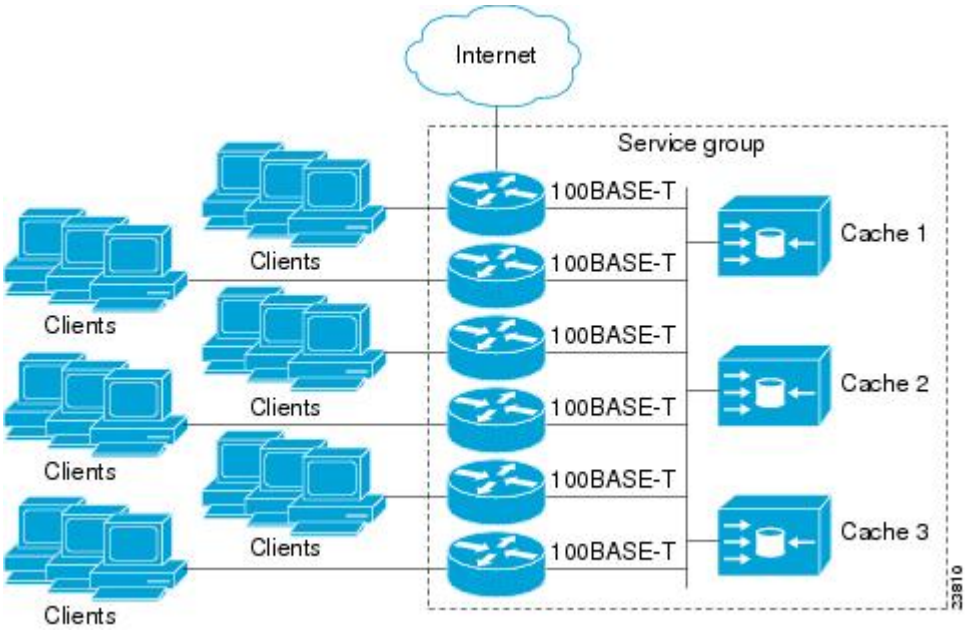
For information about Cisco ACNS commands used to configure Cisco Content Engines, see the Cisco ACNS Software Command Reference.

For more information about WAAS commands used to configure Cisco Content Engines, see the Cisco Wide Area Application Services Command Reference.

WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router could redirect content requests to a cluster. The figure below illustrates a sample configuration using multiple routers.

Figure 33: Cisco Content Engine Network Configuration Using WCCPv2



The subset of content engines within a cluster and routers connected to the cluster that are running the same service is known as a service group. Available services include TCP and UDP redirection.

In WCCPv1, the content engines were configured with the address of the single router. WCCPv2 requires that each content engine be aware of all the routers in the service group. To specify the addresses of all the routers in a service group, you must choose one of the following methods:

- Unicast—A list of router addresses for each of the routers in the group is configured on each content engine. In this case the address of each router in the group must be explicitly specified for each content engine during configuration.
- Multicast—A single multicast address is configured on each content engine. In the multicast address method, the content engine sends a single-address notification that provides coverage for all routers in the service group. For example, a content engine could indicate that packets should be sent to a multicast address of 224.0.0.100, which would send a multicast packet to all routers in the service group configured for group listening using WCCP (see the **ip wccp group-listen** or the **ipv6 wccp group-listen** interface configuration command for details).

The multicast option is easier to configure because you need only specify a single address on each content engine. This option also allows you to add and remove routers from a service group dynamically, without needing to reconfigure the content engines with a different list of addresses each time.

The following sequence of events details how WCCPv2 configuration works:

1. Each content engine is configured with a list of routers.
2. Each content engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of content engines in the group.
3. When the view is consistent across all content engines in the cluster, one content engine is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

WCCPv2 Support for Services Other Than HTTP

WCCPv2 allows redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. WCCPv2 supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and Real Audio, video, and telephony applications.

To accommodate the various types of services available, WCCPv2 introduced the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as 98) or a predefined service keyword (such as **web-cache**). This information is used to validate that service group members are all using or providing the same service.

The content engines in a service group specify traffic to be redirected by protocol (TCP or UDP) and up to eight source or destination ports. Each service group has a priority status assigned to it. The priority of a dynamic service is assigned by the content engine. The priority value is in the range of 0 to 255 where 0 is the lowest priority. The predefined web-cache service has an assigned priority of 240.

WCCPv2 Support for Multiple Routers

WCCPv2 allows multiple routers to be attached to a cluster of cache engines. The use of multiple routers in a service group allows for redundancy, interface aggregation, and distribution of the redirection load. WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the Hashed Message Authentication Code—Message Digest (HMAC MD5) standard. Shared-secret MD5 one-time authentication (set using the **ip wccp** [**password** [**0** | **7**] *password*] global configuration command) enables messages to be protected against interception, inspection, and replay.

WCCPv2 Web Cache Packet Return

If a content engine is unable to provide a requested object it has cached due to error or overload, the content engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the content engine unserviced. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the content engine cluster). This process provides error handling transparency to clients.

Typical reasons why a content engine would reject packets and initiate the packet return feature include the following:

- Instances when the content engine is overloaded and has no room to service the packets
- Instances when the content engine is filtering for certain conditions that make caching packets counterproductive (for example, when IP authentication has been turned on)

WCCPv2 Load Distribution

WCCPv2 can be used to adjust the load being offered to individual content engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. WCCPv2 allows the designated content engine to adjust the load on a particular content engine and balance the load across the content engines in a cluster. WCCPv2 uses three techniques to perform load distribution:

- Hot spot handling—Allows an individual hash bucket to be distributed across all the content engines. Prior to WCCPv2, information from one hash bucket could go to only one content engine.
- Load balancing—Allows the set of hash buckets assigned to a content engine to be adjusted so that the load can be shifted from an overwhelmed content engine to other members that have available capacity.
- Load shedding—Enables the router to selectively redirect the load to avoid exceeding the capacity of a content engine.

The use of these hashing parameters prevents one content engine from being overloaded and reduces the potential for bottlenecking.

WCCP VRF Support

The WCCP VRF Support feature enhances the WCCPv2 protocol by implementing support for virtual routing and forwarding (VRF).

The WCCP VRF Support feature allows service groups to be configured on a per-VRF basis in addition to those defined globally.

Along with the service identifier, the VRF of WCCP protocol packets arriving at the router is used to associate cache-engines with a configured service group.

The same VRF must have the interface on which redirection is applied, the interface which is connected to cache engine, and the interface on which the packet would have left if it had not been redirected.

IPv6 WCCP Tunnel Interface

The use of GRE redirection results in the creation of new tunnel interfaces. You can display these tunnel interfaces by entering the **show ipv6 interface brief | include tunnel** command:

```
Device# show ipv6 interface brief | include tunnel

Tunnel0                2001::DB8:1::1      YES unset  up                    up
Tunnel1                2001::DB8:1::1      YES unset  up                    up
Tunnel2                2001::DB8:1::1      YES unset  up                    up
Tunnel3                2001::DB8:1::1      YES unset  up                    up
Device#
```

The tunnel interfaces are automatically created in order to process outgoing GRE-encapsulated traffic for WCCP. The tunnel interfaces appear when a content engine connects and requests GRE redirection. The tunnel interfaces are not created directly by WCCP, but are created indirectly via a tunnel application

programming interface (API). WCCP does not have direct knowledge of the tunnel interfaces, but can redirect packets to them, resulting in the appropriate encapsulation being applied to the packets. After the appropriate encapsulation is applied, the packet is then sent to the content engine.

Note The tunnel interfaces are not used to connect with incoming WCCP GRE return packets.

One tunnel is created for each service group that is using GRE redirection. One additional tunnel is created to provide an IP address that allows the other tunnel group interfaces to be unnumbered but still enabled for IPv6.

You can confirm the connection between the tunnels and WCCP by entering the **show tunnel groups wccp** command:

```
Device# show tunnel groups wccp

WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
   intf: Tunnel0, locally sourced
 WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
   intf: Tunnel3, locally sourced
 WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
   intf: Tunnel2, locally sourced
```

You can display additional information about each tunnel interface by entering the **show tunnel interface** *interface-number* command:

```
Device# show tunnel interface t0

Tunnel0
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::2
   Application ID 2: WCCP : service group 0 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up

Device# show tunnel interface t1

Tunnel1
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
   Application ID 2: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up

Device# show tunnel interface t2

Tunnel2
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
   Application ID 2: WCCP : service group 318 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up

Device# show tunnel interface t3

Tunnel3
   Mode:multi-GRE/IP, Destination UNKNOWN, Source 2001::DB8:1::1
   Application ID 2: WCCP : service group 317 in "Default", ver v2, assgnmnt: hash-table
   Linestate - current up
   Internal linestate - current up, evaluated up
Device#
```

Note that the service group number shown in the examples is the internal tunnel representation of the WCCP service group number. Group 0 is the web-cache service. To determine the dynamic services, subtract 256 from the displayed service group number to convert to the WCCP service group number. For interfaces that are used for redirection, the source address shown is the WCCP router ID.

You can display information about the connected content engines and encapsulation, including software packet counters, by entering the **show adjacency** [*tunnel-interface*] [**encapsulation**] [**detail**] [**internal**] command:

```
Device# show adjacency t0

Protocol Interface                 Address
IP       Tunnel0                   2001::DB8:1::1(3)

Device# show adjacency t0 encapsulation

Protocol Interface                 Address
IPV6     Tunnel1                   2001:DB8:1::11(2)
  Encap length 48
  6000000000002FFF20010DB801000000
  00000000000000120010DB800010000
  00000000000000110000883E00000000
  Provider: TUNNEL
IPV6     Tunnel1                   2001:DB8:1::12(2)
  Encap length 48
  6000000000002FFF20010DB801000000
  00000000000000120010DB800010000
  00000000000000120000883E00000000
  Provider: TUNNEL

Device# show adjacency t0 detail

Protocol Interface                 Address
IPV6     Tunnel1                   2001:DB8:1::11(2)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 22
                                   Encap length 48
                                   6000000000002FFF20010DB801000000
                                   00000000000000120010DB800010000
                                   00000000000000110000883E00000000
                                   Tun endpt
                                   Next chain element:
                                    punt

Device# show adjacency t0 internal

Protocol Interface                 Address
IPV6     Tunnel1                   2001:DB8:1::11(2)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 22
                                   Encap length 48
                                   6000000000002FFF20010DB801000000
                                   00000000000000120010DB800010000
                                   00000000000000110000883E00000000
                                   Tun endpt
                                   Next chain element:
                                    punt
                                    parent oce 0x68C55B00
                                    frame originated locally (Null0)
                                   L3 mtu 0
                                   Flags (0x2808C6)
```

```
                                        Fixup disabled
                                        HWIDB/IDB pointers 0x200900DC/0x20090D98
                                        IP redirect disabled
                                        Switching vector: IPv6 midchain adjacency oce
                                        Next-hop cannot be inferred
                                        IP Tunnel stack to 2001:DB8:1::11 in Default (0x0)

Device#
```

WCCP Bypass Packets

WCCP intercepts IP packets and redirects those packets to a destination other than the destination that is specified in the IP header. Typically the packets are redirected from a web server on the Internet to a web cache that is local to the destination.

Occasionally a web cache cannot manage the redirected packets appropriately and returns the packets unchanged to the originating router. These packets are called bypass packets and are returned to the originating router using either Layer 2 forwarding without encapsulation (L2) or encapsulated in generic routing encapsulation (GRE). The router decapsulates and forwards the packets normally. The VRF associated with the ingress interface (or the global table if there is no VRF associated) is used to route the packet to the destination.

GRE is a tunneling protocol developed by Cisco that encapsulates packet types from a variety of protocols inside IP tunnels, creating a virtual point-to-point link over an IP network.

WCCP Closed Services and Open Services

In applications where packets are intercepted and redirected by a Cisco IOS router to external WCCP client devices, it may be necessary to block the packets for the application when a WCCP client device is not available. This blocking is achieved by configuring a WCCP closed service. When a WCCP service is configured as closed, the packets that fulfill the services, but do not have an active client device, are discarded.

By default, WCCP operates as an open service, wherein communication between clients and servers proceeds normally in the absence of an intermediary device.

The **ip wccp service-list** or the **ipv6 wccp service-list** command can be used for both closed-mode and open-mode services. Use the **service-list** keyword and *service-access-list* argument to register an application protocol type or port number. Use the **mode** keyword to select an open or closed service.

WCCP Outbound ACL Check

When WCCP is enabled for redirection on an ingress interface, the packets are redirected by WCCP and instead egress on an interface other than the destination that is specified in the IP header. The packets are still subject to ACLs configured on the ingress interface. However, redirection can cause the packets to bypass the ACL configured on the original egress interface. Packets that would have been dropped because of the ACL configured on the original egress interface can be sent out on the redirect egress interface, which poses a possible security problem. Enabling the WCCP Outbound ACL check feature ensures that redirected packets are subject to any ACL conditions configured on the original egress interface.

WCCP Service Groups

WCCP is a component of Cisco IOS software that redirects traffic with defined characteristics from its original destination to an alternative destination. The typical application of WCCP is to redirect traffic bound for a remote web server to a local web cache to improve response time and optimize network resource usage.

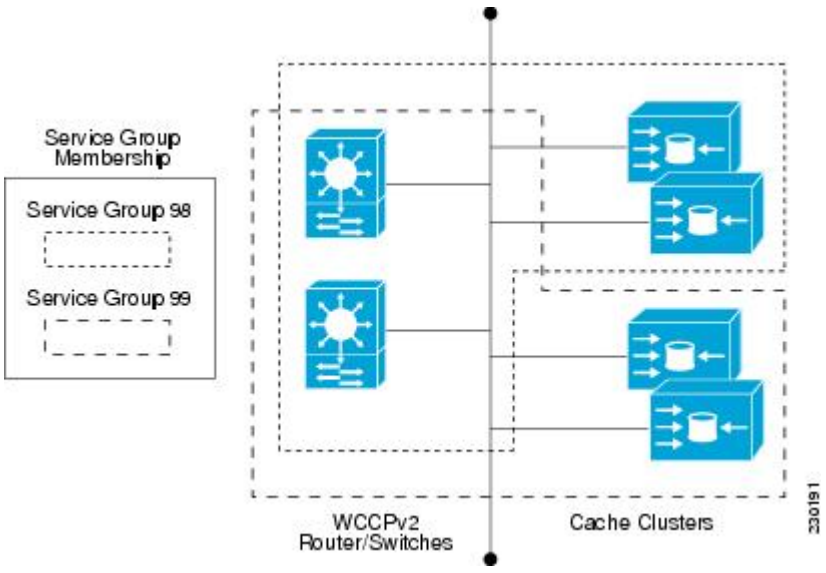
The nature of the selected traffic for redirection is defined by service groups (see figure below) specified on content engines and communicated to routers by using WCCP. The maximum number of service groups allowed across all VRFs is 256.

WCCPv2 supports up to 32 routers per service group. Each service group is established and maintained independently.

WCCPv2 uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is web cache, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the content engines. This service is referred to as a *well-known service*, because the characteristics of the web cache service are known by both the router and content engines. A description of a well-known service is not required beyond a service identification. To specify the standard web cache service, use the **ip wccp** or the **ipv6 wccp** command with the **web-cache** keyword.

Note More than one service can run on a router at the same time, and routers and content engines can be part of multiple service groups at the same time.

Figure 34: WCCP Service Groups



The dynamic services are defined by the content engines; the content engine instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group's traffic, because this information is provided by the first content engine to join the group. In a dynamic service, up to eight ports can be specified within a single protocol.

Cisco Content Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other content engine devices may use this service number for some other service.

WCCP—Check All Services

An interface may be configured with more than one WCCP service. When more than one WCCP service is configured on an interface, the precedence of a service depends on the relative priority of the service compared to the priority of the other configured services. Each WCCP service has a priority value as part of its definition. When an interface is configured with more than one WCCP service, the precedence of the packets is matched against service groups in priority order.

Note The priority of a WCCP service group cannot be configured via Cisco IOS software.

With the **ip wccp check services all** or the **ipv6 wccp check services all** command, WCCP can be configured to check all configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by a redirect ACL and by the service priority. The **ip wccp check services all** commands must be configured at global level to support multiple WCCP services.

If no WCCP services are configured with a redirect ACL, the services are considered in priority order until a service is found that matches the IP packet. If no services match the packet, the packet is not redirected. If a service matches the packet and the service has a redirect ACL configured, then the IP packet will be checked against the ACL. If the packet is rejected by the ACL, the packet will not be passed down to lower priority services unless the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured. When the **ip wccp check services all** or the **ipv6 wccp check services all** command is configured, WCCP will continue to attempt to match the packet against any remaining lower priority services configured on the interface.

WCCP—Configurable Router ID Overview

WCCP uses a router ID in its control messages that a WCCP client can use to uniquely identify a particular WCCP server. The router ID is an IP address and is used as the source address of any WCCP-generated Generic Routing Encapsulation (GRE) frames. Prior to the WCCP—Configurable Router ID feature, WCCP selected a router ID using an automatic mechanism; the highest reachable IP address on the system (or the highest loopback IP address, if there is one) was used as the WCCP router ID. The highest IP address on the system is not always the best choice as the router ID or as the source address of GRE frames. A change in addressing information on the system may cause the WCCP router ID to change unexpectedly. During this changeover period, WCCP clients briefly advertise the existence of two routers (the old router ID and the new router ID) and GRE frames are sourced from a different address.

The WCCP—Configurable Router ID feature enables you to define a WCCP source interface from which the router ID will be obtained. The IP address of this configured source interface is then used as the preferred WCCP router ID and WCCP GRE source address. When a WCCP router ID is manually configured, the router ID does not change when another IP address is added to the system. The router ID changes only when a new router ID is manually configured using the **ip wccp source- interface** or the **ipv6 wccp source- interface** command, or when the address on the manually configured interface is no longer valid.

WCCP Troubleshooting Tips

CPU usage may be very high when WCCP is enabled. The WCCP counters enable a determination of the bypass traffic directly on the router and can indicate whether the cause is high CPU usage due to enablement of WCCP. In some situations, 10 percent bypass traffic may be normal; in other situations, 10 percent may

be high. However, any figure above 25 percent should prompt a closer investigation of what is occurring in the web cache.

If the counters suggest that the level of bypass traffic is high, the next step is to examine the bypass counters in the content engine and determine why the content engine is choosing to bypass the traffic. You can log in to the content engine console and use the CLI to investigate further. The counters allow you to determine the percent of traffic being bypassed.

You can use the **clear ipv6 wccp***service-id* command to remove the IPv6 WCCP statistics (counts) maintained on the router for a particular service.

You can use the **clear wccp** command to remove all (IPv4 and IPv6) WCCP statistics (counts) maintained on the router for a particular service.

You can use the **show ipv6 wccp** command to display the IPv6 WCCP global statistics (counts).

You can use the **show wccp** command to display all (IPv4 and IPv6) WCCP global statistics (counts).

How to Configure WCCPv2—IPv6 Support

Configuring a General WCCPv2—IPv6 Session

Perform this task to configure a general IPv6 WCCPv2 session.

Until you configure a WCCP service using the **ipv6 wccp** {**web-cache** | *service-number*} global configuration command, WCCP is disabled on the router. The first use of a form of the **ipv6 wccp** command enables WCCP. By default WCCPv2 is used for services.

Using the **ipv6 wccp web-cache password** command, you can set a password for a router and the content engines in a service group. MD5 password security requires that each router and content engine that wants to join a service group be configured with the service group password. The password can be up to eight characters in length. Each content engine or router in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 wccp** [**vrf** *vrf-name*] **source-interface** *source-interface*
4. **ipv6 wccp** [**vrf** *vrf-name*] { **web-cache** | *service-number*} [**group-address** *group-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password* [**0** | **7**]]
5. **interface** *type number*
6. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **redirect** {**out** | **in**}
7. **exit**
8. **interface** *type number*
9. **ipv6 wccp redirect exclude in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 wccp** [**vrf** *vrf-name*] **source-interface** *source-interface* **Example:** `Device(config)# ipv6 wccp source-interface GigabitEthernet 0/0/0`	Configures a preferred WCCP router ID.
Step 4	**ipv6 wccp** [**vrf** *vrf-name*] { **web-cache** | *service-number*} [**group-address** *group-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password* [**0** | **7**]] **Example:** `Device(config)# ipv6 wccp web-cache password password1`	Specifies a web-cache or dynamic service to enable on the router, specifies a VRF-name to associate with the service group, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Targets an interface number for which the web cache service will run, and enters interface configuration mode.
Step 6	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **redirect** {**out** | **in**} **Example:** `Device(config-if)# ipv6 wccp web-cache redirect in`	Enables packet redirection on an outbound or inbound interface using WCCP. • As indicated by the **out** and **in** keyword options, redirection can be specified for outbound interfaces or inbound interfaces.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/2/0`	Targets an interface number on which to exclude traffic for redirection, and enters interface configuration mode.

	Command or Action	Purpose
Step 9	**ipv6 wccp redirect exclude in** **Example:** î `Device(config-if)# ipv6 wccp redirect exclude in`	(Optional) Excludes traffic on the specified interface from redirection.

Configuring Services for WCCPv2—IPv6

Perform this task to specify the number of service groups for WCCP, to configure a service group as a closed or open service, and to optionally specify a check of all services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ipv6 wccp** [**vrf** *vrf-name*] *service-number* [**service-list** *service-access-list* **mode** {**open** | **closed**}]
   - **ipv6 wccp** [**vrf** *vrf-name*] **web-cache mode** {**open** | **closed**}
4. **ipv6 wccp check services all**
5. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ipv6 wccp** [**vrf** *vrf-name*] *service-number* [**service-list** *service-access-list* **mode** {**open**	**closed**}] • **ipv6 wccp** [**vrf** *vrf-name*] **web-cache mode** {**open**	**closed**} **Example:** `Device(config)# ipv6 wccp 90 service-list 120 mode closed`	Configures a dynamic WCCP service as closed or open. or Configures a web-cache service as closed or open. **Note** When configuring the web-cache service as a closed service, you cannot specify a service access list.

	Command or Action	Purpose
	or `Device(config)# ipv6 wccp web-cache mode closed`	**Note** When configuring a dynamic WCCP service as a closed service, you must specify a service access list.
Step 4	**ipv6 wccp check services all** **Example:** `Device(config)# ipv6 wccp check services all`	(Optional) Enables a check of all WCCP services. • Use this command to configure WCCP to check the other configured services for a match and perform redirection for those services if appropriate. The caches to which packets are redirected can be controlled by the redirect ACL and not just the service description. **Note** The **ipv6 wccp check services all** command is a global WCCP command that applies to all services and is not associated with a single service.
Step 5	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **Example:** `Device(config)# ipv6 wccp 201`	Specifies the WCCP service identifier. • You can specify the standard web-cache service or a dynamic service number from 0 to 255. • The maximum number of services that can be specified is 256.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.

Registering a Router to a Multicast Address for WCCPv2— IPv6

If you decide to use the multicast address option for your service group, you must configure the router to listen for the multicast broadcasts on an interface.

For network configurations where redirected traffic needs to traverse an intervening router, the router being traversed must be configured to perform IP multicast routing. You must configure the following two components to enable traversal over an intervening router:

- Enable IP multicast routing using the **ipv6 multicast-routing** global configuration command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ipv6 wccp group-listen** interface configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing** [**vrf** *vrf-name*] [**distributed**]
4. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-address** *multicast-address*

5. **interface** *type number*
6. **ip pim** {**sparse-mode** | **sparse-dense-mode** | **dense-mode** [**proxy-register** {**list** *access-list* | **route-map** *map-name*}]}
7. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-listen**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast-routing** [**vrf** *vrf-name*] [**distributed**] **Example:** `Device(config)# ipv6 multicast-routing`	Enables IP multicast routing.
Step 4	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-address** *multicast-address* **Example:** `Device(config)# ipv6 wccp 99 group-address FF15::8000:1`	Specifies the multicast address for the service group.
Step 5	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Enables the interfaces to which the content engines will connect to receive multicast transmissions for which the web cache service will run, and enters interface configuration mode.
Step 6	**ip pim** {**sparse-mode** | **sparse-dense-mode** | **dense-mode** [**proxy-register** {**list** *access-list* | **route-map** *map-name*}]} **Example:** `Device(config-if)# ip pim dense-mode`	(Optional) Enables Protocol Independent Multicast (PIM) on an interface. **Note** To ensure correct operation of the **ipv6 wccp group-listen** command, you must enter the **ip pim** command in addition to the **ipv6 wccp group-listen** command.
Step 7	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} **group-listen** **Example:** `Device(config-if)# ipv6 wccp 99 group-listen`	Configures an interface to enable or disable the reception of IP multicast packets for WCCP.

Using Access Lists for WCCPv2—IPv6 Service Group

Perform this task to configure the router to use an access list to determine which traffic should be directed to which content engines.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**]
5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]
7. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
8. **ipv6 wccp** [**vrf** *vrf-name*] **web-cache group-list** *access-list*
9. **ipv6 wccp** [**vrf** *vrf-name*] **web-cache redirect-list** *access-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 1 remark Give access to user1`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters in length can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** {*source* [*source-wildcard*]	**any**} [**log**] **Example:** `Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0`	Creates an access list that enables or disables traffic redirection to the cache engine and permits the specified source based on a source address and wildcard mask. • Every access list needs at least one permit statement; it does not need to be the first entry. • Standard IP access lists are numbered 1 to 99 or 1300 to 1999. • If the *source-wildcard* string is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.

	Command or Action	Purpose
		• Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.5.22 is allowed to pass the access list.
Step 5	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 1 remark Give access to user1`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 6	**access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0`	Denies the specified source based on a source address and wildcard mask. • If the *source-wildcard* string is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the abbreviation any as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.7.34 is denied passing the access list.
Step 7	Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**ipv6 wccp** [**vrf** *vrf-name*] **web-cache group-list** *access-list* **Example:** `Device(config) ipv6 wccp web-cache group-list 1`	Indicates to the router from which IP addresses of content engines to accept packets.
Step 9	**ipv6 wccp** [**vrf** *vrf-name*] **web-cache redirect-list** *access-list* **Example:** `Router(config)# ipv6 wccp web-cache redirect-list 1`	(Optional) Disables caching for certain clients.

Enabling the WCCP—IPv6 Outbound ACL Check

Note When all redirection is performed in the hardware, the mode of redirection will change when outbound ACL checking is enabled. The first packet is switched in software to allow the extra ACL check to be performed before a shortcut is installed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 wccp** [**vrf** *vrf-name*] {**web-cache** | *service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*]
4. **ipv6 wccp check acl outbound**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 wccp** [**vrf** *vrf-name*] {**web-cache**	*service-number*} [**group-address** *multicast-address*] [**redirect-list** *access-list*] [**group-list** *access-list*] [**password** *password*] **Example:** `Device(config)# ipv6 wccp web-cache`	Enables support for a Cisco content engine service group or any content engine service group and configures a redirect ACL list or group ACL.
Step 4	**ipv6 wccp check acl outbound** **Example:** `Device(config)# ipv6 wccp check acl outbound`	Checks the access control list (ACL) for egress interfaces for packets redirected by WCCP.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration.

Verifying and Monitoring WCCPv2—IPv6 Configuration Settings

SUMMARY STEPS

1. **enable**
2. **show ipv6 wccp** [**vrf** *vrf-name*] [*service-number* | **web-cache**] [**detail** | **view**]
3. **show ipv6 interface**
4. **more system:running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 wccp** [**vrf** *vrf-name*] [*service-number*	**web-cache**] [**detail**	**view**] **Example:** `Device# show ipv6 wccp 24 detail`	(Optional) Displays global information related to WCCP, including the protocol version currently running, the number of content engines in the router service group, which content engine group is allowed to connect to the router, and which access list is being used. The argument and keywords are as follows: • *service-number*—(Optional) Dynamic number of the web-cache service group being controlled by the content engine. The range is from 0 to 99. For web caches that use Cisco Content Engines, the reverse proxy service is indicated by a value of 99. • **web-cache**—(Optional) Statistics for the web-cache service. • **detail**—(Optional) Other members of a particular service group or web cache that have or have not been detected. • **view**—(Optional) Information about a router or all web caches.
Step 3	**show ipv6 interface** **Example:** `Device# show ipv6 interface`	(Optional) Displays status about whether any **ip wccp redirection** commands are configured on an interface; for example, “Web Cache Redirect is enabled / disabled.”
Step 4	**more system:running-config** **Example:** `Device# more system:running-config`	(Optional) Displays contents of the currently running configuration file (equivalent to the **show running-config** command).

Configuration Examples for WCCPv2—IPv6 Support

Example: Configuring a General WCCPv2—IPv6 Session

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache password password1
Device(config)# ipv6 wccp source-interface GigabitEthernet 0/1/0
Device(config)# ipv6 wccp check services all
 Configures a check of all WCCP services.
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/2/0
Device(config-if)# ipv6 wccp redirect exclude in
Device(config-if)# exit
```

Example: WCCPv2—IPv6—Setting a Password for a Router and Content Engines

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache password password1
```

Example: WCCPv2—IPv6—Configuring a Web Cache Service

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device# copy running-config startup-config
```

The following example shows how to configure a session in which redirection of HTTP traffic arriving on Gigabit Ethernet interface 0/1/0 is enabled:

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
Device(config-if)# exit
Device# show ip interface GigabitEthernet 0/1/0
.
.
.
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
.
.
.
```

Example: WCCPv2—IPv6—Running a Reverse Proxy Service

The following example assumes that you are configuring a service group using Cisco cache engines, which use dynamic service 99 to run a reverse proxy service:

```
Device# configure terminal
Device(config)# ipv6 wccp 99
Device(config)# interface GigabitEthernet 0/1/0
Device(config-if)# ipv6 wccp 99 redirect out
```

Example: WCCPv2—IPv6—Registering a Router to a Multicast Address

```
Device# configure terminal
Device(config)# ipv6 wccp web-cache group-address 224.1.1.100
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp web cache group-listen
```

The following example shows a device configured to run a reverse proxy service, using the multicast address of 224.1.1.1. Redirection applies to packets outgoing via Gigabit Ethernet interface 0/1/0:

```
Device# configure terminal
Device(config)# ipv6 wccp 99 group-address 224.1.1.1
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp 99 redirect out
```

Example: WCCPv2—IPv6—Using Access Lists for a WCCPv2 IPv6 Service Group

To achieve better security, you can use a standard access list to notify the device which IP addresses are valid addresses for a content engine attempting to register with the current device. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```
Device(config)# access-list 10 permit host 10.1.1.1
Device(config)# access-list 10 permit host 10.1.1.2
Device(config)# access-list 10 permit host 10.1.1.3
Device(config)# ipv6 wccp web-cache group-list 10
```

To disable caching for certain clients, servers, or client/server pairs, you can use WCCP access lists. The following example shows that any requests coming from 10.1.1.1 to 10.3.1.1 will bypass the cache, and that all other requests will be serviced normally:

```
Device(config)# ipv6 wccp web-cache redirect-list 120
Device(config)# access-list 120 deny tcp host 10.1.1.1 any
Device(config)# access-list 120 deny tcp any host 10.3.1.1
Device(config)# access-list 120 permit ip any any
```

The following example configures a device to redirect web-related packets received via Gigabit Ethernet interface 0/1/0, destined to any host except 209.165.200.224:

```
Device(config)# access-list 100 deny ip any host 209.165.200.224
Device(config)# access-list 100 permit ip any any
Device(config)# ipv6 wccp web-cache redirect-list 100
```

```
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ipv6 wccp web-cache redirect in
```

Example: WCCPv2—IPv6—Configuring Outbound ACL Check

The following configuration example shows that the access list prevents traffic from network 10.0.0.0 leaving Gigabit Ethernet interface 0/1/0. Because the outbound ACL check is enabled, WCCP does not redirect that traffic. WCCP checks packets against the ACL before they are redirected.

```
Device(config)# ipv6 wccp web-cache
Device(config)# ipv6 wccp check acl outbound
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip access-group 10 out
Device(config-if)# exit
Device(config)# ipv6 wccp web-cache redirect-list redirect-out
Device(config)# access-list 10 deny 10.0.0.0 0.255.255.255
Device(config)# access-list 10 permit any
```

If the outbound ACL check is disabled, the HTTP packets from network 10.0.0.0 would be redirected to a web cache. Users with that network address could retrieve web pages even though the network administrator wanted to prevent it.

Example: WCCPv2—IPv6—Verifying WCCP Settings

The following example shows how to verify your configuration changes by using the **more system:running-config** command in privileged EXEC mode. The following example shows that both the web cache service and dynamic service 99 are enabled on the device:

```
Device# more system:running-config

 Building configuration...
 Current configuration:
 !
 version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 service udp-small-servers
 service tcp-small-servers
 !
 hostname router4
 !
 enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTh1
 enable password password1
 !
 ip subnet-zero
 ipv6 wccp web-cache
 ipv6 wccp 99
 ip domain-name cisco.com
 ip name-server 10.1.1.1
 ip name-server 10.1.1.2
 ip name-server 10.1.1.3
 !
 !
 !
 interface GigabitEthernet0/1/1
 ip address 10.3.1.2 255.255.255.0
 no ip directed-broadcast
```

```
ipv6 wccp web-cache redirect in
ipv6 wccp 99 redirect in
no ip route-cache
no ip mroute-cache
!
interface GigabitEthernet0/1/0
ip address 10.4.1.1 255.255.255.0
no ip directed-broadcast
ipv6 wccp 99 redirect in
no ip route-cache
no ip mroute-cache
!
interface Serial0
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password password1
login
!
end
```

The following example shows how to display global statistics related to WCCP:

```
Device# show ipv6 wccp web-cache detail

WCCP Client information:
WCCP Client ID:      10.1.1.2
Protocol Version:    2.0
State:               Usable
Redirection:         L2
Packet Return:       L2
Packets Redirected:  0
Connect Time:        00:20:34
Assignment:          MASK
Mask  SrcAddr    DstAddr    SrcPort DstPort
----  -------    -------    ------- -------
0000: 0x00000000 0x00001741 0x0000  0x0000
Value SrcAddr    DstAddr   SrcPort DstPort CE-IP
----- -------    -------   ------- ------- -----
0000: 0x00000000 0x00000000 0x0000 0x0000 0x3C010102 (10.1.1.2)
0001: 0x00000000 0x00000001 0x0000 0x0000 0x3C010102 (10.1.1.2)
0002: 0x00000000 0x00000040 0x0000 0x0000 0x3C010102 (10.1.1.2)
```

```
0003: 0x00000000 0x00000041 0x0000 0x0000 0x3C010102 (10.1.1.2)
0004: 0x00000000 0x00000100 0x0000 0x0000 0x3C010102 (10.1.1.2)
0005: 0x00000000 0x00000101 0x0000 0x0000 0x3C010102 (10.1.1.2)
0006: 0x00000000 0x00000140 0x0000 0x0000 0x3C010102 (10.1.1.2)
```

For more information about the **show ip wccp web-cache** command, see the *Cisco IOS IP Application Services Command Reference* document.

Example: WCCPv2—IPv6—Cisco ASR 1000 Platform Specific Configuration

The following example shows how to display platform-specific configuration and IPv6 counters information related to WCCP on a Cisco ASR 1000 Series Aggregation Services Router:

```
Device# show platform software wccp service-number ipv6 counters
Service Group (1, 61, 0) counters
        Unassigned count = 0
        Dropped due to closed service count = 0
        Bypass count = 0
        Bypass failed count = 0
        Denied count = 0
        Redirect count = 4
  CE = 2001:1:100::105, obj_id = 213, Redirect Packets = 4
```

The following example shows how to display platform-specific configuration and route processor slot information related to WCCP on a Cisco ASR 1000 Series Aggregation Services Router:

```
Device# show platform software wccp rp active service-number ipv6
IPV6 Dynamic service 61
Priority: 34, Number of clients: 1
Assign Method: Hash, Fwd Method: GRE, Ret Method: GRE
L4 proto: 6, Use Source Port: No
Is closed: No
```

The following example shows how to display platform-specific configuration and embedded service processor slot information related to WCCP on a Cisco ASR 1000 Series Aggregation Services Router:

```
Device# show platform software wccp fp active service-number ipv6
IPV6 Dynamic service 61
Priority: 34, Number of clients: 1
Assign Method: Hash, Fwd Method: GRE, Ret Method: GRE
Is closed: No
Current ACE: 0, Pending ACE: 0
New ACE: 0, New ACE completed: No
ACL id: 0
  AOM id: 0x18a, status: created
```

The following example shows how to display the WCCP service group information in the active Cisco Quantum Flow Processor (QFP) on a Cisco ASR 1000 Series Aggregation Services Router:

```
Device# show platform hardware qfp active feature wccp service id service-id ipv6
Service ID: 61
Service Type: 1
Service Priority: 34
Assign Method: 1
Hash key: 0x51
Hash buckets ppe address: 0x8bceb600
Mode: Open
State: Active
Number of Caches in this service: 1
```

```
  ce index: 0
  cache_id : 11
  Cache ip addr : 0x20010001
  Cache cfg ppe addr : 0x8bcab200
  Cache oce ppe addr : 0x891a7670
  Cache state ppe addr : 0x8bcfd288
Number of interfaces using this service: 1
  Interface: GigabitEthernet0/0/0.1
  cpp-if-h: 12
  Dir: 0
  pal-if-h: 15
  uidb sb ppe addr: 0x8bd308e0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IP addressing and services commands and configuration tasks	• *Cisco IOS IP Addressing Services Configuration Guide* • *Cisco IOS IP Addressing Services Command Reference*
WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

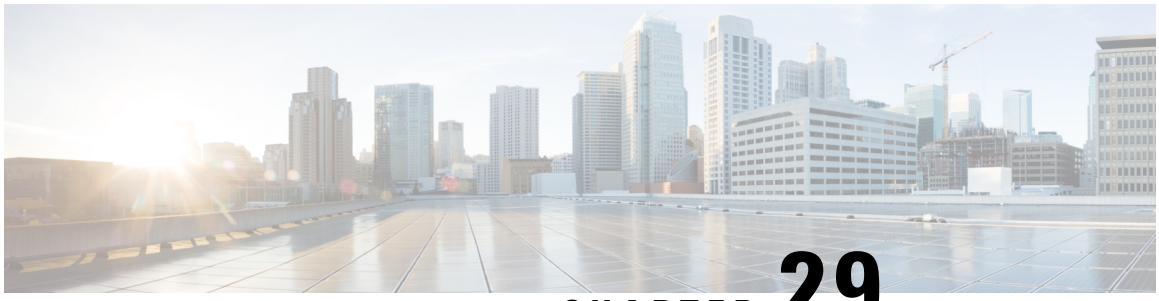
Feature Information for WCCPv2—IPv6 Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 42: Feature Information for WCCPv2 —IPv6 Support

Feature Name	Releases	Feature Information
WCCPv2—IPv6 Support	15.1(1)SY1 15.2(3)T	This feature introduces support for Web Cache Communication Protocol version 2 (WCCPv2) in an IPv6 environment. WCCP is a Cisco-developed content-routing technology that intercepts IP packets and redirects those packets to a destination other than that specified in the IP packet. Multiple routers can use WCCPv2 to service a content engine cluster. In WCCPv1, only one router could redirect content requests to a cluster. The following commands were added: **clear ipv6 wccp**, **clear wccp**, **debug ipv6 wccp**, **debug wccp**, **ipv6 wccp**, **ipv6 wccp check acl outbound**, **ipv6 wccp check services all**, **ipv6 wccp group-listen**, **ipv6 wccp redirect**, **ipv6 wccp redirect exclude in ipv6 wccp source-interface**, **show ipv6 wccp**, **show ipv6 wccp global counters**, **show wccp**, **show wccp global counters**, **show platform software wccp** *service-number* **ipv6 counters**, **show platform software wccp rp active** *service-number* **ipv6** , **show platform software wccp fp active** *service-number* **ipv6** , **show platform hardware qfp active feature wccp service id** *service-number* **ipv6** .



CHAPTER 29

WCCP with Generic GRE Support

Extended Web Cache Communication Protocol (WCCP) supports multipoint generic routing encapsulation (mGRE) return method on Cisco IOS devices. GRE-negotiated return is not supported on the Cisco Wide Area Application Services (WAAS) AppNav I/O module (IOM), customers need to use generic GRE tunnels (multipoint GRE) on the devices.

Restrictions for WCCP with Generic GRE Support

- Generic GRE tunnel does not work with a loopback source address. Because the highest numbered loopback is reserved for WCCP, customers need to use the second highest loopback address.
- WCCP traffic redirection does not work when a zone-based policy firewall is configured on a Cisco Aggregation Services Router that is configured with Cisco AppNav I/O modules. Cisco AppNav is a wide-area networking optimization solution. For WCCP traffic redirection to work, remove the zone-based policy firewall configuration from interfaces. If you are using a WAVE device, WCCP traffic redirection works correctly.
- Static and dynamic NAT with generic GRE and dynamic NAT with Layer 2 do not work when used with hardware-based Cisco AppNav appliances (for example, Wide Area Application Services [WAAS]).

Information About WCCP with Generic GRE Support

WCCP with Generic GRE Support

The generic routing encapsulation (GRE) negotiated return is not supported on AppNav I/O Module (IOM), the customers need to use Generic GRE tunnels (multipoint GRE [mGRE]) on devices. That is, a mGRE tunnel needs to be configured manually on the router if the AppNav is configured with Generic GRE return method.

Note If two multipoint generic routing encapsulation (mGRE) tunnels are configured (one programmatically generated and the other manually created) on a device, and have the same key or exist in the same VRF, do one of the following:

- Configure both tunnels with different loopback addresses.
- Configure a physical interface on manually created tunnel, and configure a loopback address on the programmatically generated tunnel.

This feature focuses on the interactions between AppNav IOM and the router. The Cisco Wide Area Application Services (WAAS) AppNav must be configured as a device mode application-accelerator and interception method WCCP.

Cisco WAAS AppNav Solution

Cisco Wide Area Application Services (WAAS) AppNav is a hardware and software solution that simplifies network integration of WAN optimization. It also overcomes the challenges related to provisioning, visibility, scalability, asymmetry, and high availability. Only a Wide Area Virtualization Engine (WAVE) appliance that contains a Cisco AppNav Controller (ANC) Interface Module can operate as an ANC. AppNav is configured as Web Cache Communication Protocol (WCCP) client of the router.

For more information on Cisco WAAS AppNav and how to configure Cisco WAAS AppNav, see "Configuring AppNav" chapter in *Cisco Wide Area Application Services Configuration Guide*.

How to Configure WCCP with Generic GRE Support

Configure WCCP Redirection with Generic GRE Configured on the Device Using a Loopback Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *loopback-nterface-number*
4. **ip address** *ip-address subnet-mask*
5. **no shutdown**
6. **exit**
7. **interface loopback** *loopback-interface-number*
8. **ip address** *ip-address subnet-mask*
9. **no shutdown**
10. **exit**
11. **ip wccp source-interface loopback** *loopback-interface-number*
12. **interface Tunnel** *tunnel-interface-number*
13. **ip address** *ip-address subnet-mask*

14. **no shutdown**
15. **no ip redirects**
16. **ip wccp redirect exclude in**
17. **tunnel source loopback** *loopback-interface-number*
18. **tunnel mode gre multipoint**
19. **end**
20. **show ip wccp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *loopback-nterface-number* **Example:** `Device(config)# interface loopback 100`	Enters interface configuration for the device.
Step 4	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.10.10.1 255.255.255.255`	Sets a primary IP address for the loopback interface.
Step 5	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the loopback interface if the interface is down.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 7	**interface loopback** *loopback-interface-number* **Example:** `Device(config)# interface loopback 1000`	Enters interface configuration for the device.

	Command or Action	Purpose
Step 8	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.11.10.1 255.255.255.255`	Sets a primary IP address for the loopback interface.
Step 9	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the loopback interface if the interface is down.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 11	**ip wccp source-interface loopback** *loopback-interface-number* **Example:** `Device(config)# ip wccp source-interface loopback 1000`	Configures a preferred Web Cache Communication Protocol (WCCP) router ID.
Step 12	**interface Tunnel** *tunnel-interface-number* **Example:** `Device(config)# interface Tunnel 10`	Enters tunnel interface configuration mode.
Step 13	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.10.20.1 255.255.255.0`	Sets a primary IP address for the tunnel interface.
Step 14	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the tunnel interface if the interface is down.
Step 15	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables the sending of ICMP redirect messages to learn routes. This command is enabled by default.
Step 16	**ip wccp redirect exclude in** **Example:** `Device(config-if)# ip wccp redirect exclude in`	Specifies that packets received on this interface be excluded from any egress redirection.

	Command or Action	Purpose
Step 17	**tunnel source loopback** *loopback-interface-number* **Example:** `Device(config-if)# tunnel source loopback 100`	Configures the loopback interface as the tunnel source.
Step 18	**tunnel mode gre multipoint** **Example:** `Device(config-if)# tunnel mode gre multipoint`	Sets the global encapsulation mode on all interfaces of a device to generic routing encapsulation (GRE).
Step 19	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 20	**show ip wccp summary** **Example:** `Device# show ip wccp summary`	Displays a summary of WCCP services.

Configure WCCP Redirection with Generic GRE Configured on a Device Using a Physical Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface GigabitEthernet** *interface-id*
4. **ip address** *ip-address subnet-mask*
5. **no shutdown**
6. **exit**
7. **interface Tunnel** *tunnel-interface-number*
8. **ip address** *ip-address subnet-mask*
9. **no shutdown**
10. **no ip redirects**
11. **ip wccp redirect exclude in**
12. **tunnel source GigabitEthernet** *interface-id*
13. **tunnel mode gre multipoint**
14. **end**
15. **show ip wccp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface GigabitEthernet** *interface-id* **Example:** `Device(config)# interface GigabitEthernet0/0/1`	Enters interface configuration for the device.
Step 4	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.10.10.1 255.255.255.0`	Sets a primary IP address for the loopback interface.
Step 5	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the loopback interface if the interface is down.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 7	**interface Tunnel** *tunnel-interface-number* **Example:** `Device(config)# interface Tunnel 10`	Enters tunnel interface configuration mode.
Step 8	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.10.20.1 255.255.255.0`	Sets a primary IP address for the tunnel interface.
Step 9	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the tunnel interface if the interface is down.

	Command or Action	Purpose
Step 10	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables the sending of ICMP redirect messages to learn routes. This command is enabled by default.
Step 11	**ip wccp redirect exclude in** **Example:** `Device(config-if)# ip wccp redirect exclude in`	Specifies that packets received on this interface be excluded from any egress redirection.
Step 12	**tunnel source GigabitEthernet** *interface-id* **Example:** `Device(config-if)# tunnel source GigabitEthernet0/0/1`	Configures the loopback interface as the tunnel source.
Step 13	**tunnel mode gre multipoint** **Example:** `Device(config-if)# tunnel mode gre multipoint`	Sets the global encapsulation mode on all interfaces of a device to generic routing encapsulation (GRE).
Step 14	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 15	**show ip wccp summary** **Example:** `Device# show ip wccp summary`	Displays a summary of WCCP services.

Configuration Examples for WCCP with Generic GRE Support

Example: Configure WCCP Redirection with Generic GRE Configured on Device Using a Loopback Interface

The following example shows how to configure Web Cache Communication Protocol (WCCP) redirection on the device using loopback interface when generic routing encapsulation (GRE) is enabled on the Cisco Wide Area Application Services (WAAS) AppNav:

```
Device> enable
Device# configure terminal
Device(config)#  interface loopback 100
Device(config-if)#  ip address 10.10.10.1 255.255.255.255
Device(config-if)#  no shutdown
```

```
Device(config-if)#  exit
Device(config)#  interface loopback 1000
Device(config-if)#  ip address 10.11.10.1 255.255.255.255
Device(config-if)#  no shutdown
Device(config-if)#  exit
Device(config)#  ip wccp source-interface loopback 1000
Device(config)#  interface Tunnel 10
Device(config-if)#  ip address 10.12.10.1 255.255.255.0
Device(config-if)#  no shutdown
Device(config-if)#  no ip redirects
Device(config-if)#  ip wccp redirect exclude in
Device(config-if)#  tunnel source loopback 100
Device(config-if)#  tunnel mode gre multipoint
Device(config-if)#  end
Device# show ip wccp summary

WCCP version 2 enabled, 2 services

Service  Clients  Routers  Assign  Redirect  Bypass
-------  -------  -------  ------  --------  ------
Default  routing  table  (Router Id: 10.10.10.1):
61          1      1       MASK   GRE     GRE
62          1      1       MASK   GRE     GRE
```

Example: Configure WCCP Redirection with Generic GRE Configured on a Device Using a Physical Interface

The following example shows how to configure Web Cache Communication Protocol (WCCP) redirection on the device using a physical interface when generic routing encapsulation (GRE) is enabled on the Cisco Wide Area Application Services (WAAS) AppNav:

```
Device> enable
Device# configure terminal
Device(config)#  interface GigabitEthernet0/0/1
Device(config-if)# ip address 10.12.10.1 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# exit
Device(config)# interface Tunnel 10
Device(config-if)# ip address 10.13.10.1 255.255.255.0
Device(config-if)# no ip redirects
Device(config-if)# ip wccp redirect exclude in
Device(config-if)# tunnel source GigabitEthernet0/0/1
Device(config-if)# tunnel mode gre multipoint
Device(config-if)# end
Device# show ip wccp summary

WCCP version 2 enabled, 2 services

Service  Clients  Routers  Assign  Redirect  Bypass
-------  -------  -------  ------  --------  ------
Default  routing  table   (Router Id: 10.10.10.1):
61          1      1       MASK      GRE     GRE
62          1      1       MASK      GRE     GRE
```

Additional References for WCCP with Generic GRE Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP addressing and services commands and configuration tasks	• IP Addressing: IPv4 Addressing Configuration Guide • Cisco IOS IP Addressing Services Command Reference
WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Application Services Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for WCCP with Generic GRE Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

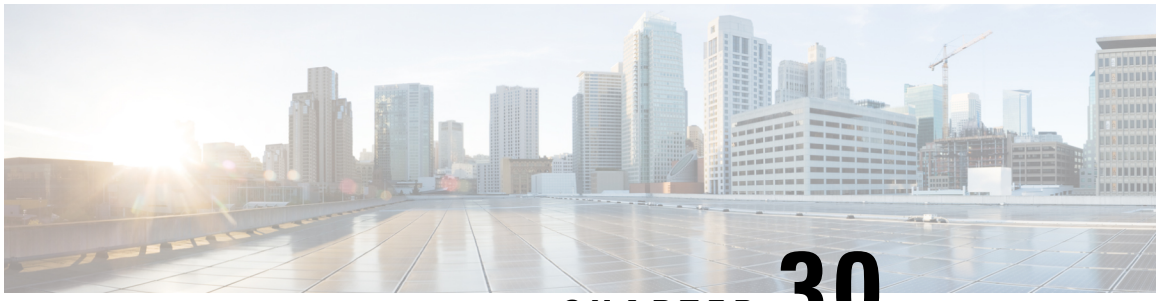
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 43: Feature Information for WCCP with Generic GRE Support

Feature Name	Releases	Feature Information
WCCP with Generic GRE Support	Cisco IOS XE Release 3.10.2	This feature provides extended WCCP support to use Generic GRE tunnels (multipoint GRE) on the devices when generic routing encapsulation (GRE) negotiated return is not supported on AppNav I/O Module (IOM).

PART IV

IP SLAs



CHAPTER 30

IP SLAs Overview

This module describes IP Service Level Agreements (SLAs). IP SLAs allows Cisco customers to analyze IP service levels for IP applications and services, to increase productivity, to lower operational costs, and to reduce the frequency of network outages. IP SLAs uses active traffic monitoring--the generation of traffic in a continuous, reliable, and predictable manner--for measuring network performance. Using IP SLAs, service provider customers can measure and provide service level agreements, and enterprise customers can verify service levels, verify outsourced service level agreements, and understand network performance. IP SLAs can perform network assessments, verify quality of service (QoS), ease the deployment of new services, and assist administrators with network troubleshooting. IP SLAs can be accessed using the Cisco software commands or Simple Network Management Protocol (SNMP) through the Cisco Round-Trip Time Monitor (RTTMON) and syslog Management Information Bases (MIBs).

Information About IP SLAs

IP SLAs Technology Overview

Cisco IP SLAs uses active traffic monitoring--the generation of traffic in a continuous, reliable, and predictable manner--for measuring network performance. IP SLAs sends data across the network to measure performance between multiple network locations or across multiple network paths. It simulates network data and IP services, and collects network performance information in real time. The information collected includes data about response time, one-way latency, jitter (interpacket delay variance), packet loss, voice quality scoring, network resource availability, application performance, and server response time. IP SLAs performs active monitoring by generating and analyzing traffic to measure performance either between Cisco devices or from a Cisco device to a remote IP device such as a network application server. Measurement statistics provided by the various IP SLAs operations can be used for troubleshooting, for problem analysis, and for designing network topologies.

Using IP SLAs, service provider customers can measure and provide service level agreements, and enterprise customers can verify service levels, verify outsourced service level agreements, and understand network performance for new or existing IP services and applications. IP SLAs uses unique service level assurance metrics and methodology to provide highly accurate, precise service level assurance measurements.

Depending on the specific IP SLAs operation, statistics of delay, packet loss, jitter, packet sequence, connectivity, path, server response time, and download time can be monitored within the Cisco device and

stored in both CLI and SNMP MIBs. The packets have configurable IP and application layer options such as a source and destination IP address, User Datagram Protocol (UDP)/TCP port numbers, a type of service (ToS) byte (including Differentiated Services Code Point [DSCP] and IP Prefix bits), a Virtual Private Network (VPN) routing/forwarding instance (VRF), and a URL web address.

Being Layer-2 transport independent, IP SLAs can be configured end-to-end over disparate networks to best reflect the metrics that an end-user is likely to experience. Performance metrics collected by IP SLAs operations include the following:

- Delay (both round-trip and one-way)
- Jitter (directional)
- Packet loss (directional)
- Packet sequencing (packet ordering)
- Path (per hop)
- Connectivity (directional)
- Server or website download time
- Voice quality scores

Because IP SLAs is accessible using SNMP, it also can be used by performance monitoring applications like CiscoWorks Internetwork Performance Monitor (IPM) and other third-party Cisco partner performance management products. For details about network management products that use IP SLAs, see http://www.cisco.com/go/ipsla .

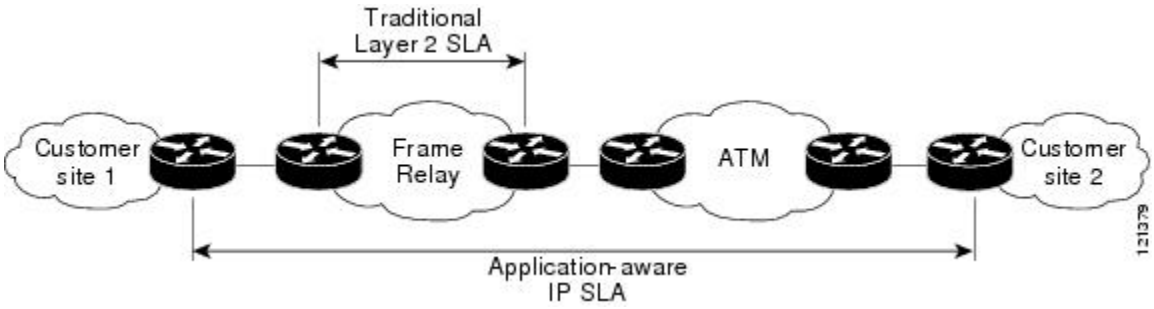
SNMP notifications based on the data gathered by an IP SLAs operation allow the router to receive alerts when performance drops below a specified level and when problems are corrected. IP SLAs uses the Cisco RTTMON MIB for interaction between external Network Management System (NMS) applications and the IP SLAs operations running on the Cisco devices. For a complete description of the object variables referenced by the IP SLAs feature, refer to the text of the CISCO-RTTMON-MIB.my file, available from the Cisco MIB website .

Service Level Agreements

Internet commerce has grown significantly in the past few years as the technology has advanced to provide faster, more reliable access to the Internet. Many companies now need online access and conduct most of their business online and any loss of service can affect the profitability of the company. Internet service providers (ISPs) and even internal IT departments now offer a defined level of service--a service level agreement--to provide their customers with a degree of predictability.

The latest performance requirements for business-critical applications, voice over IP (VoIP) networks, audio and visual conferencing, and VPNs are creating internal pressures on converged IP networks to become optimized for performance levels. Network administrators are increasingly required to support service level agreements that support application solutions. The figure below shows how IP SLAs has taken the traditional concept of Layer 2 service level agreements and applied a broader scope to support end-to-end performance measurement, including support of applications.

Figure 35: Scope of Traditional Service Level Agreement Versus IP SLAs



IP SLAs provides the following improvements over a traditional service level agreement:

- End-to-end measurements--The ability to measure performance from one end of the network to the other allows a broader reach and more accurate representation of the end-user experience.
- Sophistication--Statistics such as delay, jitter, packet sequence, Layer 3 connectivity, and path and download time that are broken down into bidirectional and round-trip numbers provide more data than just the bandwidth of a Layer 2 link.
- Ease of deployment--Leveraging the existing Cisco devices in a large network makes IP SLAs easier and cheaper to implement than the physical probes often required with traditional service level agreements.
- Application-aware monitoring--IP SLAs can simulate and measure performance statistics generated by applications running over Layer 3 through Layer 7. Traditional service level agreements can only measure Layer 2 performance.
- Pervasiveness--IP SLAs support exists in Cisco networking devices ranging from low-end to high-end devices and switches. This wide range of deployment gives IP SLAs more flexibility over traditional service level agreements.

When you know the performance expectations for different levels of traffic from the core of your network to the edge of your network, you can confidently build an end-to-end application-aware service level agreement.

Benefits of IP SLAs

- IP SLAs monitoring
  - Provides service level agreement monitoring, measurement, and verification.
- Network performance monitoring
  - Measures the jitter, latency, or packet loss in the network.
  - Provides continuous, reliable, and predictable measurements.
- IP service network health assessment
  - Verifies that the existing QoS is sufficient for new IP services.
- Edge-to-edge network availability monitoring

  - Provides proactive verification and connectivity testing of network resources (for example, indicates the network availability of a Network File System (NFS) server used to store business critical data from a remote site).

- Troubleshooting of network operation
  - Provides consistent, reliable measurement that immediately identifies problems and saves troubleshooting time.

- Voice over IP (VoIP) performance monitoring
- Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) performance monitoring and network verification

Restriction for IP SLAs

- With *SR_5_label_push* template, IP SLA DMM is not supported on RSP3 module.
- The maximum supported scale number of CFM and IP SLA over the port channel is only .

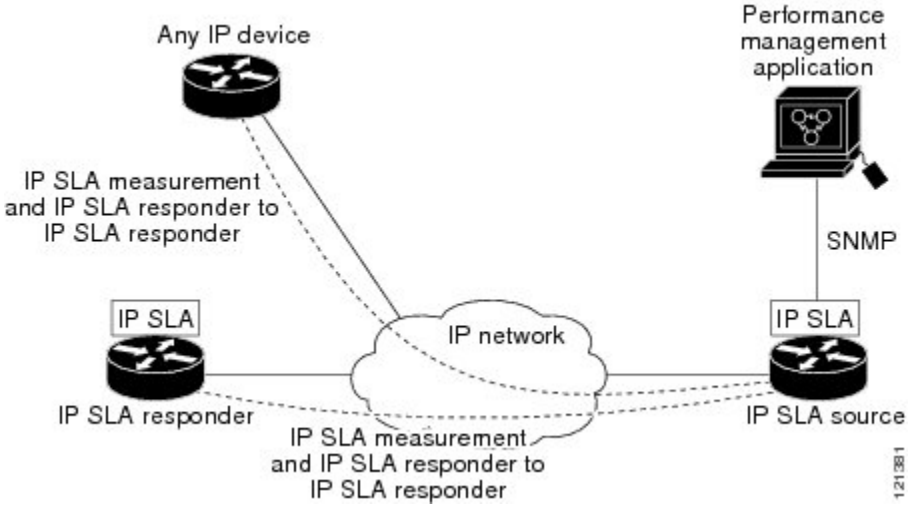
Network Performance Measurement Using IP SLAs

Using IP SLAs, a network engineer can monitor the performance between any area in the network: core, distribution, and edge. Monitoring can be done anytime, anywhere, without deploying a physical probe.

The IP SLAs Probe Enhancements feature is an application-aware synthetic operation agent that monitors network performance by measuring response time, network resource availability, application performance, jitter (interpacket delay variance), connect time, throughput, and packet loss. Performance can be measured between any Cisco device that supports this feature and any remote IP host (server), Cisco routing device, or mainframe host. Performance measurement statistics provided by this feature can be used for troubleshooting, for problem analysis, and for designing network topologies.

IP SLAs uses generated traffic to measure network performance between two networking devices. The figure below shows how IP SLAs starts when the IP SLAs device sends a generated packet to the destination device. After the destination device receives the packet, and depending on the type of IP SLAs operation, the device will respond with time-stamp information for the source to make the calculation on performance metrics. An IP SLAs operation performs a network measurement from the source device to a destination in the network using a specific protocol such as UDP.

Figure 36: IP SLAs Operations



To implement IP SLAs network performance measurement you need to perform these tasks:

1. Enable the IP SLAs Responder, if appropriate.
2. Configure the required IP SLAs operation type.
3. Configure any options available for the specified IP SLAs operation type.
4. Configure threshold conditions, if required.
5. Schedule the operation to run, then let the operation run for a period of time to gather statistics.
6. Display and interpret the results of the operation using Cisco software commands or an NMS system with SNMP.

IP SLAs Responder and IP SLAs Control Protocol

The IP SLAs Responder is a component embedded in the destination Cisco routing device that allows the system to anticipate and respond to IP SLAs request packets. The IP SLAs Responder provides an enormous advantage with accurate measurements without the need for dedicated probes and additional statistics not available via standard ICMP-based measurements. The patented IP SLAs Control Protocol is used by the IP SLAs Responder providing a mechanism through which the responder can be notified on which port it should listen and respond. Only a Cisco device can be a source for a destination IP SLAs Responder.

The figure "IP SLAs Operations" in the "Network Performance Measurement Using IP SLAs" section shows where the IP SLAs Responder fits in relation to the IP network. The IP SLAs Responder listens on a specific port for control protocol messages sent by an IP SLAs operation. Upon receipt of the control message, the responder will enable the specified UDP or TCP port for the specified duration. During this time, the responder accepts the requests and responds to them. The responder disables the port after it responds to the IP SLAs packet, or when the specified time expires. For added security, MD5 authentication for control messages is available.

Enabling the IP SLAs Responder on the destination device is not required for all IP SLAs operations. For example, if services that are already provided by the destination device (such as Telnet or HTTP) are chosen,

the IP SLAs Responder need not be enabled. For non-Cisco devices, the IP SLAs Responder cannot be configured and IP SLAs can send operational packets only to services native to those devices.

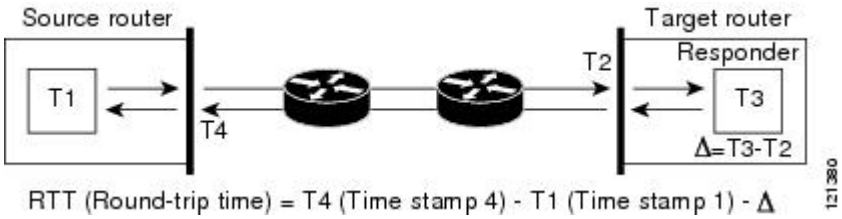
Response Time Computation for IP SLAs

Devices may take tens of milliseconds to process incoming packets, due to other high-priority processes. This delay affects the response times because the reply to test packets might be sitting on queue while waiting to be processed. In this situation, the response times would not accurately represent true network delays. IP SLAs minimizes these processing delays on the source device as well as on the target device (if IP SLAs Responder is being used), in order to determine true round-trip times. IP SLAs test packets use time stamping to minimize the processing delays.

When enabled, the IP SLAs Responder allows the target device to take two time stamps both when the packet arrives on the interface at interrupt level and again just as it is leaving, eliminating the processing time. At times of high network activity, an ICMP ping test often shows a long and inaccurate response time, while an IP SLAs test shows an accurate response time due to the time stamping on the responder.

The figure below demonstrates how the responder works. Four time stamps are taken to make the calculation for round-trip time. At the target device, with the responder functionality enabled time stamp 2 (TS2) is subtracted from time stamp 3 (TS3) to produce the time spent processing the test packet as represented by delta. This delta value is then subtracted from the overall round-trip time. Notice that the same principle is applied by IP SLAs on the source device where the incoming time stamp 4 (TS4) is also taken at the interrupt level to allow for greater accuracy.

Figure 37: IP SLAs Responder Time Stamping



An additional benefit of the two time stamps at the target device is the ability to track one-way delay, jitter, and directional packet loss. Because much network behavior is asynchronous, it is critical to have these statistics. However, to capture one-way delay measurements the configuration of both the source device and target device with Network Time Protocol (NTP) is required. Both the source and target need to be synchronized to the same clock source. One-way jitter measurements do not require clock synchronization.

IP SLAs Operation Scheduling

After an IP SLAs operation has been configured, you must schedule the operation to begin capturing statistics and collecting error information. When scheduling an operation, it can start immediately or start at a certain month, day, and hour. There is a pending option to set the operation to start at a later time. The pending option is also an internal state of the operation visible through SNMP. The pending state is also used when an operation is a reaction (threshold) operation waiting to be triggered. You can schedule a single IP SLAs operation or a group of operations at one time.

Multioperations scheduling allows you to schedule multiple IP SLAs operations using a single Cisco software command or the CISCO RTTMON-MIB. This feature allows you to control the amount of IP SLAs monitoring traffic by scheduling the operations to run at evenly distributed times. This distribution of IP SLAs operations helps minimize the CPU utilization and thereby enhances the scalability of the network.

For more details about the IP SLAs multioperations scheduling functionality, see the “IP SLAs-Multioperation Scheduling of IP SLAs Operations” module of the *IP SLAs Configuration Guide* .

IP SLAs Operation Threshold Monitoring

To support successful service level agreement monitoring or to proactively measure network performance, threshold functionality becomes essential. Consistent reliable measurements immediately identify issues and can save troubleshooting time. To confidently roll out a service level agreement you need to have mechanisms that notify you immediately of any possible violation. IP SLAs can send SNMP traps that are triggered by events such as the following:

- Connection loss
- Timeout
- Round-trip time threshold
- Average jitter threshold
- One-way packet loss
- One-way jitter
- One-way mean opinion score (MOS)
- One-way latency

Alternately, an IP SLAs threshold violation can trigger another IP SLAs operation for further analysis. For example, the frequency could be increased or an ICMP path echo or ICMP path jitter operation could be initiated for troubleshooting.

Determining the type of threshold and the level to set can be complex, and it depends on the type of IP service being used in the network. For more details on using thresholds with IP SLAs operations, see the “IP SLAs-Proactive Threshold Monitoring of IP SLAs Operations” module of the *IP SLAs Configuration Guide* .

MPLS VPN Awareness

The IP SLAs MPLS VPN Awareness feature provides the capability to monitor IP service levels within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). Using IP SLAs within MPLS VPNs allows service providers to plan, provision, and manage IP VPN services according to the service level agreement for a customer. IP SLAs operations can be configured for a specific VPN by specifying a VPN routing and forwarding (VRF) name.

History Statistics

IP SLAs maintains the following three types of history statistics:

- Aggregated statistics--By default, IP SLAs maintains two hours of aggregated statistics for each operation. Value from each operation cycle is aggregated with the previously available data within a given hour. The Enhanced History feature in IP SLAs allows for the aggregation interval to be shorter than an hour.

- Operation snapshot history--IP SLAs maintains a snapshot of data for each operation instance that matches a configurable filter, such as all, over threshold, or failures. The entire set of data is available and no aggregation takes place.
- Distribution statistics--IP SLAs maintains a frequency distribution over configurable intervals. Each time IP SLAs starts an operation, a new history bucket is created until the number of history buckets matches the specified size or the lifetime of the operation expires. By default, the history for an IP SLAs operation is not collected. If history is collected, each bucket contains one or more history entries from the operation. History buckets do not wrap.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco ACNS software configuration information	• *Cisco ACNS Software Caching Configuration Guide, Release 4.2* • Cisco ACNS Software listing page on Cisco.com
IP access list overview, configuration tasks, and commands	*Cisco IOS Security Command Reference*
IP addressing and services commands and configuration tasks	• *Cisco IOS IP Addressing Services Configuration Guide* • *Cisco IOS IP Addressing Services Command Reference*
WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

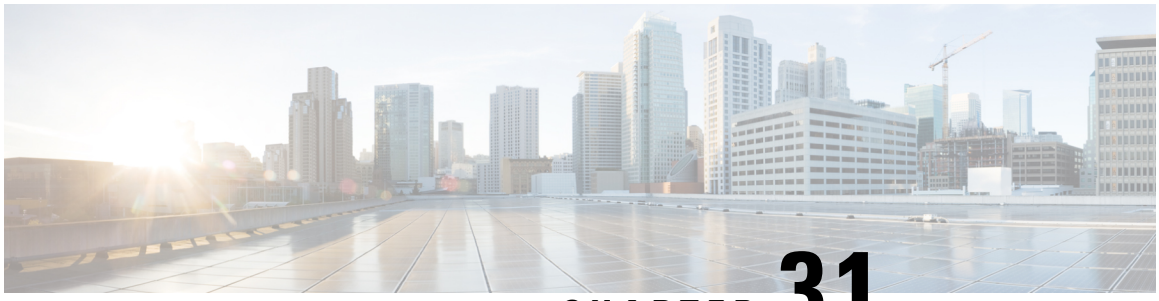
MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 31

Configuring IP SLAs UDP Jitter Operations

This document describes how to configure an IP Service Level Agreements (SLAs) UDP jitter operation to analyze round-trip delay, one-way delay, one-way jitter, one-way packet loss, and connectivity in networks that carry UDP traffic in IPv4 or IPv6 networks. This module also explains how the data gathered using the UDP jitter operation can be displayed and analyzed using Cisco software commands.

Prerequisites for IP SLAs UDP Jitter Operations

- Time synchronization, such as that provided by the Network Time Protocol (NTP), is required between the source and the target device to provide accurate one-way delay (latency) measurements. To configure NTP on source and target devices, perform the tasks in the "Performing Basic System Management" chapter of the *Basic System Management Configuration Guide*. Time synchronization is not required for one-way jitter and packet loss measurements. If time is not synchronized between source and target devices, one-way jitter and packet loss data are returned, but values of "0" are returned for the one-way delay measurements provided by the UDP jitter operation.
- Before configuring any IP Service Level Agreements (SLAs) application, use the **show ip sla application** command to verify that the operation type is supported on the software image.

Restrictions for IP SLAs UDP Jitter Operations

- Multiple SLA probes configured with same source and destination IP and port number must not be run simultaneously.

Information About IP SLAs UDP Jitter Operations

IP SLAs UDP Jitter Operation

The IP Service Level Agreements (SLAs) UDP jitter operation diagnoses network suitability for real-time traffic applications such as VoIP, video over IP, or real-time conferencing.

Jitter means inter-packet delay variance. When multiple packets are sent consecutively from a source to a destination, for example, 10 ms apart, and if the network is behaving ideally, the destination should receive the packets 10 ms apart. But if there are delays in the network (like queuing, arriving through alternate routes, and so on) the arrival delay between packets might be greater than or less than 10 ms. Using this example, a positive jitter value indicates that packets arrived greater than 10 ms apart. If packets arrive 12 ms apart, then positive jitter is 2 ms; if packets arrive 8 ms apart, negative jitter is 2 ms. For delay-sensitive networks like VoIP, positive jitter values are undesirable, and a jitter value of 0 is ideal.

However, the IP SLAs UDP jitter operation does more than just monitor jitter. As the UDP jitter operation includes data returned by the IP SLAs UDP operation, the UDP jitter operation can be used as a multipurpose data gathering operation. The packets that IP SLAs generate carry packet-sending and receiving sequence information, and sending and receiving time stamps from the source and the operational target. Based on this information, UDP jitter operations are capable of measuring the following:

- Per-direction jitter (source to destination and destination to source)
- Per-direction packet loss
- Per-direction delay (one-way delay)
- Round-trip delay (average round-trip time)

As paths for sending and receiving data may be different (asymmetric), the per-direction data allows you to more readily identify where congestion or other problems are occurring in the network.

The UDP jitter operation functions by generating synthetic (simulated) UDP traffic. Asymmetric probes support custom-defined packet sizes per direction with which different packet sizes can be sent in request packets (from the source device to the destination device) and in response packets (from the destination device to the source device).

The UDP jitter operation sends N number of UDP packets, each of size S, T milliseconds apart, from a source device to a destination device, at a given frequency of F. In response, UDP packets of size P is sent from the destination device to the source device. By default, ten packet frames (N), each with a payload size of 10 bytes (S), are generated every 10 ms (T), and the operation is repeated every 60 seconds (F). Each of these parameters is user-configurable, so as to best simulate the IP service that you provide, as shown in the table below.

Table 44: UDP Jitter Operation Parameters

UDP Jitter Operation Parameter	Default	Configuration Commands
Number of packets (N)	10 packets	**udp-jitter num-packets**
Payload size per request packet (S)	10 bytes	**request-data-size**

UDP Jitter Operation Parameter	Default	Configuration Commands
Payload size per response packet (P)	The default response data size varies depending on the type of IP SLAs operation configured. **Note** If the **response-data-size** command is not configured, then the response data size value is the same as the request data size value.	**response-data-size**
Time between packets, in milliseconds (T)	10 ms	**udp-jitter interval**
Elapsed time before the operation repeats, in seconds (F)	60 seconds	**frequency (IP SLA)**

The IP SLAs operations function by generating synthetic (simulated) network traffic. A single IP SLAs operation (for example, IP SLAs operation 10) repeats at a given frequency for the lifetime of the operation.

How to Configure IP SLAs UDP Jitter Operations

Configuring the IP SLAs Responder on a Destination Device

Note A responder should not configure a permanent port for a sender. If the responder configures a permanent port for a sender, even if the packets are successfully sent (no timeout or packet-loss issues), the jitter value is zero.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla responder**
   - **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla responder**
	• **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
	Example:
	`Device(config)# ip sla responder`
	`Device(config)# ip sla responder udp-echo ipaddress 192.0.2.132 port 5000 vrf vrf1`
		(Optional) Temporarily enables IP SLAs responder functionality on a Cisco device in response to control messages from the source.
		(Optional; required only if protocol control is disabled on the source.) Enables IP SLAs responder functionality on the specified IP address, port and VRF.
		• Protocol control is enabled by default.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring and Scheduling a UDP Jitter Operation on a Source Device

Perform only one of the following tasks:

- Configuring a Basic UDP Jitter Operation on a Source Device
- Configuring a UDP Jitter Operation with Additional Characteristics

Configuring a Basic UDP Jitter Operation on a Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]
5. **frequency** *seconds*
6. **end**
7. **show ip sla configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Starts configuring an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Device(config-ip-sla)# udp-jitter 192.0.2.135 5000`	Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration mode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and destination devices.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-jitter)# end`	Exits UDP Jitter configuration mode and returns to privileged EXEC mode.
Step 7	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

What to do next

To configure the percentile option for your operation, see the “Configuring the IP SLAs—Percentile Support for Filtering Outliers” module.

Configuring a UDP Jitter Operation with Additional Characteristics

Note

- The IP Service Level Agreements (SLAs) UDP jitter operation does not support the IP SLAs History feature because of the large volume of data involved with UDP jitter operations. This means that the following commands are not supported for UDP jitter operations: **history buckets-kept**, **history filter**, **history lives-kept**, **samples-of-history-kept**, and **show ip sla history**.
- The MIB used by IP SLAs (CISCO-RTTMON-MIB) limits the hours-of-statistics kept for the UDP jitter operation to two hours. Configuring a larger value using the **history hours-of-statistics** *hours* global configuration change does not increase the value beyond two hours. However, the Data Collection MIB can be used to collect historical data for the operation. For more information, see the CISCO-DATA-COLLECTION-MIB.

Before you begin

Before configuring a UDP jitter operation on a source device, the IP SLAs Responder must be enabled on the target device (the operational target). The IP SLAs Responder is available only on Cisco IOS software-based devices. To enable the Responder, perform the task in the "Configuring the IP SLAs Responder on the Destination Device" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]
5. **history distributions-of-statistics-kept** *size*
6. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
7. **frequency** *seconds*
8. **history hours-of-statistics-kept** *hours*
9. **owner** *owner-id*
10. **request-data-size** *bytes*
11. **response-data-size** *bytes*
12. **history statistics-distribution-interval** *milliseconds*
13. **tag** *text*
14. **threshold** *milliseconds*
15. **timeout** *milliseconds*
16. Enter one of the following commands:
    - **tos** *number*
    - **traffic-class** *number*
17. **flow-label** *number*
18. **verify-data**
19. **vrf** *vrf-name*
20. **end**

21. **show ip sla configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Device(config-ip-sla)# udp-jitter 192.0.2.134 5000`	Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration mode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both source and target devices.
Step 5	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-jitter)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop for an IP SLAs operation.
Step 6	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-jitter)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 7	**frequency** *seconds* **Example:** `Device(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 8	**history hours-of-statistics-kept** *hours* **Example:**	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.

	Command or Action	Purpose
	`Device(config-ip-sla-jitter)# history hours-of-statistics-kept 4`	
Step 9	**owner** *owner-id* **Example:** `Device(config-ip-sla-jitter)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 10	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-jitter)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation request packet.
Step 11	**response-data-size** *bytes* **Example:** `Device(config-ip-sla-jitter)# response-data-size 25`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation response packet.
Step 12	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-jitter)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 13	**tag** *text* **Example:** `Device(config-ip-sla-jitter)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 14	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-jitter)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 15	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-jitter)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 16	Enter one of the following commands: • **tos** *number* • **traffic-class** *number* **Example:** `Device(config-ip-sla-jitter)# tos 160`	(Optional) Defines the ToS byte in the IPv4 header of an IP SLAs operation. or (Optional) Defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

	Command or Action	Purpose
	`Device(config-ip-sla-jitter)# traffic-class 160`	
Step 17	**flow-label** *number* **Example:** `Device(config-ip-sla-jitter)# flow-label 112233`	(Optional) Defines the flow label field in the IPv6 header for a supported IP SLAs operation.
Step 18	**verify-data** **Example:** `Device(config-ip-sla-jitter)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 19	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-jitter)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) VPNs using IP SLAs operations.
Step 20	**end** **Example:** `Device(config-ip-sla-jitter)# end`	Exits UDP jitter configuration mode and returns to privileged EXEC mode.
Step 21	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

What to do next

To configure the percentile option for your operation, see the "Configuring the IP SLAs—Percentile Support for Filtering Outliers" module.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:

- **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
- **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]

4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range*	**schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever**	*seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm* [:*ss*]}] **Example:** `Device(config)# ip sla schedule 10 life forever start-time now` `Device(config)# ip sla group schedule 10 schedule-period frequency` `Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now` `Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Verifying IP SLAs UDP Jitter Operations

SUMMARY STEPS

1. **enable**
2. **show ip sla configuration**
3. **show ip sla group schedule**
4. **show ip sla statistics**
5. **show ip sla statistics 2 details**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show ip sla configuration**

Displays IP SLAs configuration details.

Example:

```
Device# show ip sla configuration

IP SLAs Infrastructure Engine-III
Entry number: 5
Owner: ownername
Tag: text
Operation timeout (milliseconds): 9999
Type of operation to perform: udp-jitter
Target address/Source address: 192.0.2.115/0.0.0.0
Target port/Source port: 5/0
Type Of Service parameter: 0x5
Request size (ARR data portion): 100
Response size (ARR data portion): 200
Packet Interval (milliseconds)/Number of packets: 20/10
Verify data: No
Operation Stats Precision : microseconds
Timestamp Location Optimization: enabled
Operation Packet Priority : high
NTP Sync Tolerance : 0 percent
Vrf Name:
Control Packets: enabled
```

Step 3 **show ip sla group schedule**

Displays IP SLAs group schedule details.

Example:

```
Device# show ip sla group schedule

Group Entry Number: 1
Probes to be scheduled: 6-9,3-4
Total number of probes: 6
Schedule period: 10
Mode: even
Group operation frequency: Equals schedule period
Status of entry (SNMP RowStatus): Active
Next Scheduled Start Time: Pending trigger
Life (seconds): 3600
Entry Ageout (seconds): never
```

Step 4 **show ip sla statistics**

Displays IP SLAs statistics.

Example:

```
Device# show ip sla statistics

Type of operation: udp-jitter
Packet Loss Values:
Loss Source to Destination: 19
Source to Destination Loss Periods Number: 19
Source to Destination Loss Period Length Min/Max: 1/1
Source to Destination Inter Loss Period Length Min/Max: 1/546
Loss Destination to Source: 0
Destination to Source Loss Periods Number: 0
Destination to Source Loss Period Length Min/Max: 0/0
Destination to Source Inter Loss Period Length Min/Max: 0/0
Out Of Sequence: 0 Tail Drop: 0
Packet Late Arrival: 0 Packet Skipped: 0
```

- udp-jitter has the ability to detect in which direction a packet was lost in. It also calculates statistics about the periods of packet loss
- Loss Source to Destination: 19—Indicates that 19 packets were sent from the sender but never reached the responder.
- Source to Destination Loss Periods Number: 19—Indicates that there were 19 incidents of packet loss (an incident of packet loss is a period where packets are lost, irrespective of the actual number of lost packets.)
- Source to Destination Loss Period Length Min/Max: 1/1—indicates that all packets lost in this direction are isolated; there are no instances of multiple lost packets back-to-back.
- Source to Destination Inter Loss Period Length Min/Max: 1/546—indicates that the minimum gap between lost packets is 1, and the maximum gap between successive packet losses is 546 successfully sent packets.

Step 5 **show ip sla statistics 2 details**

Displays IPSLAs latest operation statistics

Example:

```
Device# show ip sla statistics 2 details

IPSLA operation id: 2
Type of operation: udp-jitter
Latest RTT: 1 milliseconds
Latest operation start time: 07:45:28 GMT Thu Aug 28 2014
Latest operation return code: OK
Over thresholds occurred: FALSE
RTT Values:
Number Of RTT: 10 RTT Min/Avg/Max: 1/1/1 milliseconds
Latency one-way time:
Number of Latency one-way Samples: 6
Source to Destination Latency one way Min/Avg/Max: 1/1/1 milliseconds
Destination to Source Latency one way Min/Avg/Max: 0/0/0 milliseconds
Source to Destination Latency one way Sum/Sum2: 6/6
Destination to Source Latency one way Sum/Sum2: 0/0
Jitter Time:
Number of SD Jitter Samples: 9
Number of DS Jitter Samples: 9
Source to Destination Jitter Min/Avg/Max: 0/1/1 milliseconds
Destination to Source Jitter Min/Avg/Max: 0/0/0 milliseconds
Source to destination positive jitter Min/Avg/Max: 1/1/1 milliseconds
Source to destination positive jitter Number/Sum/Sum2: 3/3/3
Source to destination negative jitter Min/Avg/Max: 1/1/1 milliseconds
Source to destination negative jitter Number/Sum/Sum2: 3/3/3
```

```
Destination to Source positive jitter Min/Avg/Max: 0/0/0 milliseconds
Destination to Source positive jitter Number/Sum/Sum2: 0/0/0
Destination to Source negative jitter Min/Avg/Max: 0/0/0 milliseconds
Destination to Source negative jitter Number/Sum/Sum2: 0/0/0
Interarrival jitterout: 0 Interarrival jitterin: 0
Jitter AVG: 1
Over Threshold:
Number Of RTT Over Threshold: 0 (0%)
Packet Loss Values:
Loss Source to Destination: 0
Source to Destination Loss Periods Number: 0
Source to Destination Loss Period Length Min/Max: 0/0
Source to Destination Inter Loss Period Length Min/Max: 0/0
Loss Destination to Source: 0
Destination to Source Loss Periods Number: 0
Destination to Source Loss Period Length Min/Max: 0/0
Destination to Source Inter Loss Period Length Min/Max: 0/0
Out Of Sequence: 0 Tail Drop: 0 Packet Late Arrival: 0
Packet Skipped: 0
Voice Score Values:
Calculated Planning Impairment Factor (ICPIF): 0
Mean Opinion Score (MOS): 0
Number of successes: 2
Number of failures: 0
Operation time to live: Forever
Operational state of entry: Active
Last time this entry was reset: Never
```

Configuration Examples for IP SLAs UDP Jitter Operations

Example: Configuring a UDP Jitter Operation

In the following example, two operations are configured as UDP jitter operations, with operation 2 starting five seconds after the first operation. Both operations will run indefinitely.

```
configure terminal
 ip sla 1
  udp-jitter 192.0.2.115 65051 num-packets 20
  request-data-size 160
  tos 128
  frequency 30
 ip sla schedule 1 start-time after 00:05:00
 ip sla 2
  udp-jitter 192.0.2.115 65052 num-packets 20 interval 10
  request-data-size 20
  tos 64
  frequency 30
 ip sla schedule 2 start-time after 00:05:05
```

Enter the following command on the target (destination) device to temporarily enable the IP SLAs responder functionality on a Cisco device in response to control messages from the source device.

```
ip sla responder
```

Additional References for IP SLAs UDP Jitter Operations

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

MIBs

MIBs	MIBs Link
• CISCO-DATA-COLLECTION-MIB • CISCO-RTTMON-MIB • IPV6-FLOW-LABEL-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

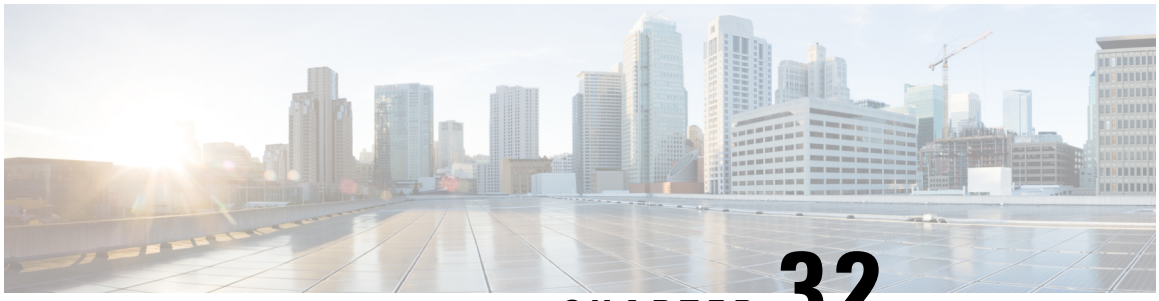
Feature Information for IP SLAs UDP Jitter Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 45: Feature Information for the IP SLAs UDP Jitter Operation

Feature Name	Releases	Feature Information
IP SLAs—UDP Jitter Operation	Cisco IOS XE Release 2.1 Cisco IOS XE 3.1.0SG Cisco IOS XE Release 3.2SE	The IP SLAs UDP jitter operation allows you to measure round-trip delay, one-way delay, one-way jitter, one-way packet loss, and connectivity in networks that carry UDP traffic.
IP SLAs for IPv6 (UDP Jitter, UDP Echo, ICMP Echo, TCP Connect)	Cisco IOS XE Release 2.1 Cisco IOS XE 3.1.0SG Cisco IOS XE Release 3.2SE	The IP SLAs for IPv6 (UDP Jitter, UDP Echo, ICMP Echo, TCP Connect) feature adds support for operability in IPv6 networks.
IP SLAs—Asymmetric Probe Support for UDP Jitter	Cisco IOS XE Release 3.10S	The IP SLAs—Asymmetric Probe Support for UDP Jitter feature supports the configuration of custom-defined packet sizes in response packets. The following command was introduced: **response-data-size**. In Cisco IOS XE Release 3.10S, support was added for the Cisco ASR 1000 Series Routers.



CHAPTER 32

IP SLAs Multicast Support

This module describes how to configure and schedule an IP Service Level Agreements (SLAs) multicast UDP jitter operation for measuring and reporting statistics such as one way latency, jitter, and packet loss for each multicast receiver in a user-specified multicast group. .

Prerequisites for IP SLAs Multicast Support

- Time synchronization, such as that provided by Network Time Protocol (NTP), is required between the source and the target device in order to provide accurate one-way delay (latency) measurements. To configure NTP on the source and target devices, perform the tasks in the "Performing Basic System Management" chapter of the *Network Management Configuration Guide*. Time synchronization is not required for the one-way jitter and packet loss measurements. However, if the time is not synchronized between the source and target devices, one-way jitter and packet loss data will be returned, but values of "0" will be returned for the one-way delay measurements provided by the UDP jitter operation.
- All devices must be part of the same VRF in order for IP SLAs multicast operations to succeed.
- The devices on which the responder and probe are to configured must both be running Cisco software images that support the IP SLAs Multicast Support feature. Before configuring any IP SLAs application, use the **show ip sla application** command to verify that the operation type is supported on your software image.

Restrictions for IP SLAs Multicast Support

The multicast UDP Jitter operation can provide only One Way (OW) data.

Information About IP SLAs Multicast Support

Multicast UDP Jitter Operations

A multicast UDP jitter operation measures and reports statistics, such as one way latency, jitter, and packet loss, for each multicast receiver in a user-specified multicast group. Multicast UDP jitter operations enable you to perform the following tasks:

- Analyze and evaluate the performance of a multicast network after deploying a new multicast network application or implementing new multicast-based protocols on the network.
- Check the network behavior for multicast before actually utilizing the multicast network for an important event.
- Take a proactive approach to monitoring a network to isolate possible problem areas.

The sender in a multicast UDP jitter operation sends UDP packets at a specified interval from the source device to a multicast IP address. During the initial configuration, a specified endpoint list provides a list of all the responders to be contacted for a given multicast operation. The multicast subsystem sends a unicast control packet to each of the multicast receivers in the endpoint list, utilizing the unicast path. A control message is sent to each receiver so that it can join the multicast group.

The IP SLAs multicast responder on the multicast receiver receives the UDP packets and records the time-stamp data.

A list of valid responders that have completed a successful IGMP join is maintained on the sender side. Once the responder list is received, multicast packet generation can proceed.

Because all multicast traffic is one way, from sender on the source to responder on the receiver, each responder that is part of the operation is responsible for performing local calculations and for storing the statistics. The statistics are sent back to the sender to be displayed at the end of each cycle of the operation (after all packets have been transmitted to the responder). Because the responder does not maintain a history of the statistics, and also releases all associated memory after sending the information to the sender, each scheduled operation (based on the frequency) is considered a new operation by the multicast responder, with no relationship to the previous one.

Multicast UDP jitter operations are supported in IPv4 networks.

How to Configure IP SLAs Multicast Support

Configuring the IP SLAs Responder on a Destination Device

Note A responder should not configure a permanent port for a sender. If the responder configures a permanent port for a sender, even if the packets are successfully sent (no timeout or packet-loss issues), the jitter value is zero.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla responder**
   - **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla responder**
	• **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
	Example:
	`Device(config)# ip sla responder`
	`Device(config)# ip sla responder udp-echo ipaddress 192.0.2.132 port 5000 vrf vrf1`
		(Optional) Temporarily enables IP SLAs responder functionality on a Cisco device in response to control messages from the source.
		(Optional; required only if protocol control is disabled on the source.) Enables IP SLAs responder functionality on the specified IP address, port and VRF.
		• Protocol control is enabled by default.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Creating a List of Multicast Responders on the Source Device

Before you begin

All responders to be added to the endpoint list (of responders) must first be configured on the destination device. For configuration information, see the "Configuring an IP SLAs Responder on the Destination Device" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla endpoint-list type ip** *template-name*
4. **description** *description*
5. **ip-address** *address* [*-address* | **,** *... ,* *address*] **port** *port*
6. **end**
7. **show ip sla endpoint-list** [**type ip** [*template-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla endpoint-list type ip** *template-name* **Example:** `Device(config)# ip sla endpoint-list type ip mcast-rcvrs`	Begins configuring an endpoint list and enters endpoint-list configuration mode.
Step 4	**description** *description* **Example:** `Device(config-epl)# description list of receivers`	(Optional) Adds descriptive text to the template being configured.
Step 5	**ip-address** *address* [*-address*	**,** *... ,* *address*] **port** *port* **Example:** `Device(config-epl)# ip-address 10.1.1.1-13 port 6500`	Adds the IPv4 or IPv6 address of a multicast responder to the endpoint list being configured. • Repeat this command until all desired addresses are configured. • Use the **no** from of this command to modify the endpoint list by removing one or more addresses.
Step 6	**end** **Example:** `Device(config-epl)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show ip sla endpoint-list** [**type ip** [*template-name*]] **Example:** `Device# show ip sla endpoint-list type ip mcast-rcvrs`	(Optional) Displays the configuration of the endpoint list.

Configuring Multicast UDP Jitter Operations

Note

- The IP SLAs UDP jitter operation does not support the IP SLAs History feature (statistics history buckets) because of the large data volume involved with UDP jitter operations. Therefore, the following commands are not supported for UDP jitter operations: **history buckets-kept**, **history filter**, **history lives-kept**, **samples-of-history-kept**, and **show ip sla history**.
- The MIB used by IP SLAs (CISCO-RTTMON-MIB) limits the hours-of-statistics kept for the UDP jitter operation to two hours. Configuring a larger value using the **history hours-of-statistics** *hours* global configuration change will not increase the value beyond two hours. However, the Data Collection MIB can be used to collect historical data for the operation. For information, see the CISCO-DATA-COLLECTION-MIB at http://www.cisco.com/go/mibs).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* **endpoint-list** *endpoint-list* [**ssm**] [**source-ip** *ip-address*] [**source-port** *port-number*] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]
5. **control retry** *retries*
6. **control timeout** *seconds*
7. **dscp** *dscp-value*
8. **tree-init** *number*
9. **history distributions-of-statistics-kept** *size*
10. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
11. **frequency** *seconds*
12. **history hours-of-statistics-kept** *hours*
13. **owner** *owner-id*
14. **request-data-size** *bytes*
15. **history statistics-distribution-interval** *milliseconds*
16. **tag** *text*
17. **threshold** *milliseconds*
18. **timeout** *milliseconds*
19. **tos** *number*
20. **verify-data**

21. **vrf** *vrf-name*
22. **end**
23. **show ip sla configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* **endpoint-list** *endpoint-list* [**ssm**] [**source-ip** *ip-address*] [**source-port** *port-number*] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Device(config-ip-sla)# udp-jitter 239.1.1.1 5000 endpoint-list mcast-rcvrs source-ip 10.10.10.106 source-port 7012 num-packets 50 interval 25`	Configures the IP SLAs operation as a multicast UDP jitter operation and enters multicast UDP jitter configuration mode.
Step 5	**control retry** *retries* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# control retry 2`	(Optional) Configures the number of times a sending device will resend a control protocol message.
Step 6	**control timeout** *seconds* **Example:** `Device(config-ip-sla-multicast-jitter)# control timeout 4`	(Optional) Configures the number of seconds that the destination device will wait for a control protocol message.
Step 7	**dscp** *dscp-value* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# dscp 10`	(Optional) Configures the DSCP value for the operation.

	Command or Action	Purpose
Step 8	**tree-init** *number* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# tree-init 1`	(Optional) Sets up the multicast tree.
Step 9	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 10	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-multicast-jitter-oper)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 11	**frequency** *seconds* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 12	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 13	**owner** *owner-id* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 14	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 15	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

	Command or Action	Purpose
Step 16	**tag** *text* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 17	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 18	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 19	**tos** *number* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# tos 160`	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
Step 20	**verify-data** **Example:** `Device(config-ip-sla-multicast-jitter-oper)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 21	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-multicast-jitter-oper)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) VPNs using IP SLAs operations.
Step 22	**end** **Example:** `Device(config-ip-sla-multicast-jitter-oper)# end`	Returns to privileged EXEC mode.
Step 23	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range*	**schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life**	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose
	{**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]	

	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Configuration Examples for IP SLAs Multicast Support

Example: Multicast UDP Jitter Operation

```
Device# show ip sla endpoint-list

Endpoint-list Name: multicast
    Description:
    ip-address 192.0.2.1 port 1111
    ip-address 192.0.2.2 port 2222
    ip-address 192.0.2.3 port 3333
```

```
Device# show ip sla configuration 22

IP SLAs Infrastructure Engine-III
Entry number: 22
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: udp-jitter
Target address/Source address: 224.1.1.1/0.0.0.0
Target port/Source port: 2460/0
Type Of Service parameter: 0x0
Request size (ARR data portion): 32
Packet Interval (milliseconds)/Number of packets: 20/10
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:

  sno    oper-id             dest-ip-addr  !<---Responders in endpoint list: multicast
    1   976271337             192.0.2.1
    2  1632881300             192.0.2.2
    3  2138021658             192.0.2.3
```

Additional References for IP SLAs Multicast Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP SLAs commands	Cisco IOS IP SLAs Command Reference
Information about Cisco IP SLAs	"Cisco IOS IP SLAs Overview" module of the *IP SLAs Configuration Guide*

MIBs

MIB	MIBs Link
CISCO-IPSLA-TC-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

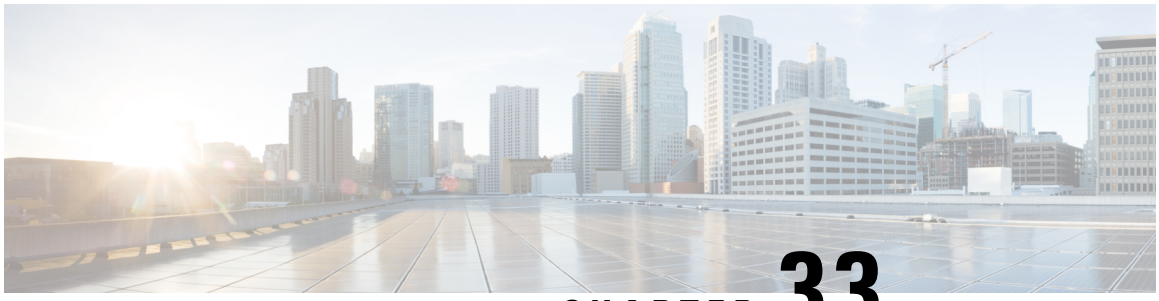
Feature Information for IPSLA Multicast Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 46: Feature Information for IPSLA Multicast Support

Feature Name	Releases	Feature Information
IPSLA Multicast Support	15.2(4)M 15.3(1)S Cisco IOS XE Release 3.8S 15.1(2)SG Cisco IOS XE Release 3.4SG	This feature introduced the multicast UDP jitter operation for measuring and reporting statistics such as one way latency, jitter, and packet loss for each multicast receiver in a user-specified multicast group. The following commands were introduced or modified: **clock-tolerance ntp oneway**, **control (IP SLA)**, **dscp (IP SLA)**, **history distributions-of-statistics-kept**, **history enhanced**, **history hours-of-statistics-kept**, **ip-address (endpoint list)**, **operation-packet priority**, **owner**, **precision**, **show ip sla application**, **show ip sla configuration**, **show ip sla endpoint-list**, **show ip sla statistics**, **show ip sla statistics aggregated**, **tag (IP SLA)**, **timeout (IP SLA)**, **tos**, **tree-init**, **udp-jitter**, **verify-data (IP SLA)**, **vrf**.



CHAPTER 33

Configuring IP SLAs UDP Jitter Operations for VoIP

This document describes how to configure an IP Service Level Agreements (SLAs) User Datagram Protocol (UDP) jitter operation to proactively monitor Voice over IP (VoIP) quality levels in your network, allowing you to guarantee VoIP quality levels to your users in IPv4 or IPv6 networks. The IP SLAs VoIP UDP jitter operation accurately simulates VoIP traffic using common codecs and calculates consistent voice quality scores (MOS and ICPIF) between Cisco devices in the network.

Note The term "Voice" in this document should be taken to mean any Internet telephony applications. The term "Voice over IP" can include the transmission of multimedia (both voice and video) over IP networks.

Restrictions for IP SLAs UDP Jitter Operations for VoIP

- This feature uses UDP traffic to generate approximate Voice over IP scores. It does not provide support for the Real-Time Transport Protocol (RTP).
- ICPIF and MOS values provided by this feature, while consistent within IP SLAs, are estimates only and are intended only for relative comparisons. The values may not match values determined using other methods.
- Predictions of customer opinion (such as those listed for the E-Model transmission rating factor R and derived Mean Opinion Scores) determined by any method are intended only for transmission planning and analysis purposes and should not be interpreted as reflecting actual customer opinions.

Information About IP SLAs UDP Jitter Operations for VoIP

The Calculated Planning Impairment Factor (ICPIF)

The ICPIF originated in the 1996 version of ITU-T recommendation G.113, "Transmission impairments," as part of the formula *Icpif = Itot - A.* ICPIF is actually an acronym for "(Impairment) Calculated Planning Impairment Factor," but should be taken to simply mean the "calculated planning impairment factor." The ICPIF attempts to quantify, for comparison and planning purposes, the key impairments to voice quality that are encountered in the network.

The ICPIF is the sum of measured impairment factors (total impairments, or *Itot*) minus a user-defined access Advantage Factor (*A*) that is intended to represent the user's expectations, based on how the call was placed (for example, a mobile call versus a land-line call). In its expanded form, the full formula is expressed as:

Icpif = Io + Iq + Idte + Idd + Ie - A

where

- *Io* represents impairments caused by non-optimal loudness rating,
- *Iq* represents impairments caused by PCM quantizing distortion,
- *Idte* represents impairments caused by talker echo,
- *Idd* represents impairments caused by one-way transmission times (one-way delay),
- *Ie* represents impairments caused by equipment effects, such as the type of codec used for the call and packet loss, and
- *A* represents an access Advantage Factor (also called the user Expectation Factor) that compensates for the fact that users may accept some degradation in quality in return for ease of access.

ICPIF values are expressed in a typical range of 5 (very low impairment) to 55 (very high impairment). ICPIF values numerically less than 20 are generally considered "adequate." While intended to be an objective measure of voice quality, the ICPIF value is also used to predict the subjective effect of combinations of impairments. The table below, taken from G.113 (02/96), shows how sample ICPIF values are expected to correspond to subjective quality judgement.

Table 47: Quality Levels as a Function of Total Impairment Factor ICPIF

Upper Limit for ICPIF	Speech Communication Quality
5	Very good
10	Good
20	Adequate
30	Limiting case
45	Exceptional limiting case
55	Customers likely to react strongly (complaints, change of network operator)

For further details on the ICPIF, see the 1996 version of the G.113 specification.

Note The latest version of the ITU-T G.113 Recommendation (2001), no longer includes the ICPIF model. Instead, it refers implementers to G.107: "The Impairment Factor method, used by the E-model of ITU-T G.107, is now recommended. The earlier method that used Quantization Distortion Units is no longer recommended." The full E-Model (also called the ITU-T Transmission Rating Model), expressed as $R = Ro - Is - Id - Ie + A$, provides the potential for more accurate measurements of call quality by refining the definitions of impairment factors (see the 2003 version of the G.107 for details). Though the ICPIF shares terms for impairments with the E-Model, the two models should not be confused. The IP SLAs VoIP UDP Operation feature takes advantage of observed correspondences between the ICPIF, transmission rating factor R, and MOS values, but does not yet support the E-Model.

IP SLAs uses a simplified ICPIF formula, defined in more detail later in this document.

Mean Opinion Scores (MOS)

The quality of transmitted speech is a subjective response of the listener. Each codec used for transmission of Voice over IP provides a certain level of quality. A common benchmark used to determine the quality of sound produced by specific codecs is MOS. With MOS, a wide range of listeners have judged the quality of voice samples sent using particular codecs, on a scale of 1 (poor quality) to 5 (excellent quality). The opinion scores are averaged to provide the mean for each sample. The table below shows MOS ratings and the corresponding description of quality for each value.

Table 48: MOS Ratings

Score	Quality	Description of Quality Impairment
5	Excellent	Imperceptible
4	Good	Just perceptible, but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying but not objectionable
1	Bad	Very annoying and objectionable

As the MOS ratings for codecs and other transmission impairments are known, an estimated MOS can be computed and displayed based on measured impairments. This estimated value is designated as MOS-CQE (Mean Opinion Score; Conversational Quality, Estimated) by the ITU in order to distinguish it from objective or subjective MOS values (see *P.800.1 Mean Opinion Score (MOS) terminology - ITU* for details).

Voice Performance Monitoring Using IP SLAs

One of the key metrics in measuring voice and video quality over an IP network is jitter. Jitter is the name used to indicate the variation in delay between arriving packets (inter-packet delay variance). Jitter affects voice quality by causing uneven gaps in the speech pattern of the person talking. Other key performance parameters for voice and video transmission over IP networks include latency (delay) and packet loss. IP SLAs is an embedded active monitoring feature of Cisco software that provides a means for simulating and

measuring these parameters in order to ensure your network is meeting or exceeding service-level agreements with your users.

IP SLAs provides a UDP jitter operation, which consists of UDP probe packets sent across the network from an origin device to a specific destination (called the operational target). This synthetic traffic is used to record the amount of jitter for the connection, as well as the round-trip time, per-direction packet loss, and one-way delay time (one-way latency). The term “synthetic traffic” indicates that the network traffic is simulated; that is, the traffic is generated by IP SLAs. Data, in the form of collected statistics, can be displayed for multiple tests over a user-defined period of time, allowing you to see, for example, how the network performs at different times of the day, or over the course of a week. The jitter probe has the advantage of utilizing the IP SLAs Responder to provide minimal latency at the receiving end.

The IP SLAs VoIP UDP jitter operation modifies the standard UDP jitter operation by adding the capability to return MOS and ICPIF scores in the data collected by the operation, in addition to the metrics already gathered by the UDP jitter operation. This VoIP-specific implementation provides even more useful information in determining the performance of your VoIP network, thereby improving your ability to perform network assessment, troubleshooting, and health monitoring.

Codec Simulation Within IP SLAs

The IP SLAs VoIP UDP jitter operation computes statistics by sending n UDP packets, each of size s, sent t milliseconds apart, from a given source device to a given target device, at a given frequency f. The target device must be running the Cisco IP SLAs Responder in order to process the probe operations.

To generate MOS and ICPIF scores, you must specify the codec type used for the connection when configuring the VoIP UDP jitter operation. Based on the type of codec you configure for the operation, the number of packets (n), the size of each payload (s), the inter-packet time interval (t), and the operational frequency (f) will be auto-configured with default values. However, you are given the option, if needed, to manually configure these parameters in the syntax of the**udp-jitter** command.

The table below shows the default parameters that are configured for the operation by codec.

Table 49: Default VoIP UDP Jitter Operation Parameters by Codec

Codec	Default Request Size (Packet Payload) (s)	Default Interval Between Packets (t)	Default Number of Packets (n)	Frequency of Probe Operations (f)
G.711 mu-Law (g711ulaw)	160 + 12 RTP bytes	20 ms	1000	Once every 1 minute
G.711 A-Law (g711alaw)	160 + 12 RTP bytes	20 ms	1000	Once every 1 minute
G.729A (g729a)	20 + 12 RTP bytes	20 ms	1000	Once every 1 minute

For example, if you configure the VoIP UDP jitter operation to use the characteristics for the g711ulaw codec, by default a probe operation will be sent once a minute (**f**). Each probe operation would consist of 1000 packets (**n**), with each packet containing 180 bytes of synthetic data (**s**), sent 20 milliseconds apart (**t**).

The IP SLAs ICPIF Value

ICPIF value computation with Cisco software is based primarily on the two main factors that can impair voice quality: delayed packets and lost packets. Because packet delay and packet loss can be measured by IP SLAs, the full ICPIF formula, *Icpif = Io + Iq + Idte + Idd + Ie - A*, is simplified by assuming the values of *Io* , *Iq* , and *Idte* are zero, resulting in the following formula:

Total Impairment Factor (Icpif) = Delay Impairment Factor (Idd) + Equipment Impairment Factor (Ie) - Expectation/Advantage Factor (A)

This means that the ICPIF value is computed by adding a Delay Impairment Factor, which is based on a measurement of delayed packets, and an Equipment Impairment Factor, which is based on a measurement of lost packets. From this sum of the total impairments measured in the network, an impairment variable (the Expectation Factor) is subtracted to yield the ICPIF.

This is the same formula used by Cisco Gateways to calculate the ICPIF for received VoIP data streams.

The Delay Impairment Factor

The Delay Impairment Factor (*Idd*) is a number based on two values. One value is fixed and is derived using the static values (as defined in the ITU standards) for Codec Delay, Look Ahead Delay, and Digital Signal Processing (DSP) Delay. The second value is variable and is based on the measured one-way delay (round-trip time measurement divided by 2). The one-way delay value is mapped to a number using a mapping table that is based on a G.107 (2002 version) analytic expression. The table below shows sample correspondences between the one-way delay measured by IP SLAs and Delay Impairment Factor values.

Table 50: Sample Correspondence of One-Way Delay to ICPIF Delay Impairment

One-Way Delay (ms)	Delay Impairment Factor
50	1
100	2
150	4
200	7

The Equipment Impairment Factor

The Equipment Impairment Factor (*Ie*) is a number based on the amount of measured packet loss. The amount of measured packet loss, expressed as a percentage of total number of packets sent, corresponds an Equipment Impairment Factor that is defined by codec. The table below shows sample correspondences between the packet loss measured by IP SLAs and Equipment Impairment Factor values.

Table 51: Sample Correspondence of Measured Packet Loss to ICPIF Equipment Impairment

Packet Loss (as a percentage of total number of packets sent)	Equipment Impairment Value for PCM (G.711) Codecs	Equipment Impairment Value for the CS-ACELP (G.729A) Codec
2%	12	20
4%	22	30

Packet Loss (as a percentage of total number of packets sent)	Equipment Impairment Value for PCM (G.711) Codecs	Equipment Impairment Value for the CS-ACELP (G.729A) Codec
6%	28	38
8%	32	42

The Expectation Factor

The Expectation Factor, also called the Advantage Factor (*A*), is intended to represent the fact that users may accept some degradation in quality in return for ease of access. For example, a mobile phone user in a hard-to-reach location may have an expectation that the connection quality will not be as good as a traditional land-line connection. This variable is also called the Advantage Factor (short for Access Advantage Factor) because it attempts to balance an increased access advantage against a decline in voice quality.

The table below, adapted from ITU-T Rec. G.113, defines a set of provisional maximum values for *A* in terms of the service provided.

Table 52: Advantage Factor Recommended Maximum Values

Communication Service	Advantage / Expectation Factor: Maximum value of A
Conventional wire-line (land-line)	0
Mobility (cellular connections) within a building	5
Mobility within a Geographical area or moving in a vehicle	10
Access to hard-to-reach location; (for example, via multi-hop satellite connections)	20

These values are only suggestions. To be meaningful, the use of the factor *A* and its selected value in a specific application should be used consistently in any planning model you adopt. However, the values in the table above should be considered as the absolute upper limits for *A* .

The default Advantage Factor for IP SLAs VoIP UDP jitter operations is always zero.

The IP SLAs MOS Value

IP SLAs uses an observed correspondence between ICPIF and MOS values to estimate an MOS value. Usage of the abbreviation MOS within the context of this feature should be taken to represent the MOS-CQE (Mean Opinion Score; Conversational Quality, Estimated).

The E model, as defined in G.107 (03/2003), predicts the subjective quality that is experienced by an average listener by combining the impairment caused by transmission parameters (such as loss and delay) into a single rating, the transmission rating factor R (the R Factor). This rating, expressed in a scale of 0 (worst) to 100 (best) can be used to predict subjective user reactions, such as the MOS. Specifically, the MOS can be obtained from the R Factor with a converting formula. Conversely, a modified inverted form can be used to calculate R Factors from MOS values.

There is also a relationship between the ICPIF value and the R Factor. IP SLAs takes advantage of this correspondence by deriving the approximate MOS score from an estimated R Factor, which, in turn, is derived

from the ICPIF score. The table below shows the resulting MOS values that will be generated for corresponding ICPIF values.

Table 53: Correspondence of ICPIF Values to MOS Values

ICPIF Range	MOS	Quality Category
0 - 3	5	Best
4 - 13	4	High
14 - 23	3	Medium
24 - 33	2	Low
34 - 43	1	Poor

IP SLAs will always express the estimated MOS value as a number in the range of 1 to 5, with 5 being the best quality. A MOS value of 0 (zero) indicates that MOS data could not be generated for the operation.

How to Configure IP SLAs UDP Jitter Operations for VoIP

Configuring the IP SLAs Responder on a Destination Device

Note A responder should not configure a permanent port for a sender. If the responder configures a permanent port for a sender, even if the packets are successfully sent (no timeout or packet-loss issues), the jitter value is zero.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla responder**
   - **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla responder**
	• **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
	Example:
	`Device(config)# ip sla responder`
	`Device(config)# ip sla responder udp-echo ipaddress 192.0.2.132 port 5000 vrf vrf1`
		(Optional) Temporarily enables IP SLAs responder functionality on a Cisco device in response to control messages from the source.
		(Optional; required only if protocol control is disabled on the source.) Enables IP SLAs responder functionality on the specified IP address, port and VRF.
		• Protocol control is enabled by default.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring and Scheduling an IP SLAs VoIP UDP Jitter Operation

Note

- Currently, IP SLAs supports only the following speech codecs (compression methods):
  - G.711 A Law (g711alaw: 64 kbps PCM compression method)
  - G.711 mu Law (g711ulaw: 64 kbps PCM compression method)
  - G.729A (g729a: 8 kbps CS-ACELP compression method)
- The following commands, available in UDP jitter configuration mode, are not valid for UDP jitter (codec) operations:
  - **history distributions-of-statistics-kept**
  - **history statistics-distribution-interval**
  - **request-data-size**
- Specifying the codec-type will configure the appropriate default values for the **codec-interval**, **codec-size**, and **codec-numpacket** options. You should not specify values for the interval, size, and number of packet options unless you have a specific reason to override the defaults (for example, approximating a different codec).
- The **show ip sla configuration** command will list the values for the "Number of statistic distribution buckets kept" and "Statistic distribution interval (milliseconds)," but these values do not apply to jitter (codec) operations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* **codec** *codec-type* [**codec-numpackets** *number-of-packets*] [**codec-size** *number-of-bytes*] [**codec-interval** *milliseconds*] [**advantage-factor** *value*] [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}]
5. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
6. **frequency** *seconds*
7. **history hours-of-statistics-kept** *hours*
8. **owner** *owner-id*
9. **tag** *text*
10. **threshold** *milliseconds*
11. **timeout** *milliseconds*
12. Do one of the following:
    - **tos** *number*
    - **traffic-class** *number*
13. **flow-label** *number*
14. **verify-data**
15. **vrf** *vrf-name*
16. **end**
17. **show ip sla configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* **codec** *codec-type* [**codec-numpackets** *number-of-packets*] [**codec-size** *number-of-bytes*] [**codec-interval** *milliseconds*] [**advantage-factor** *value*]	Configures the operation as a jitter (codec) operation that will generate VoIP scores in addition to latency, jitter, and packet loss statistics.

	Command or Action	Purpose
	[**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `Device(config-ip-sla)# udp-jitter 209.165.200.225 16384 codec g711alaw advantage-factor 10`	
Step 5	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-jitter)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 6	**frequency** *seconds* **Example:** `Device(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 7	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-jitter)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 8	**owner** *owner-id* **Example:** `Device(config-ip-sla-jitter)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 9	**tag** *text* **Example:** `Device(config-ip-sla-jitter)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 10	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-jitter)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 11	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-jitter)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 12	Do one of the following: • **tos** *number*	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.

	Command or Action	Purpose

	• **traffic-class** *number*
	Example:
	`Device(config-ip-sla-jitter)# tos 160`
	Example:
	`Device(config-ip-sla-jitter)# traffic-class 160`
		or
		(Optional) In an IPv6 network only, defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

Step 13	**flow-label** *number* **Example:** `Device(config-ip-sla-jitter)# flow-label 112233`	(Optional) In an IPv6 network only, defines the flow label field in the IPv6 header for a supported IP SLAs operation.
Step 14	**verify-data** **Example:** `Device(config-ip-sla-jitter)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 15	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-jitter)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 16	**end** **Example:** `Device(config-ip-sla-jitter)# end`	Returns to privileged EXEC mode.
Step 17	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range*	**schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever**	*seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm* [:*ss*]}] **Example:** `Device(config)# ip sla schedule 10 life forever start-time now` `Device(config)# ip sla group schedule 10 schedule-period frequency` `Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now` `Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Configuration Examples for IP SLAs UDP Jitter Operations for VoIP

Example IP SLAs VoIP UDP Operation Configuration

The following example assumes that the Cisco IP SLAs Responder is enabled on the device at 209.165.200.225.

```
Device> enable

Password:
Device# configure terminal

Enter configuration commands, one per line.  End with the end command.
Device(config)# ip sla 10
```

```
Device(config-sla)# udp-jitter 209.165.200.225 16384 codec g711alaw advantage-factor 2

Device(config-sla-jitter)# owner admin_bofh
Device(config-sla-jitter)# exit

Device(config)# ip sla schedule 10 start-time now

Device(config)# exit

Device#
Device# show running-config | begin ip sla 10

ip sla 10
 udp-jitter 209.165.200.225 16384 codec g711alaw advantage-factor 2
 owner admin_bofh
ip sla schedule 10 start-time now
 .
 .
 .
Device# show ip sla configuration 10

Entry number: 10
Owner: admin_bofh
Tag:
Type of operation to perform: jitter
Target address: 209.165.200.225
Source address: 0.0.0.0
Target port: 16384
Source port: 0
Operation timeout (milliseconds): 5000
Codec Type: g711alaw
Codec Number Of Packets: 1000
Codec Packet Size: 172
Codec Interval (milliseconds): 20
Advantage Factor: 2
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Operation frequency (seconds): 60
Next Scheduled Start Time: Start Time already passed
Life (seconds): 3600
Entry Ageout (seconds): never
Status of entry (SNMP RowStatus): Active
Connection loss reaction enabled: No
Timeout reaction enabled: No
Verify error enabled: No
Threshold reaction type: Never
Threshold (milliseconds): 5000
Threshold Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Reaction Type: None
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
Enhanced History:
```

When a codec type is configured for a jitter operation, the standard jitter “Request size (ARR data portion),” “Number of packets,” and “Interval (milliseconds)” parameters will not be displayed in the **show ip sla configuration** command output. Instead, values for “Codec Packet Size,” “Codec Number of Packets,” and “Codec Interval (milliseconds)” are displayed.

Example IP SLAs VoIP UDP Operation Statistics Output

Use the **show ip sla statistics** command to display Voice scores (ICPIF and MOS values) for the jitter (codec) operation.

```
Device# show ip sla statistics 10

Entry number: 10
Modification time: 12:57:45.690 UTC Sun Oct 26 2003
Number of operations attempted: 1
Number of operations skipped: 0
Current seconds left in Life: Forever
Operational state of entry: Active
Last time this entry was reset: Never
Connection loss occurred: FALSE
Timeout occurred: FALSE
Over thresholds occurred: FALSE
Latest RTT (milliseconds): 19
Latest operation start time: 12:57:45.723 Sun Oct 26 2003
Latest operation return code: OK
!
Voice Scores:
ICPIF:  20          MOS Score:  3.20
!
RTT Values:
NumOfRTT: 10     RTTAvg: 19      RTTMin: 19     RTTMax: 20
RTTSum:  191    RTTSum2: 3649
Packet Loss Values:
PacketLossSD: 0 PacketLossDS: 0
PacketOutOfSequence: 0  PacketMIA: 0    PacketLateArrival: 0
InternalError: 0        Busies: 0
Jitter Values:
NumOfJitterSamples: 9
MinOfPositivesSD: 0     MaxOfPositivesSD: 0
NumOfPositivesSD: 0     SumOfPositivesSD: 0     Sum2PositivesSD: 0
MinOfNegativesSD: 0     MaxOfNegativesSD: 0
NumOfNegativesSD: 0     SumOfNegativesSD: 0     Sum2NegativesSD: 0
MinOfPositivesDS: 1     MaxOfPositivesDS: 1
NumOfPositivesDS: 1     SumOfPositivesDS: 1     Sum2PositivesDS: 1
MinOfNegativesDS: 1     MaxOfNegativesDS: 1
NumOfNegativesDS: 1     SumOfNegativesDS: 1     Sum2NegativesDS: 1
Interarrival jitterout: 0       Interarrival jitterin: 0
One Way Values:
NumOfOW: 0
OWMinSD: 0      OWMaxSD: 0      OWSumSD: 0      OWSum2SD: 0
OWMinDS: 0      OWMaxDS: 0      OWSumDS: 0      OWSum2DS: 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Related Topic	Document Title
Voice over IP (VoIP) codecs	Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation
Jitter in Packet Voice Networks	Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms) shtml

Standards and RFCs

Standard[8]/RFC[9]	Title
ITU-T Recommendation G.107 (2003)	The E-model, a computation model for use in transmission planning
ITU-T Recommendation G.113 (1996)	*Transmission impairments*
ITU-T Recommendation G.113 (2001)	Transmission impairments due to speech processing
ITU-T Recommendation G.711 (1998)	*Pulse code modulation (PCM) of voice frequencies* (also known as the G.711 Voice Codec)
ITU-T Recommendation G.729 Annex A (1996)	*Reduced complexity 8 kbit/s CS-ACELP speech codec* (also known as the G.729/A/B Speech Codec)
ITU-T Recommendation P.800.1 (2003)	Mean Opinion Score (MOS) terminology
RFC 768	*User Datagram Protocol*
RFC 1889	*RTP: A Transport Protocol for Real-Time Applications*

[8] Full support by this feature for listed RFCs is not claimed. ITU Telecommunication Standards ("ITU-T Recommendations In Force") can be obtained from http://www.itu.ch. Summary definitions are available from a variety of internet sources.

[9] Full support by this feature for listed RFCs is not claimed.

MIBs

MIB	MIB Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP SLAs VoIP UDP Jitter Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 54: Feature Information for the IP SLAs VoIP UDP Jitter Operation

Feature Name	Releases	Feature Information
IP SLAs - UDP Based VoIP Operation		The IP SLAs User Datagram Protocol (UDP) jitter operation allows you to measure round-trip delay, one-way delay, one-way jitter, one-way packet loss, and connectivity in networks that carry UDP traffic.
IP SLAs for IPv6 (UDP Jitter, UDP Echo, ICMP Echo, TCP Connect)		Support was added for operability in IPv6 networks.

Glossary

codec --In the context of IP Telephony, a codec is a compression and decompression algorithm used to transfer voice and video data more efficiently. Voice codec types are typically referred to using the ITU recommendation number that defines the algorithm (for example, “G.711” instead of “PCM”).

CS-ACELP --The codec type defined in the reference documents G.729 and G.729A, *Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)* .

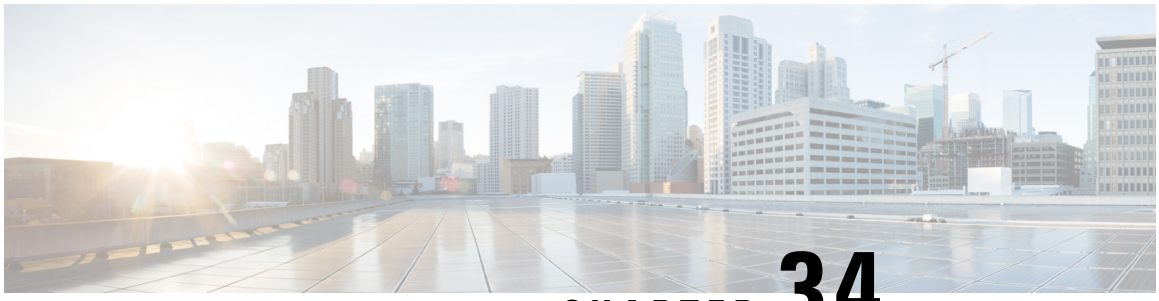
ITU --The International Telecommunication Union. The ITU is an international organization within the United Nations System where governments and the private sector coordinate global telecom networks and services. The ITU Telecommunication Standardization Sector (ITU-T), responsible for defining standards (Recommendations) covering all fields of telecommunications, is one of the three operational sectors of the ITU. The ITU web site is at http://www.itu.int.

ITU-T --ITU Telecommunication Standardization Sector. The ITU-T is one of the three operational sectors of the ITU, and is responsible for defining standards (called ITU-T Recommendations) covering all fields of telecommunications.

MOS-CQE (Mean Opinion Score; Conversational Quality, Estimated)--The score calculated by a network planning model which aims at predicting the quality in a conversational application situation. Estimates of conversational quality carried out according to ITU-T Rec. G.107, when transformed to a mean opinion score (MOS), give results in terms of MOS-CQE.[10]

PCM --The codec type defined in the reference document G.711, *Pulse code modulation (PCM) of voice frequencies* .

[10] Definition from ITU-T Recommendation P.800.1. Used in accordance with the ITU Copyright and Disclaimer Notice.



CHAPTER 34

IP SLAs QFP Time Stamping

This module describes how to configure the IP SLA QFP Time Stamping feature for IP Service Level Agreements (SLAs) UDP jitter operations. This new probe and responder structure enables more accurate network performance measurements.

Prerequisites for IP SLAs QFP Time Stamping

- The devices on which the responder and probe are to configured must both be running Cisco software images that support QFP time stamping in order for the IP SLAs QFP Time Stamping feature to work.
- Time synchronization, such as that provided by NTP, is required between the source and the target device in order to provide accurate one-way delay (latency) measurements. To configure NTP on the source and target devices, perform the tasks in the "Performing Basic System Management" chapter of the *Network Management Configuration Guide*.
- Before configuring any IP SLAs application, you can use the **show ip sla application** command to verify that the operation type is supported on your software image.

Restrictions for IP SLA QFP Time Stamping

- After rebooting the sender or responder devices, the Forward Processor (FP) and Route Processor (RP) times can be inaccurate until SNTP synchronizes the FP clock to the RP clock. To avoid running an operation before the device FP and RP times are stable, wait several minutes after a reboot before starting the UDP jitter operation.
- The one way delay value reported by an IP SLAs UDP jitter operation are dependent on the NTP synchronization level. Even if the device is synchronized, if the NTP offset values on the device are large, then one way values can be inaccurate. In cases where offset value becomes too large, the one way

value may not be reported. Also, the NTP offset value on the device can fluctuate and these changes will be reflected in one way values reported.

- If you configure the optimized time stamp location on the source device and the device on which the targeted IP SLAs Responder is configured does not support the optimized time stamp location, the IP SLAs operation will fail.
- IP SLAs QFP Time Stamping is not supported on the Cisco CSR 1000v or Cisco ISRv.

Information About IP SLAs QFP Time Stamping

IP SLAs UDP Jitter Operation

The IP Service Level Agreements (SLAs) UDP jitter operation diagnoses network suitability for real-time traffic applications such as VoIP, video over IP, or real-time conferencing.

Jitter means inter-packet delay variance. When multiple packets are sent consecutively from a source to a destination, for example, 10 ms apart, and if the network is behaving ideally, the destination should receive the packets 10 ms apart. But if there are delays in the network (like queuing, arriving through alternate routes, and so on) the arrival delay between packets might be greater than or less than 10 ms. Using this example, a positive jitter value indicates that packets arrived greater than 10 ms apart. If packets arrive 12 ms apart, then positive jitter is 2 ms; if packets arrive 8 ms apart, negative jitter is 2 ms. For delay-sensitive networks like VoIP, positive jitter values are undesirable, and a jitter value of 0 is ideal.

However, the IP SLAs UDP jitter operation does more than just monitor jitter. As the UDP jitter operation includes data returned by the IP SLAs UDP operation, the UDP jitter operation can be used as a multipurpose data gathering operation. The packets that IP SLAs generate carry packet-sending and receiving sequence information, and sending and receiving time stamps from the source and the operational target. Based on this information, UDP jitter operations are capable of measuring the following:

- Per-direction jitter (source to destination and destination to source)
- Per-direction packet loss
- Per-direction delay (one-way delay)
- Round-trip delay (average round-trip time)

As paths for sending and receiving data may be different (asymmetric), the per-direction data allows you to more readily identify where congestion or other problems are occurring in the network.

The UDP jitter operation functions by generating synthetic (simulated) UDP traffic. Asymmetric probes support custom-defined packet sizes per direction with which different packet sizes can be sent in request packets (from the source device to the destination device) and in response packets (from the destination device to the source device).

The UDP jitter operation sends N number of UDP packets, each of size S, T milliseconds apart, from a source device to a destination device, at a given frequency of F. In response, UDP packets of size P is sent from the destination device to the source device. By default, ten packet frames (N), each with a payload size of 10 bytes (S), are generated every 10 ms (T), and the operation is repeated every 60 seconds (F). Each of these parameters is user-configurable, so as to best simulate the IP service that you provide, as shown in the table below.

Table 55: UDP Jitter Operation Parameters

UDP Jitter Operation Parameter	Default	Configuration Commands
Number of packets (N)	10 packets	**udp-jitter num-packets**
Payload size per request packet (S)	10 bytes	**request-data-size**
Payload size per response packet (P)	The default response data size varies depending on the type of IP SLAs operation configured. **Note** If the **response-data-size** command is not configured, then the response data size value is the same as the request data size value.	**response-data-size**
Time between packets, in milliseconds (T)	10 ms	**udp-jitter interval**
Elapsed time before the operation repeats, in seconds (F)	60 seconds	**frequency (IP SLA)**

The IP SLAs operations function by generating synthetic (simulated) network traffic. A single IP SLAs operation (for example, IP SLAs operation 10) repeats at a given frequency for the lifetime of the operation.

QFP Time Stamping

IP SLAs UDP jitter is the most widely-used IP SLAs operation for measuring metrics such as round-trip time, one-way delay, jitter, and packet loss. The accuracy of measurements depends on the location where the time stamps are taken while the packet moves from the sender to responder, and back.

Typically, time stamps for IP SLAs operations are taken in the IP SLAs process at the Route Processor (RP). This time-stamp location results in inaccurate and inconsistent measurements because the time stamps are subject to scheduling delays experienced at the RP. QFP time stamping moves the location of the time stamping from the RP to the Cisco Packet Processor (CPP).

However, to measure the one-way delay, the clocks on the source and target devices must be synchronized. Because device CPP clocks cannot be synchronized directly to an external clock source, the RP clocks are synchronized with an external clock source and SNTP is used to synchronize RP and Forwarding Processor (FP) clocks. The accuracy of the RP-FP synchronization is poor. To address this issue, the enhanced UDP jitter probe in the QFP Time Stamping feature stores both the RP and CPP time stamps. RTT and jitter calculations utilize the CPP time stamps, and one-way calculations continue to be based on RP time stamping. Therefore, time synchronization, such as that provided by NTP, is required between the source and the target device in order to provide accurate one-way delay (latency) measurements. One-way latency values are computed using RP time stamps are corrected by applying estimated-correction algorithms based on CPP time stamps.

QFP time stamping includes an enhanced UDP probe and enhanced responder. The devices on which the UDP probe and IP SLAs responder are configured must both be running Cisco software images that support QFP time stamping and the optimized time stamp location (for more accurate RTT measurements). If the UDP jitter operation is targeted to an responder on a device that does not support the optimized time stamp location, the IP SLAs probe will fail.

How to Configure IP SLAs QFP Time Stamping

Configuring the IP SLAs Responder on the Destination Device

Note A responder should not configure a permanent port for the same sender. If the responder configures a permanent port for the same sender, even if the packets are successfully sent (no timeout or packet-loss issues), the jitter values will be zero.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip sla responder**
   - **ip sla responder udp-echo ipaddress** *ip-address* **port** *port*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ip sla responder**
	• **ip sla responder udp-echo ipaddress** *ip-address* **port** *port*
	Example:
	`Device(config)# ip sla responder`
	Example:
	`Device(config)# ip sla responder udp-echo ipaddress 172.29.139.132 port 5000`
		(Optional) Temporarily enables IP SLAs Responder functionality on a Cisco device in response to control messages from the source.
		(Optional) Required only if protocol control is disabled on the source. Enables IP SLAs responder functionality on the specified IP address and port.
		• Protocol control is enabled by default.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring and Scheduling a UDP Jitter Operation on a Source Device

Perform only one of the following tasks:

- Configuring a Basic UDP Jitter Operation on a Source Device
- Configuring a UDP Jitter Operation with Additional Characteristics

Configuring a Basic UDP Jitter Operation with QFP Time Stamping

Perform this task to configure a UDP jitter probe with QFP time stamping on the source device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]
5. **frequency** *seconds*
6. **precision microseconds**
7. **optimize timestamp**
8. **end**
9. **show ip sla configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:**	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.

	Command or Action	Purpose
	`Device(config)# ip sla 10`	
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Device(config-ip-sla)# udp-jitter 172.29.139.134 5000`	Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration submode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and destination devices.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**precision microseconds** **Example:** `Device(config-ip-sla-jitter)# precision microseconds`	Enables QFP time stamping.
Step 7	**optimize timestamp** **Example:** `Device(config-ip-sla-jitter)# optimize timestamp`	(Optional) For Cisco ASR 1000 Series routers only. Enables CPP ticks which is more accurate than cpp UNIX time. **Note** If the Responder does not support cpp ticks, the IP SLAs operation will fail.
Step 8	**end** **Example:** `Device(config-ip-sla-jitter)# end`	Returns to privileged EXEC mode.
Step 9	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Configuring a UPD Jitter Operation with QFP Time Stamping and Additional Characteristics

Note

- The IP SLAs UDP jitter operation does not support the IP SLAs History feature (statistics history buckets) because of the large data volume involved with UDP jitter operations. This means that the following commands are not supported for UDP jitter operations: **history buckets-kept**, **history filter**, **history lives-kept**, **samples-of-history-kept**, and **show ip sla history**.
- The MIB used by IP SLAs (CISCO-RTTMON-MIB) limits the hours-of-statistics kept for the UDP jitter operation to two hours. Configuring a larger value using the **history hours-of-statistics** *hours* global configuration change will not increase the value beyond two hours. However, the Data Collection MIB can be used to collect historical data for the operation. For information, see the CISCO-DATA-COLLECTION-MIB at http://www.cisco.com/go/mibs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]
5. **precision microseconds**
6. **optimize timestamp**
7. **history distributions-of-statistics-kept** *size*
8. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
9. **frequency** *seconds*
10. **history hours-of-statistics-kept** *hours*
11. **owner** *owner-id*
12. **request-data-size** *bytes*
13. **history statistics-distribution-interval** *milliseconds*
14. **tag** *text*
15. **threshold** *milliseconds*
16. **timeout** *milliseconds*
17. Do one of the following:
    - **tos** *number*
    - **traffic-class** *number*
18. **flow-label** *number*
19. **verify-data**
20. **vrf** *vrf-name*
21. **end**
22. **show ip sla configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Device(config-ip-sla)# udp-jitter 172.29.139.134 5000`	Configures the IP SLAs operation as a UDP jitter operation and enters UDP jitter configuration submode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices.
Step 5	**precision microseconds** **Example:** `Device(config-ip-sla-jitter)# precision microseconds`	Enables QFP time stamping.
Step 6	**optimize timestamp** **Example:** `Device(config-ip-sla-jitter)# optimize timestamp`	(Optional) For Cisco ASR 1000 Series routers only, optimizes the time stamp location for IP SLAs. **Note** If the device on which the targeted IP SLAs Responder is configured does not also support the optimized time stamp location, the IP SLAs operation will fail.
Step 7	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-jitter)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

	Command or Action	Purpose
Step 8	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-jitter)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 9	**frequency** *seconds* **Example:** `Device(config-ip-sla-jitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 10	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-jitter)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 11	**owner** *owner-id* **Example:** `Device(config-ip-sla-jitter)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 12	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-jitter)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 13	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-jitter)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 14	**tag** *text* **Example:** `Device(config-ip-sla-jitter)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 15	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-jitter)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 16	**timeout** *milliseconds* **Example:**	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

	Command or Action	Purpose
	`Device(config-ip-sla-jitter)# timeout 10000`	

Step 17	Do one of the following:
	• **tos** *number*
	• **traffic-class** *number*
	Example:
	`Device(config-ip-sla-jitter)# tos 160`
	Example:
	`Device(config-ip-sla-jitter)# traffic-class 160`
		(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
		or
		(Optional) In an IPv6 network only, defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

Step 18	**flow-label** *number* **Example:** `Device(config-ip-sla-jitter)# flow-label 112233`	(Optional) In an IPv6 network only, defines the flow label field in the IPv6 header for a supported IP SLAs operation.
Step 19	**verify-data** **Example:** `Device(config-ip-sla-jitter)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 20	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-jitter)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 21	**end** **Example:** `Device(config-ip-sla-jitter)# end`	Returns to privileged EXEC mode.
Step 22	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range*	**schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever**	*seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm* [:*ss*]}] **Example:** `Device(config)# ip sla schedule 10 life forever start-time now` `Device(config)# ip sla group schedule 10 schedule-period frequency`	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose

	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP SLAs operation is not running and not generating statistics, add the **verify-data** command to the configuration of the operation (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP SLAs operation, see the “Configuring Proactive Threshold Monitoring” section.

operation)

To display and interpret the results of an IP SLAs operation, use the **show ip sla statistics** command. Check the output for fields that correspond to criteria in your service level agreement to determine whether the service metrics are acceptable.

Configuration Examples for IP SLAs QFP Time Stamping

Example: Configuring a UDP Operation with QFP Time Stamping

In the following example, two operations are configured as enhanced UDP jitter operations with QFP time stamping and the optimized time stamp location. Operation 2 starts five seconds after the first operation.

Note The device on which ther esponder is configured must (also) support the optimized time stamp location or the probe will fail.

On the source (sender) device:

```
ip sla 1
 udp-jitter 192.0.2.134 5000 num-packets 20
 request-data-size 160
 tos 128
 frequency 30
 precision microseconds     !enables QFP time stamping
 optimize timestamp         !configures optimized time stamp location
ip sla schedule 1 start-time after 00:05:00
ip sla 2
 udp-jitter 192.0.2.134 65052 num-packets 20 interval 10
 request-data-size 20
 tos 64
 frequency 30
 precision microseconds
 optimize timestamp
ip sla schedule 2 start-time after 00:05:05
```

On the destination (responder) device:

```
ip sla responder
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

MIBs

MIBs	MIBs Link
• CISCO-RTTMON-MIB • IPV6-FLOW-LABEL-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

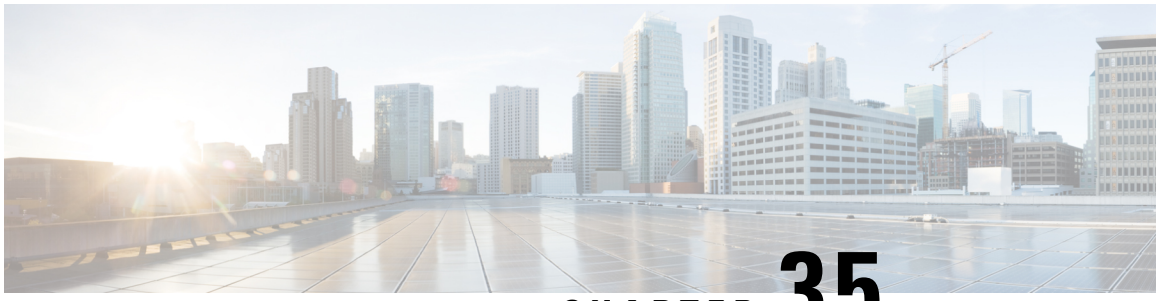
Feature Information for IP SLAs QFP Time Stamping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 56: Feature Information for IP SLAs QFP Time Stamping

Feature Name	Releases	Feature Information
IP SLAs QFP Time Stamping	Cisco IOS XE Release 3.7S	This feature enables IP SLAs Cisco Packet Processor (CPP) time stamping to improve the accuracy of IP SLAs UDP jitter operations. For Cisco ASR 1000 Series routers only, this feature also supports optimizing the time stamp location for more accurate RTT measurements. The following commands were introduced or modified: **optimize timestamp**, **precision microseconds**, **show ip sla configuration**.



CHAPTER 35

Configuring IP SLAs LSP Health Monitor Operations

This module describes how to configure an IP Service Level Agreements (SLAs) label switched path (LSP) Health Monitor. LSP health monitors enable you to to proactively monitor Layer 3 Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). This feature provides automated end-to-end verification in the control plane and data plane for all LSPs between the participating Provider Edge (PE) devices. This end-to-end (PE-to-PE device) approach ensures that LSP connectivity is verified along the paths that customer traffic is sent. Consequently, customer-impacting network connectivity issues that occur within the MPLS core will be detected by the LSP Health Monitor. Once configured, the LSP Health Monitor will automatically create and delete IP SLAs LSP ping or LSP traceroute operations based on network topology.

Prerequisites for LSP Health Monitor Operations

- The participating PE devices of an LSP Health Monitor operation must support the MPLS LSP ping feature. It is recommended that the Provider (P) devices also support the MPLS LSP Ping feature in order to obtain complete error reporting and diagnostics information.
- Ensure that the source PE device has enough memory to support the desired LSP Health Monitor functionality. Enabling the LSP discovery option can potentially have a significant impact on device memory. If there is not enough memory available during the LSP discovery process, the process will gracefully terminate and an error message will be displayed.

Note The destination PE devices of an LSP Health Monitor operation do not require the IP SLAs Responder to be enabled.

Restrictions for LSP Health Monitor Operations

- Once an LSP Health Monitor operation is started, its configuration parameters should not be changed until the operation has ended. Changing the configuration parameters while the operation is actively running could cause delays in obtaining network connectivity statistics.

Information About LSP Health Monitor Operations

Benefits of the LSP Health Monitor

- End-to-end LSP connectivity measurements across equal-cost multipaths for determining network availability or testing network connectivity in MPLS networks
- Proactive threshold monitoring through SNMP trap notifications and syslog messages
- Reduced network troubleshooting time for MPLS networks
- Scalable network error detection using fast retry capability
- Creation and deletion of IP SLAs operations based on network topology
- Discovery of Border Gateway Protocol (BGP) next hop neighbors based on local VPN routing and forwarding instances (VRFs) and global routing tables
- Multioperation scheduling of IP SLAs operations
- Pseudo-wire connectivity testing between MPLS network edges, with threshold violations and scalable operation scheduling
- Monitoring and SNMP trap alerts for round-trip time (RTT) threshold violations, connection loss, and command response timeouts

How the LSP Health Monitor Works

The LSP Health Monitor feature provides the capability to proactively monitor Layer 3 MPLS VPNs. The general process for how the LSP Health Monitor works is as follows:

1. The user configures an LSP Health Monitor operation and the BGP next hop neighbor discovery process is enabled.

   Configuring an LSP Health Monitor operation is similar to configuring a standard IP SLAs operation. To illustrate, all operation parameters for an LSP Health Monitor operation are configured after an identification number for the operation is specified. However, unlike standard IP SLAs operations, these configured parameters are then used as the base configuration for the individual IP SLAs LSP ping and LSP traceroute operations that will be created by the LSP Health Monitor. The LSP discovery process can potentially have a significant impact on the memory and CPU of the source PE device. To prevent unnecessary device performance issues, careful consideration should be taken when configuring the operational and scheduling parameters of an LSP Health Monitor operation.

When the BGP next hop neighbor discovery process is enabled, a database of BGP next hop neighbors in use by any VRF associated with the source PE device is generated based on information from the local VRF and global routing tables. For more information about the BGP next hop neighbor discovery process, see the "Discovery of Neighboring PE Devices" section.

Note By default, only a single path between the source and destination PE devices is discovered. If the LSP discovery option is enabled, the equal-cost multipaths between the source and destination PE devices are discovered. For more information on how the LSP discovery process works, see the "LSP Discovery Process" section.

2. The user configures proactive threshold monitoring parameters for the LSP Health Monitor operation. For more information about proactive threshold monitoring, see the "Proactive Threshold Monitoring for the LSP Health Monitor" section.

   Depending on the proactive threshold monitoring configuration options chosen, SNMP trap notifications or syslog messages are generated as threshold violations are met.

3. The user configures multioperation scheduling parameters for the LSP Health Monitor operation. For more information about multioperation scheduling, see the "Multioperation Scheduling for the LSP Health Monitor" section.

   Once the LSP Health Monitor operation is started, a single IP SLAs operation is automatically created (based on parameters configured in Step 1) for each applicable PE (BGP next hop) neighbor. The IP SLAs operations will measure network connectivity between the source PE device and the discovered destination PE device. The start time and frequency of each measurement is based on the multioperation scheduling parameters defined by the user.

Addition and Deletion of IP SLAs Operations

The LSP Health Monitor receives periodic notifications about BGP next hop neighbors that have been added to or removed from a particular VPN. This information is stored in a queue maintained by the LSP Health Monitor. Based on the information in the queue and user-specified time intervals, new IP SLAs operations are automatically created for newly discovered PE devices and existing IP SLAs operations are automatically deleted for any PE devices that are no longer valid. The automatic deletion of operations can be disabled. However, disabling this function is not recommended because these operations would then need to be deleted manually.

If the LSP discovery option is enabled, creation of LSP discovery groups for newly discovered BGP next hop neighbors will follow the same process as described in the "LSP Discovery Process" section. If a BGP next hop neighbor is removed from a particular VPN, all the corresponding LSP discovery groups and their associated individual IP SLAs operations and statistics are removed from the LSP discovery group database.

Access Lists for Filtering BGP Next Hop Neighbors

Standard IP access lists can be configured to restrict the number of IP SLAs operations that are automatically created by the LSP Health Monitor. When the IP SLAs access list parameter is configured, the list of BGP next hop neighbors discovered by the LSP Health Monitor is filtered based on the conditions defined by the associated standard IP access list. In other words, the LSP Health Monitor will automatically create IP SLAs operations only for those BGP next hop neighbors with source addresses that satisfy the criteria permitted by the standard IP access list.

Unique Identifier for Each Automatically Created IP SLAs Operation

The IP SLAs operations automatically created by the LSP Health Monitor are uniquely identified by their owner field. The owner field of an operation is generated using all the parameters that can be configured for that particular operation. If the length of the owner field is longer than 255 characters, it will be truncated.

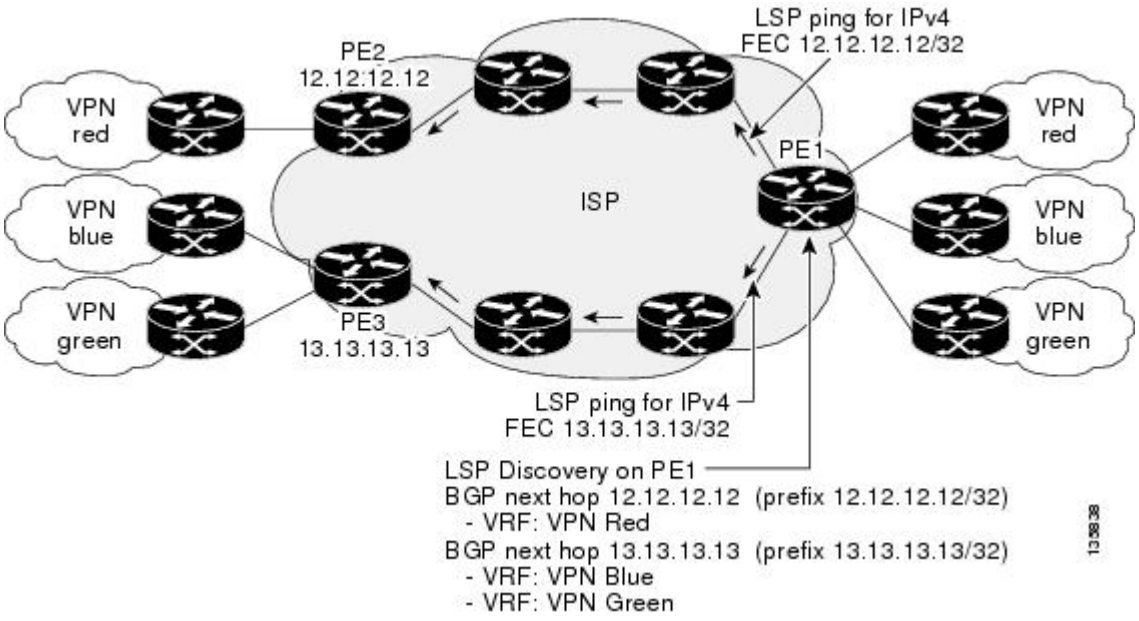
Discovery of Neighboring PE Devices

A BGP next hop neighbor discovery process is used to find the BGP next hop neighbors in use by any VRF associated with the source PE device. In most cases, these neighbors will be PE devices.

When the BGP next hop neighbor discovery process is enabled, a database of BGP next hop neighbors in use by any VRF associated with the source PE device is generated based on information from the local VRF and global routing tables. As routing updates are received, new BGP next hop neighbors are added to and deleted from the database immediately.

The figure below shows how the BGP next hop neighbor discovery process works for a simple VPN scenario for an Internet service provider (ISP). In this example, there are three VPNs associated with device PE1: red, blue, and green. From the perspective of device PE1, these VPNs are reachable remotely through BGP next hop neighbors PE2 (device ID: 12.12.12.12) and PE3 (device ID: 13.13.13.13). When the BGP next hop neighbor discovery process is enabled on device PE1, a database is generated based on the local VRF and global routing tables. The database in this example contains two BGP next hop device entries: PE2 12.12.12.12 and PE3 13.13.13.13. The routing entries are maintained per next hop device to distinguish which next hop devices belong within which particular VRF. For each next hop device entry, the IPv4 Forward Equivalence Class (FEC) of the BGP next hop device in the global routing table is provided so that it can be used by the MPLS LSP ping operation.

Figure 38: BGP Next Hop Neighbor Discovery for a Simple VPN



LSP Discovery

The LSP discovery option of an LSP Health Monitor operation provides the capability to discover the equal-cost multipaths for carrying MPLS traffic between the source and destination PE devices. Network connectivity measurements can then be performed for each of the paths that were discovered.

The general process for LSP discovery is as follows:

1. BGP next hop neighbors are discovered using the BGP next hop neighbor discovery process. For more information about the BGP next hop neighbor discovery process, see the "Discovery of Neighboring PE Routers" section.

   Once the LSP Health Monitor operation is started, a single IP SLAs operation is automatically created for each applicable PE (BGP next hop) neighbor. Only a single path to each applicable PE neighbor is discovered during this initial step of the LSP discovery process. For each next hop neighbor, the LSP Health Monitor creates an LSP discovery group (that initially consists of only the one discovered path) and assigns the group with a unique identification number. For more information about LSP discovery groups, see the "LSP Discovery Groups" section.

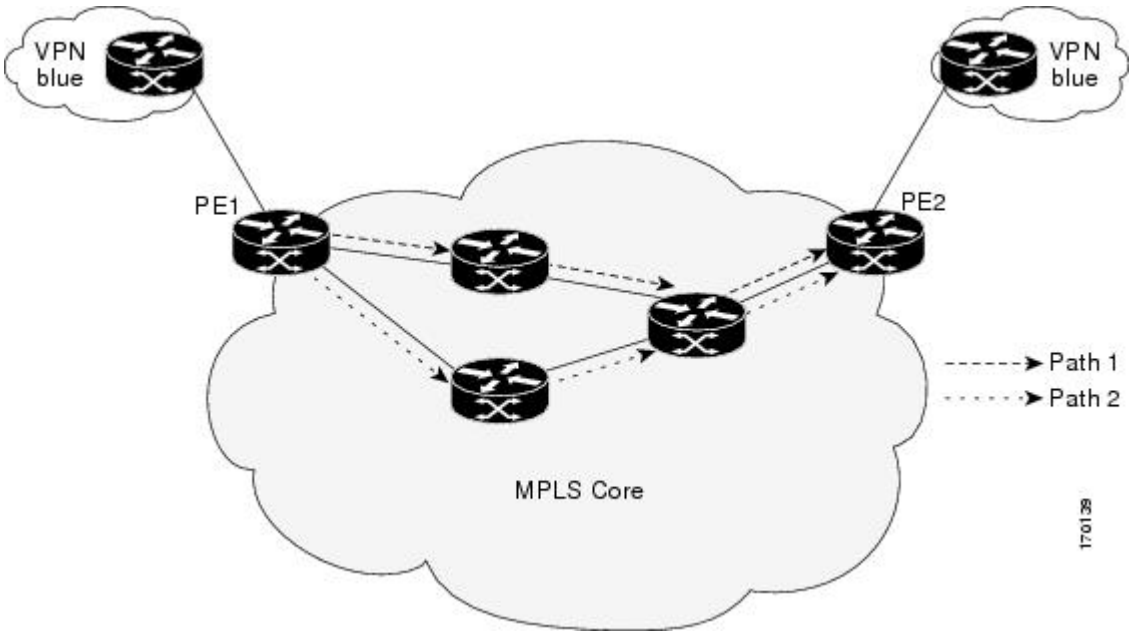
2. An LSP discovery request is sent by the LSP Health Monitor to the LSP discovery subsystem for each applicable BGP next hop neighbor. For each next hop neighbor in which an appropriate response is received, MPLS echo requests are sent one-by-one from the source PE device to discover the equal-cost multipaths. The parameters that uniquely identify each equal-cost multipath (127/8 destination IP address [LSP selector] and the PE outgoing interface) are added to the associated LSP discovery database.

Note For a given LSP Health Monitor operation, the user can define the maximum number of BGP next hop neighbors that can be concurrently undergoing LSP discovery.

3. Each individual IP SLAs operation (created for each applicable PE neighbor) uses an IP SLAs LSP ping superoperation to measure network connectivity across all equal-cost multipaths between the source PE device and discovered destination PE device. The IP SLAs superoperation operates by sending an LSP ping packet to the destination PE device and adjusting the LSP ping 127/8 LSP selector IP address for each discovered equal-cost multipath. For example, assume that there are three equal-cost multipaths to a destination PE device and the identified LSP selector IP addresses are 127.0.0.1, 127.0.0.5, and 127.0.0.6. The IP SLAs superoperation would sequentially send three LSP ping packets using the identified LSP selector IP addresses for directing the superoperation across the three paths. This technique ensures that there is only a single IP SLAs LSP ping operation for each source and destination PE device pair, and significantly reduces the number of active LSP ping operations sent by the source PE device.

The figure below illustrates a simple VPN scenario. This network consists of a core MPLS VPN with two PE devices (device PE1 and device PE2) belonging to the VRF named VPN blue. Suppose device PE1 is the source PE device for an LSP Health Monitor operation with the LSP discovery option enabled and that device PE2 is discovered by the BGP discovery process as a BGP next hop neighbor to device PE1. If path 1 and path 2 are equal-cost multipaths between device PE1 to device PE2, then the LSP discovery process would create an LSP discovery group consisting of path 1 and path 2. An IP SLAs LSP ping superoperation would also be created to monitor network availability across each path.

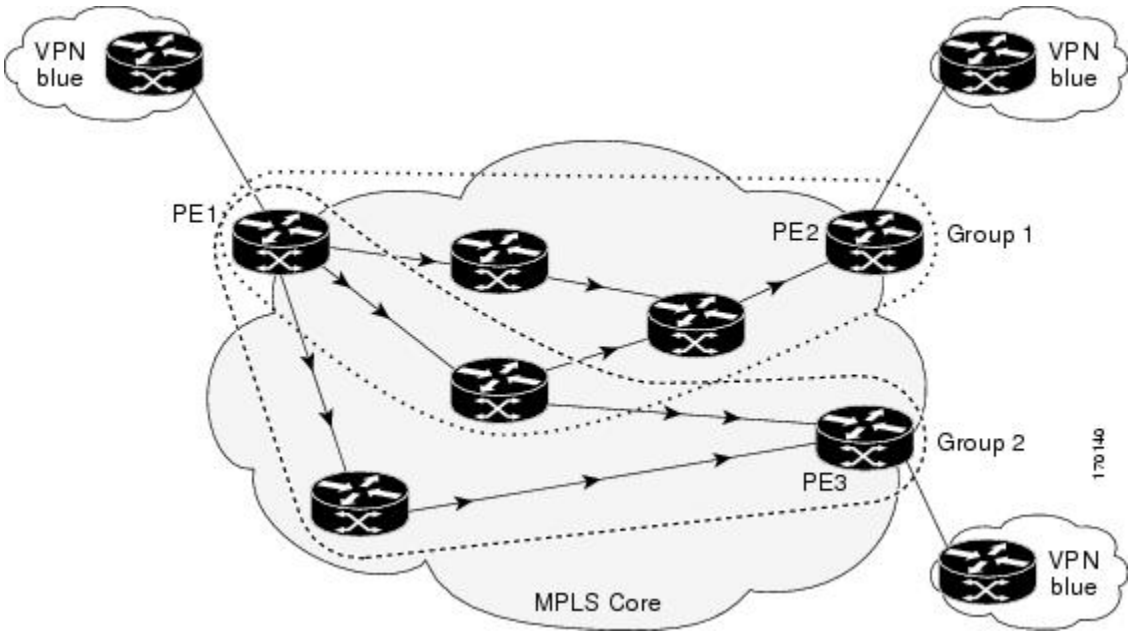
Figure 39: LSP Discovery for a Simple VPN



LSP Discovery Groups

A single LSP Health Monitor operation can be comprised of several LSP discovery groups depending on the number of BGP next hop neighbors discovered by the BGP next hop neighbor discovery process. Each LSP discovery group corresponds to one BGP next hop neighbor and is assigned a unique identification number (starting with the number 1). The figure below illustrates a simple VPN scenario. This network consists of a core MPLS VPN with three PE devices (device PE1, PE2, and PE3) belonging to the VRF named VPN blue. Suppose device PE1 is the source PE device for an LSP Health Monitor operation with the LSP discovery option enabled and that device PE2 and PE3 are discovered by the BGP discovery process as BGP next hop neighbors to device PE1. LSP discovery group 1 is created for the equal-cost multipaths between device PE1 to device PE2 and LSP discovery group 2 is created for the equal-cost multipaths between device PE1 to device PE3.

Figure 40: LSP Discovery Groups for a Simple VPN



Once the LSP Health Monitor operation is started, a single IP SLAs operation is automatically created for each applicable PE (BGP next hop) neighbor. Each IP SLAs operation (created for each applicable PE neighbor) uses an IP SLAs LSP ping superoperation to measure network connectivity across all equal-cost multipaths between the source PE device and discovered destination PE device. Each LSP ping superoperation corresponds to a single LSP discovery group.

The LSP ping superoperation operates by sending an LSP ping packet to the destination PE device and adjusting the LSP ping 127/8 LSP selector IP address for each discovered equal-cost multipath. The network connectivity statistics collected by each equal-cost multipath is aggregated and stored in one-hour increments (data can be collected for a maximum of two hours). Results are stored as group averages representative of all the equal-cost multipaths within the LSP discovery group for a given one-hour increment.

Each equal-cost multipath discovered between the source PE device and a BGP next hop neighbor is uniquely identified with the following parameters:

- 127/8 destination IP address (LSP selector) within the local host IP address range
- PE outgoing interface

The database for an LSP discovery group is updated if any of the following events occur:

- The corresponding LSP ping superoperation sends an LSP ping packet.
- An active equal-cost multipath is added to or deleted from the LSP discovery group.
- The user enters the Cisco command to delete all the aggregated statistical data for a particular LSP discovery group.

IP SLAs LSP Ping and LSP Traceroute

The LSP Health Monitor feature introduces support for the IP SLAs LSP ping and IP SLAs LSP traceroute operations. These operations are useful for troubleshooting network connectivity issues and determining network availability in an MPLS VPN. When using the LSP Health Monitor, IP SLAs LSP ping and LSP traceroute operations are automatically created to measure network connectivity between the source PE device and the discovered destination PE devices. Individual IP SLAs LSP ping and LSP traceroute operations can also be manually configured. Manual configuration of these operations can be useful for troubleshooting a connectivity issue.

The IP SLAs LSP ping and IP SLAs LSP traceroute operations are based on the same infrastructure used by the MPLS LSP Ping and MPLS LSP Traceroute features, respectively, for sending and receiving echo reply and request packets to test LSPs.

The LSP discovery does not support IP SLAs traceroute operations.

Proactive Threshold Monitoring for the LSP Health Monitor

Proactive threshold monitoring support for the LSP Health Monitor feature provides the capability for triggering SNMP trap notifications and syslog messages when user-defined reaction conditions (such as a connection loss or timeout) are met. Configuring threshold monitoring for an LSP Health Monitor operation is similar to configuring threshold monitoring for a standard IP SLAs operation.

LSP Discovery Option Enabled

If the LSP discovery option for an LSP Health Monitor operation is enabled, SNMP trap notifications can be generated when one of the following events occurs:

- LSP discovery for a particular BGP next hop neighbor fails.
- Operational status of an LSP discovery group changes.

Possible reasons for which LSP discovery can fail for a particular BGP next hop neighbor are as follows:

- Expiration of time allowed for a BGP next hop neighbor to respond to an LSP discovery request.
- Return code is “Broken” or “Unexplorable” for all paths leading to the BGP next hop neighbor.

The table below describes the conditions for which the operational status of an LSP discovery group can change. Whenever an individual IP SLAs LSP ping operation of an LSP discovery group is executed, a return code is generated. Depending on the value of the return code and the current status of the LSP discovery group, the group status can change.

Table 57: Conditions for Which an LSP Discovery Group Status Changes

Individual IP SLAs Operation Return Code	Current Group Status = UP	Current Group Status = PARTIAL	Current Group Status = DOWN
OK	No group status change.	If return codes for all paths in the group are OK, then the group status changes to UP.	Group status changes to PARTIAL.

Individual IP SLAs Operation Return Code	Current Group Status = UP	Current Group Status = PARTIAL	Current Group Status = DOWN
Broken or Unexplorable	Group status changes to PARTIAL.	If return codes for all paths in the group are Broken or Unexplorable, then the group status changes to DOWN.	No group status change.

The return code for an individual IP SLAs LSP ping operation can be one of the following:

- OK--Indicates that the LSP is working properly. The customer VPN traffic will be sent across this path.
- Broken--Indicates that the LSP is broken. Customer VPN traffic will not be sent across this path and may be discarded.
- Unexplorable--Indicates that not all the paths to this PE neighbor have been discovered. This may be due to a disruption along the LSP or because the number of 127/8 IP addresses used for LSP selection has been exhausted.

The status of an LSP discovery group can be one of the following:

- UNKNOWN--Indicates that group status has not yet been determined and that the paths belonging to the group are in the process of being tested for the first time. Once this initial test is complete, the group status will change to UP, PARTIAL, or DOWN.
- UP--Indicates that all the paths within the group are active and no operation failures have been detected.
- PARTIAL--Indicates that an operation failure has been detected for one or more, but not all, of the paths within the group.
- DOWN--Indicates that an operation failure has been detected for all the paths within the group.

Secondary Frequency Option

With the introduction of the LSP Health Monitor feature, a new threshold monitoring parameter has been added that allows you to specify a secondary frequency. If the secondary frequency option is configured and a failure (such as a connection loss or timeout) is detected for a particular path, the frequency at which the path is remeasured will increase to the secondary frequency value (testing at a faster rate). When the configured reaction condition is met (such as N consecutive connection losses or N consecutive timeouts), an SNMP trap and syslog message can be sent and the measurement frequency will return to its original frequency value.

Multioperation Scheduling for an LSP Health Monitor

Multioperation scheduling support for the LSP Health Monitor feature provides the capability to easily schedule the automatically created IP SLAs operations (for a given LSP Health Monitor operation) to begin at intervals equally distributed over a specified duration of time (schedule period) and to restart at a specified frequency. Multioperation scheduling is particularly useful in cases where the LSP Health Monitor is enabled on a source PE device that has a large number of PE neighbors and, therefore, a large number of IP SLAs operations running at the same time.

Newly created IP SLAs operations (for newly discovered BGP next hop neighbors) are added to the same schedule period as the operations that are currently running. To prevent too many operations from starting at

the same time, the multioperation scheduling feature will schedule the operations to begin at random intervals uniformly distributed over the schedule period.

Configuring a multioperation schedule for an LSP Health Monitor is similar to configuring a standard multioperation schedule for a group of individual IP SLAs operations.

LSP Discovery Enabled

When a multioperation schedule for an LSP Health Monitor operation with LSP discovery is started, the BGP next hop neighbors are discovered, and network connectivity to each applicable neighbor is monitored using only a single LSP. Initially, network connectivity between the source PE device and discovered destination PE device is measured across only a single path. This initial condition is the same as if an LSP Health Monitor operation was performed without LSP discovery.

Specific information about the IP SLAs LSP ping operations that are created for newly discovered equal-cost paths during the succeeding iterations of the LSP discovery process are stored in the LSP discovery group database. These newly created IP SLAs LSP ping operations will start collecting data at the next iteration of network connectivity measurements for their associated LSP discovery group.

The start times for the individual IP SLAs LSP ping operations for each LSP discovery group is based on the number of LSP discovery groups and the schedule period of the multioperation schedule. For example, if three LSP discovery groups (Group 1, Group 2, and Group 3) are scheduled to run over a period of 60 seconds, the first LSP ping operation of Group 1 will start at 0 seconds, the first LSP ping operation of Group 2 will start at 20 seconds, and the first LSP ping operation of Group 3 will start at 40 seconds. The remaining individual IP SLAs LSP ping operations for each LSP discovery group will run sequentially after completion of the first LSP ping operation. For each LSP discovery group, only one LSP ping operation runs at a time.

How to Configure LSP Health Monitor Operations

Configuring an LSP Health Monitor Operation

Perform only one of the following tasks:

Configuring an LSP Health Monitor Operation without LSP Discovery on a PE Device

Note If LSP discovery is disabled, only a single path between the source PE device and each BGP next hop neighbor is discovered.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls discovery vpn next-hop**
4. **mpls discovery vpn interval** *seconds*
5. **auto ip sla mpls-lsp-monitor** *operation-number*
6. Do one of the following:
   - **type echo** [**ipsla-vrf-all** | **vrf** *vpn-name*]

- **type pathEcho** [**ipsla-vrf-all** | **vrf** *vpn-name*]

7. **access-list** *access-list-number*
8. **scan-interval** *minutes*
9. **delete-scan-factor** *factor*
10. **force-explicit-null**
11. **exp** *exp-bits*
12. **lsp-selector** *ip-address*
13. **reply-dscp-bits** *dscp-value*
14. **reply-mode** {**ipv4** | **router-alert**}
15. **request-data-size** *bytes*
16. **secondary-frequency** {**both** | **connection-loss** | **timeout**} *frequency*
17. **tag** *text*
18. **threshold** *milliseconds*
19. **timeout** *milliseconds*
20. **ttl** *time-to-live*
21. **exit**
22. **auto ip sla mpls-lsp-monitor reaction-configuration** *operation-number* **react** {**connectionLoss** | **timeout**} [**action-type** *option*] [**threshold-type** {**consecutive** [*occurrences*] | **immediate** | **never**}]
23. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls discovery vpn next-hop** **Example:** `Device(config)# mpls discovery vpn next-hop`	(Optional) Enables the MPLS VPN BGP next hop neighbor discovery process. **Note** This command is automatically enabled when the **auto ip sla mpls-lsp-monitor** command is entered.
Step 4	**mpls discovery vpn interval** *seconds* **Example:** `Device(config)# mpls discovery vpn interval 120`	(Optional) Specifies the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor discovery database of an MPLS VPN.
Step 5	**auto ip sla mpls-lsp-monitor** *operation-number* **Example:**	Begins configuration for an LSP Health Monitor operation and enters auto IP SLA MPLS configuration mode.

	Command or Action	Purpose
	`Device(config)# auto ip sla mpls-lsp-monitor 1`	**Note** Entering this command automatically enables the **mpls discovery vpn next-hop** command.

Step 6	Do one of the following:
	• **type echo** [**ipsla-vrf-all** | **vrf** *vpn-name*]
	• **type pathEcho** [**ipsla-vrf-all** | **vrf** *vpn-name*]
	Example:
	`Device(config-auto-ip-sla-mpls)# type echo ipsla-vrf-all`
	Example:
	`Device(config-auto-ip-sla-mpls)# type pathEcho ipsla-vrf-all`
		Enters MPLS parameters configuration submode and allows the user to configure the parameters for an IP SLAs LSP ping operation using the LSP Health Monitor.
		or
		Enters MPLS parameters configuration submode and allows the user to configure the parameters for an IP SLAs LSP traceroute operation using the LSP Health Monitor.

Step 7	**access-list** *access-list-number* **Example:** `Device(config-auto-ip-sla-mpls-params)# access-list 10`	(Optional) Specifies the access list to apply to an LSP Health Monitor operation.
Step 8	**scan-interval** *minutes* **Example:** `Device(config-auto-ip-sla-mpls-params)# scan-interval 5`	(Optional) Sets the timer for the IP SLAs LSP Health Monitor database.
Step 9	**delete-scan-factor** *factor* **Example:** `Device(config-auto-ip-sla-mpls-params)# delete-scan-factor 2`	(Optional) Specifies the number of times the LSP Health Monitor should check the scan queue before automatically deleting IP SLAs operations for BGP next hop neighbors that are no longer valid. • The default scan factor is 1. Each time the LSP Health Monitor checks the scan queue for updates, it deletes IP SLAs operations for BGP next hop neighbors that are no longer valid. • If the scan factor is set to 0, IP SLAs operations will not be automatically deleted by the LSP Health Monitor. This configuration is not recommended. • This command must be used with the **scan-interval** command.
Step 10	**force-explicit-null** **Example:** `Device(config-auto-ip-sla-mpls-params)# force-explicit-null`	(Optional) Adds an explicit null label to all echo request packets of an IP SLAs operation.

	Command or Action	Purpose
Step 11	**exp** *exp-bits* **Example:** `Device(config-auto-ip-sla-mpls-params)# exp 5`	(Optional) Specifies the experimental field value in the header for an echo request packet of an IP SLAs operation.
Step 12	**lsp-selector** *ip-address* **Example:** `Device(config-auto-ip-sla-mpls-params)# lsp-selector 127.0.0.10`	(Optional) Specifies the local host IP address used to select the LSP of an IP SLAs operation.
Step 13	**reply-dscp-bits** *dscp-value* **Example:** `Device(config-auto-ip-sla-mpls-params)# reply-dscp-bits 5`	(Optional) Specifies the differentiated services codepoint (DSCP) value for an echo reply packet of an IP SLAs operation.
Step 14	**reply-mode** {**ipv4** | **router-alert**} **Example:** `Device(config-auto-ip-sla-mpls-params)# reply-mode router-alert`	(Optional) Specifies the reply mode for an echo request packet of an IP SLAs operation. • The default reply mode is an IPv4 UDP packet.
Step 15	**request-data-size** *bytes* **Example:** `Device(config-auto-ip-sla-mpls-params)# request-data-size 200`	(Optional) Specifies the protocol data size for a request packet of an IP SLAs operation.
Step 16	**secondary-frequency** {**both** | **connection-loss** | **timeout**} *frequency* **Example:** `Device(config-auto-ip-sla-mpls-params)# secondary-frequency connection-loss 10`	(Optional) Sets the faster measurement frequency (secondary frequency) to which an IP SLAs operation should change when a reaction condition occurs.
Step 17	**tag** *text* **Example:** `Device(config-auto-ip-sla-mpls-params)# tag testgroup`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 18	**threshold** *milliseconds* **Example:** `Device(config-auto-ip-sla-mpls-params)# threshold 6000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.

	Command or Action	Purpose
Step 19	**timeout** *milliseconds* **Example:** `Device(config-auto-ip-sla-mpls-params)# timeout 7000`	(Optional) Specifies the amount of time the IP SLAs operation waits for a response from its request packet.
Step 20	**ttl** *time-to-live* **Example:** `Device(config-auto-ip-sla-mpls-params)# ttl 200`	(Optional) Specifies the maximum hop count for an echo request packet of an IP SLAs operation.
Step 21	**exit** **Example:** `Device(config-auto-ip-sla-mpls-params)# exit`	Exits MPLS parameters configuration submode and returns to global configuration mode.
Step 22	**auto ip sla mpls-lsp-monitor reaction-configuration** *operation-number* **react** {**connectionLoss** | **timeout**} [**action-type** *option*] [**threshold-type** {**consecutive** [*occurrences*] | **immediate** | **never**}] **Example:** `Device(config)# auto ip sla mpls-lsp-monitor reaction-configuration 1 react connectionLoss action-type trapOnly threshold-type consecutive 3`	(Optional) Configures certain actions to occur based on events under the control of the LSP Health Monitor.
Step 23	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the LSP Health Monitor Operation with LSP Discovery on a PE Device

Note

- The LSP Health Monitor with LSP Discovery feature supports Layer 3 MPLS VPNs only.
- The LSP discovery option does not support IP SLAs LSP traceroute operations.
- The LSP discovery option does not support IP SLAs VCCV operations.
- The LSP discovery process can potentially have a significant impact on the memory and CPU of the source PE device. To prevent unnecessary device performance issues, careful consideration should be taken when configuring the operational and scheduling parameters of an LSP Health Monitor operation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **mpls discovery vpn next-hop**
4. **mpls discovery vpn interval** *seconds*
5. **auto ip sla mpls-lsp-monitor** *operation-number*
6. **type echo** [**ipsla-vrf-all** | **vrf** *vpn-name*]
7. Configure optional parameters for the IP SLAs LSP echo operation.
8. **path-discover**
9. **hours-of-statistics-kept** *hours*
10. **force-explicit-null**
11. **interval** *milliseconds*
12. **lsp-selector-base** *ip-address*
13. **maximum-sessions** *number*
14. **scan-period** *minutes*
15. **session-timeout** *seconds*
16. **timeout** *seconds*
17. **exit**
18. **exit**
19. **auto ip sla mpls-lsp-monitor reaction-configuration** *operation-number* **react lpd** {**lpd-group** [**retry** *number*] | **tree-trace**} [**action-type trapOnly**]
20. **ip sla logging traps**
21. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls discovery vpn next-hop** **Example:** `Device(config)# mpls discovery vpn next-hop`	(Optional) Enables the MPLS VPN BGP next hop neighbor discovery process. **Note** This command is automatically enabled when the **auto ip sla mpls-lsp-monitor** command is entered.
Step 4	**mpls discovery vpn interval** *seconds* **Example:** `Device(config)# mpls discovery vpn interval 120`	(Optional) Specifies the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor discovery database of an MPLS VPN.

	Command or Action	Purpose
Step 5	**auto ip sla mpls-lsp-monitor** *operation-number* **Example:** `Device(config)# auto ip sla mpls-lsp-monitor 1`	Begins configuration for an LSP Health Monitor operation and enters auto IP SLAs MPLS configuration mode. **Note** Entering this command automatically enables the **mpls discovery vpn next-hop** command.
Step 6	**type echo** [**ipsla-vrf-all** | **vrf** *vpn-name*] **Example:** `Device(config-auto-ip-sla-mpls)# type echo ipsla-vrf-all`	Enters MPLS parameters configuration mode and allows the user to configure the parameters for an IP SLAs LSP ping operation using the LSP Health Monitor.
Step 7	Configure optional parameters for the IP SLAs LSP echo operation.	(Optional) See Steps 7 through 21 in the "Configuring an LSP Health Monitor Operation Without LSP Discovery on a PE Device" section.
Step 8	**path-discover** **Example:** `Device(config-auto-ip-sla-mpls-params)# path-discover`	Enables the LSP discovery option for an IP SLAs LSP Health Monitor operation and enters LSP discovery parameters configuration submode.
Step 9	**hours-of-statistics-kept** *hours* **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# hours-of-statistics-kept 1`	(Optional) Sets the number of hours for which LSP discovery group statistics are maintained for an LSP Health Monitor operation.
Step 10	**force-explicit-null** **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# force-explicit-null`	(Optional) Adds an explicit null label to all echo request packets of an LSP Health Monitor operation.
Step 11	**interval** *milliseconds* **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# interval 2`	(Optional) Specifies the time interval between MPLS echo requests that are sent as part of the LSP discovery process for an LSP Health Monitor operation.
Step 12	**lsp-selector-base** *ip-address* **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# lsp-selector-base 127.0.0.2`	(Optional) Specifies the base IP address used to select the LSPs belonging to the LSP discovery groups of an LSP Health Monitor operation.

	Command or Action	Purpose
Step 13	**maximum-sessions** *number* **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# maximum-sessions 2`	(Optional) Specifies the maximum number of BGP next hop neighbors that can be concurrently undergoing LSP discovery for a single LSP Health Monitor operation. **Note** Careful consideration should be used when configuring this parameter to avoid a negative impact on the device's CPU.
Step 14	**scan-period** *minutes* **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# scan-period 30`	(Optional) Sets the amount of time after which the LSP discovery process can restart for an LSP Health Monitor operation.
Step 15	**session-timeout** *seconds* **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# session-timeout 60`	(Optional) Sets the amount of time the LSP discovery process for an LSP Health Monitor operation waits for a response to its LSP discovery request for a particular BGP next hop neighbor.
Step 16	**timeout** *seconds* **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# timeout 4`	(Optional) Sets the amount of time the LSP discovery process for an LSP Health Monitor operation waits for a response to its echo request packets. **Note** Careful consideration should be used when configuring this parameter to avoid a negative impact on the device's CPU.
Step 17	**exit** **Example:** `Device(config-auto-ip-sla-mpls-lpd-params)# exit`	Exits LSP discovery parameters configuration submode and returns to MPLS parameters configuration mode.
Step 18	**exit** **Example:** `Device(config-auto-ip-sla-mpls-params)# exit`	Exits MPLS parameters configuration mode and returns to global configuration mode.
Step 19	**auto ip sla mpls-lsp-monitor reaction-configuration** *operation-number* **react lpd** {**lpd-group** [**retry** *number*] | **tree-trace**} [**action-type trapOnly**] **Example:** `Device(config)# auto ip sla mpls-lsp-monitor reaction-configuration 1 react lpd lpd-group retry 3 action-type trapOnly`	(Optional) Configures the proactive threshold monitoring parameters for an LSP Health Monitor operation with LSP discovery enabled.
Step 20	**ip sla logging traps** **Example:**	(Optional) Enables the generation of SNMP system logging messages specific to IP SLAs trap notifications.

	Command or Action	Purpose
	`Device(config)# ip sla logging traps`	
Step 21	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Scheduling LSP Health Monitor Operations

Note
- The LSP discovery process can potentially have a significant impact on the memory and CPU of the source PE device. Careful consideration should be taken when configuring the scheduling parameters to prevent too many IP SLAs LSP ping operations from running at the same time. The schedule period should be set to a relatively large value for large MPLS VPNs.
- Newly created IP SLAs operations (for newly discovered BGP next hop neighbors) are added to the same mulioperation schedule period as the operations that are currently running. To prevent too many operations from starting at the same time, the multioperation scheduler will schedule the operations to begin at random intervals uniformly distributed over the schedule period.

Before you begin

- All IP SLAs operations to be scheduled must be already configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **auto ip sla mpls-lsp-monitor schedule** *operation-number* **schedule-period** *seconds* [**frequency** [*seconds*]] [**start-time** {**after** *hh* **:** *mm* **:** *ss* | *hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **now** | **pending**}]
4. **exit**
5. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**auto ip sla mpls-lsp-monitor schedule** *operation-number* **schedule-period** *seconds* [**frequency** [*seconds*]] [**start-time** {**after** *hh* **:** *mm* **:** *ss* | *hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **now** | **pending**}] **Example:** `Device(config)# auto ip sla mpls-lsp-monitor schedule 1 schedule-period 60 start-time now`	Configures the scheduling parameters for an LSP Health Monitor operation.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays the IP SLAs configuration details.

Troubleshooting Tips

Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an individual IP SLAs LSP ping or LSP traceroute operation. Use the **debug ip sla mpls-lsp-monitor** command to help troubleshoot issues with an IP SLAs LSP Health Monitor operation.

What to Do Next

To display the results of an individual IP SLAs operation use the **show ip sla statistics** and **show ip sla statistics aggregated** commands. Checking the output for fields that correspond to criteria in your service level agreement will help you determine whether the service metrics are acceptable.

Manually Configuring and Scheduling an IP SLAs LSP Ping or LSP Traceroute Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. Do one of the following:
   - **mpls lsp ping ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}]
   - **mpls lsp trace ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}]
5. **exp** *exp-bits*

6. **request-data-size** *bytes*
7. **secondary-frequency** {**connection-loss** | **timeout**} *frequency*
8. **tag** *text*
9. **threshold** *milliseconds*
10. **timeout** *milliseconds*
11. **ttl** *time-to-live*
12. **exit**
13. **ip sla reaction-configuration** *operation-number* [**react** *monitored-element*] [**threshold-type** {**never** | **immediate** | **consecutive** [*consecutive-occurrences*] | **xofy** [*x-value y-value*] | **average** [*number-of-probes*]}] [**threshold-value** *upper-threshold lower-threshold*] [**action-type** {**none** | **trapOnly** | **triggerOnly** | **trapAndTrigger**}]
14. **ip sla logging traps**
15. **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**]
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 1`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	Do one of the following: • **mpls lsp ping ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}] • **mpls lsp trace ipv4** *destination-address destination-mask* [**force-explicit-null**] [**lsp-selector** *ip-address*] [**src-ip-addr** *source-address*] [**reply** {**dscp** *dscp-value* | **mode** {**ipv4** | **router-alert**}}] **Example:** `Device(config-ip-sla)# mpls lsp ping ipv4 192.168.1.4 255.255.255.255 lsp-selector 127.1.1.1`	• The first example configures the IP SLAs operation as an LSP ping operation and enters LSP ping configuration mode. • The second example configures the IP SLAs operation as an LSP trace operation and enters LSP trace configuration mode.

	Command or Action	Purpose
	Example: `Device(config-ip-sla)# mpls lsp trace ipv4 192.168.1.4 255.255.255.255 lsp-selector 127.1.1.1`	
Step 5	**exp** *exp-bits* **Example:** `Device(config-sla-monitor-lspPing)# exp 5`	(Optional) Specifies the experimental field value in the header for an echo request packet of an IP SLAs operation. **Note** The LSP ping configuration mode is used in this example and in the remaining steps. Except where noted, the same commands are also supported in the LSP trace configuration mode.
Step 6	**request-data-size** *bytes* **Example:** `Device(config-sla-monitor-lspPing)# request-data-size 200`	(Optional) Specifies the protocol data size for a request packet of an IP SLAs operation.
Step 7	**secondary-frequency** {**connection-loss** | **timeout**} *frequency* **Example:** `Device(config-sla-monitor-lspPing)# secondary-frequency connection-loss 10`	(Optional) Sets the faster measurement frequency (secondary frequency) to which an IP SLAs operation should change when a reaction condition occurs. • This command is for IP SLAs LSP ping operations only. LSP trace configuration mode does not support this command.
Step 8	**tag** *text* **Example:** `Device(config-sla-monitor-lspPing)# tag testgroup`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 9	**threshold** *milliseconds* **Example:** `Device(config-sla-monitor-lspPing)# threshold 6000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 10	**timeout** *milliseconds* **Example:** `Device(config-sla-monitor-lspPing)# timeout 7000`	(Optional) Specifies the amount of time the IP SLAs operation waits for a response from its request packet.
Step 11	**ttl** *time-to-live* **Example:** `Device(config-sla-monitor-lspPing)# ttl 200`	(Optional) Specifies the maximum hop count for an echo request packet of an IP SLAs operation.
Step 12	**exit** **Example:**	Exits LSP ping or LSP trace configuration submode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-sla-monitor-lspPing)# exit`	
Step 13	**ip sla reaction-configuration** *operation-number* [**react** *monitored-element*] [**threshold-type** {**never** | **immediate** | **consecutive** [*consecutive-occurrences*] | **xofy** [*x-value y-value*] | **average** [*number-of-probes*]}] [**threshold-value** *upper-threshold lower-threshold*] [**action-type** {**none** | **trapOnly** | **triggerOnly** | **trapAndTrigger**}] **Example:** `Device(config)# ip sla reaction-configuration 1 react connectionLoss threshold-type consecutive 3 action-type traponly`	(Optional) Configures certain actions to occur based on events under the control of IP SLAs.
Step 14	**ip sla logging traps** **Example:** `Device(config)# ip sla logging traps`	(Optional) Enables the generation of SNMP system logging messages specific to IP SLAs trap notifications.
Step 15	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**] **Example:** `Device(config)# ip sla schedule 1 start-time now`	Configures the scheduling parameters for an IP SLAs operation.
Step 16	**exit** **Example:** `Device(config)# exit`	Exits global configuration submode and returns to privileged EXEC mode.

Troubleshooting Tips

Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an individual IP SLAs LSP ping or LSP traceroute operation.

What to Do Next

To display the results of an individual IP SLAs operation use the **show ip sla statistics** and **show ip sla statistics aggregated** commands. Checking the output for fields that correspond to criteria in your service level agreement will help you determine whether the service metrics are acceptable.

Verifying and Troubleshooting LSP Health Monitor Operations

SUMMARY STEPS

1. **debug ip sla error** [*operation-number*]
2. **debug ip sla mpls-lsp-monitor** [*operation-number*]
3. **debug ip sla trace** [*operation-number*]
4. **show ip sla mpls-lsp-monitor collection-statistics** [*group-id*]
5. **show ip sla mpls-lsp-monitor configuration** [*operation-number*]
6. **show ip sla mpls-lsp-monitor lpd operational-state** [*group-id*]
7. **show ip sla mpls-lsp-monitor neighbors**
8. **show ip sla mpls-lsp-monitor scan-queue** *operation-number*
9. **show ip sla mpls-lsp-monitor summary** [*operation-number* [**group** [*group-id*]]]
10. **show ip sla statistics** [*operation-number*] [**details**]
11. **show ip sla statistics aggregated** [*operation-number*] [**details**]
12. **show mpls discovery vpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**debug ip sla error** [*operation-number*] **Example:** `Device# debug ip sla error`	(Optional) Enables debugging output of IP SLAs operation run-time errors.
Step 2	**debug ip sla mpls-lsp-monitor** [*operation-number*] **Example:** `Device# debug ip sla mpls-lsp-monitor`	(Optional) Enables debugging output of LSP Health Monitor operations.
Step 3	**debug ip sla trace** [*operation-number*] **Example:** `Device# debug ip sla trace`	(Optional) Enables debugging output for tracing the execution of IP SLAs operations.
Step 4	**show ip sla mpls-lsp-monitor collection-statistics** [*group-id*] **Example:** `Device# show ip sla mpls-lsp-monitor collection-statistics 100001`	(Optional) Displays the statistics for IP SLAs operations belonging to an LSP discovery group of an LSP Health Monitor operation. **Note** This command is applicable only if the LSP discovery option is enabled.
Step 5	**show ip sla mpls-lsp-monitor configuration** [*operation-number*] **Example:** `Device# show ip sla mpls-lsp-monitor configuration 1`	(Optional) Displays configuration settings for LSP Health Monitor operations.

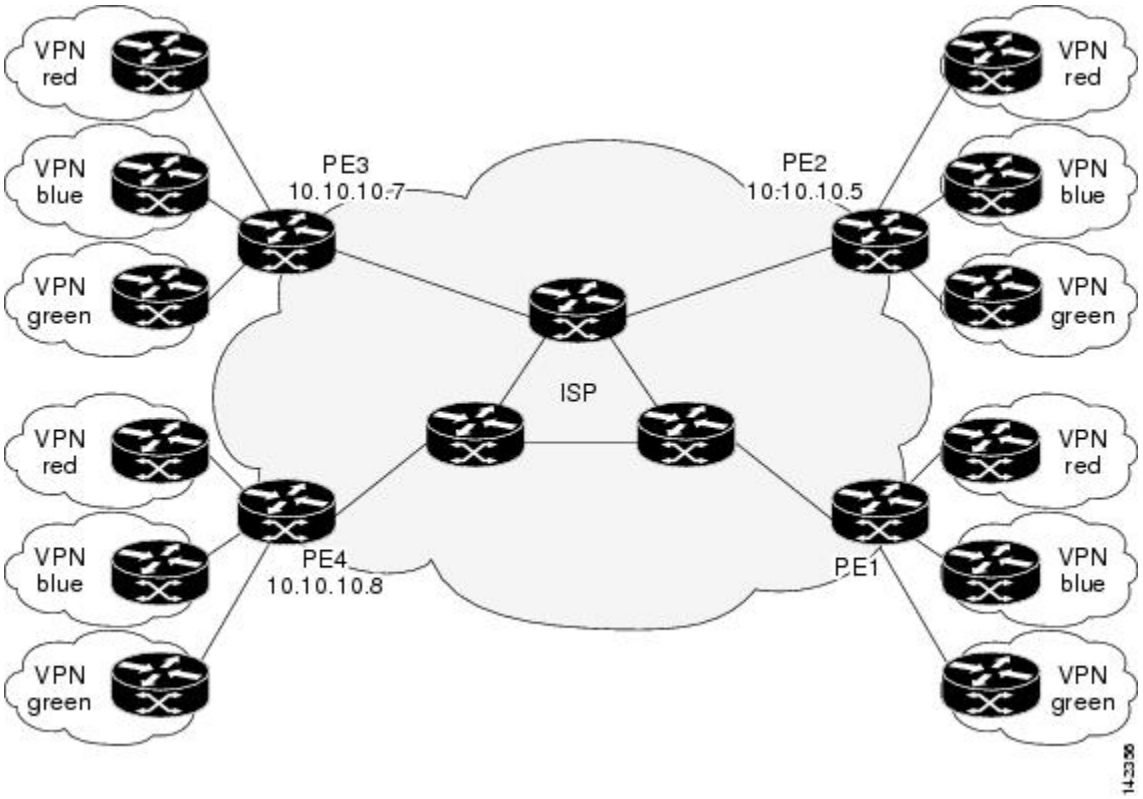
	Command or Action	Purpose
Step 6	**show ip sla mpls-lsp-monitor lpd operational-state** [*group-id*] **Example:** `Device# show ip sla mpls-lsp-monitor lpd operational-state 100001`	(Optional) Displays the operational status of the LSP discovery groups belonging to an LSP Health Monitor operation. **Note** This command is applicable only if the LSP discovery option is enabled.
Step 7	**show ip sla mpls-lsp-monitor neighbors** **Example:** `Device# show ip sla mpls-lsp-monitor neighbors`	(Optional) Displays routing and connectivity information about MPLS VPN BGP next hop neighbors discovered by the LSP Health Monitor operation.
Step 8	**show ip sla mpls-lsp-monitor scan-queue** *operation-number* **Example:** `Device# show ip sla mpls-lsp-monitor scan-queue 1`	(Optional) Displays information about adding or deleting BGP next hop neighbors from a particular MPLS VPN of an LSP Health Monitor operation.
Step 9	**show ip sla mpls-lsp-monitor summary** [*operation-number* [**group** [*group-id*]]] **Example:** `Device# show ip sla mpls-lsp-monitor summary`	(Optional) Displays BGP next hop neighbor and LSP discovery group information for LSP Health Monitor operations. **Note** This command is applicable only if the LSP discovery option is enabled.
Step 10	**show ip sla statistics** [*operation-number*] [**details**] **Example:** `Device# show ip sla statistics 100001`	(Optional) Displays the current operational status and statistics of all IP SLAs operations or a specified operation. **Note** This command applies only to manually configured IP SLAs operations.
Step 11	**show ip sla statistics aggregated** [*operation-number*] [**details**] **Example:** `Device# show ip sla statistics aggregated 100001`	(Optional) Displays the aggregated statistical errors and distribution information for all IP SLAs operations or a specified operation. **Note** This command applies only to manually configured IP SLAs operations.
Step 12	**show mpls discovery vpn** **Example:** `Device# show mpls discovery vpn`	(Optional) Displays routing information relating to the MPLS VPN BGP next hop neighbor discovery process.

Configuration Examples for LSP Health Monitors

Example Configuring and Verifying the LSP Health Monitor Without LSP Discovery

The figure below illustrates a simple VPN scenario for an ISP. This network consists of a core MPLS VPN with four PE devices belonging to three VPNs: red, blue, and green. From the perspective of device PE1, these VPNs are reachable remotely through BGP next hop devices PE2 (device ID: 10.10.10.5), PE3 (device ID: 10.10.10.7), and PE4 (device ID: 10.10.10.8).

Figure 41: Network Used for LSP Health Monitor Example



The following example shows how to configure operation parameters, proactive threshold monitoring, and scheduling options on PE1 (see the figure above) using the LSP Health Monitor. In this example, the LSP discovery option is enabled for LSP Health Monitor operation 1. Operation 1 is configured to automatically create IP SLAs LSP ping operations for all BGP next hop neighbors (PE2, PE3, and PE4) in use by all VRFs (red, blue, and green) associated with device PE1. The BGP next hop neighbor process is enabled, and the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor discovery database is set to 60 seconds. The time interval at which the LSP Health Monitor checks the scan queue for BGP next hop neighbor updates is set to 1 minute. The secondary frequency option is enabled for both connection loss and timeout events, and the secondary frequency is set to 10 seconds. As specified by the proactive threshold monitoring configuration, when three consecutive connection loss or timeout events

occur, an SNMP trap notification is sent. Multioperation scheduling and the generation of IP SLAs SNMP system logging messages are enabled.

PE1 Configuration

```
mpls discovery vpn interval 60
mpls discovery vpn next-hop
!
auto ip sla mpls-lsp-monitor 1
 type echo ipsla-vrf-all
 timeout 1000
 scan-interval 1
 secondary-frequency both 10
!
auto ip sla mpls-lsp-monitor reaction-configuration 1 react connectionLoss threshold-type
consecutive 3 action-type trapOnly
auto ip sla mpls-lsp-monitor reaction-configuration 1 react timeout threshold-type consecutive
 3 action-type trapOnly
ip sla traps
snmp-server enable traps rtr
!
auto ip sla mpls-lsp-monitor schedule 1 schedule-period 60 start-time now
```

The following is sample output from the **show ip sla mpls-lsp-monitor configuration** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor configuration 1
Entry Number : 1
Modification time   : *12:18:21.830 PDT Fri Aug 19 2005
Operation Type      : echo
Vrf Name            : ipsla-vrf-all
Tag                 :
EXP Value           : 0
Timeout(ms)         : 1000
Threshold(ms)       : 5000
Frequency(sec)      : Equals schedule period
LSP Selector        : 127.0.0.1
ScanInterval(min)   : 1
Delete Scan Factor  : 1
Operations List     : 100001-100003
Schedule Period(sec): 60
Request size        : 100
Start Time          : Start Time already passed
SNMP RowStatus      : Active
TTL value           : 255
Reply Mode          : ipv4
Reply Dscp Bits     :
Secondary Frequency : Enabled on Timeout
         Value(sec) : 10
Reaction Configs    :
    Reaction        : connectionLoss
    Threshold Type  : Consecutive
    Threshold Count : 3
    Action Type     : Trap Only
    Reaction        : timeout
    Threshold Type  : Consecutive
    Threshold Count : 3
    Action Type     : Trap Only
```

The following is sample output from the **show mpls discovery vpn** command for PE1:

```
PE1# show mpls discovery vpn
```

```
Refresh interval set to 60 seconds.
Next refresh in 46 seconds
Next hop 10.10.10.5 (Prefix: 10.10.10.5/32)
        in use by: red, blue, green
Next hop 10.10.10.7 (Prefix: 10.10.10.7/32)
        in use by: red, blue, green
Next hop 10.10.10.8 (Prefix: 10.10.10.8/32)
        in use by: red, blue, green
```

The following is sample output from the **show ip sla mpls-lsp-monitor neighbors** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor neighbors
IP SLA MPLS LSP Monitor Database : 1
BGP Next hop 10.10.10.5 (Prefix: 10.10.10.5/32)  OK
  ProbeID: 100001 (red, blue, green)
BGP Next hop 10.10.10.7 (Prefix: 10.10.10.7/32)  OK
  ProbeID: 100002 (red, blue, green)
BGP Next hop 10.10.10.8 (Prefix: 10.10.10.8/32)  OK
  ProbeID: 100003 (red, blue, green)
```

The following is sample output from the **show ip sla mpls-lsp-monitor scan-queue 1** and **debug ip sla mpls-lsp-monitor** commands when IP connectivity from PE1 to PE4 is lost. This output shows that connection loss to each of the VPNs associated with PE4 (red, blue, and green) was detected and that this information was added to the LSP Health Monitor scan queue. Also, since PE4 is no longer a valid BGP next hop neighbor, the IP SLAs operation for PE4 (Probe 10003) is being deleted.

```
PE1# show ip sla mpls-lsp-monitor scan-queue 1
Next scan Time after: 20 Secs
Next Delete scan Time after: 20 Secs
BGP Next hop    Prefix            vrf                               Add/Delete?
10.10.10.8      0.0.0.0/0         red                               Del(100003)
10.10.10.8      0.0.0.0/0         blue                              Del(100003)
10.10.10.8      0.0.0.0/0         green                             Del(100003)
PE1# debug ip sla mpls-lsp-monitor
IP SLAs MPLSLM debugging for all entries is on
*Aug 19 19:48: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in DeleteQ(1)
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing vrf red from tree entry 10.10.10.8
*Aug 19 19:56: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in DeleteQ(1)
*Aug 19 19:56: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in DeleteQ(1)
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing vrf blue from tree entry 10.10.10.8
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing vrf green from tree entry 10.10.10.8
*Aug 19 19:49: IP SLAs MPLSLM(1):Removing Probe 100003
```

The following is sample output from the **show ip sla mpls-lsp-monitor scan-queue 1** and **debug ip sla mpls-lsp-monitor** commands when IP connectivity from PE1 to PE4 is restored. This output shows that each of the VPNs associated with PE4 (red, blue, and green) were discovered and that this information was added to the LSP Health Monitor scan queue. Also, since PE4 is a newly discovered BGP next hop neighbor, a new IP SLAs operation for PE4 (Probe 100005) is being created and added to the LSP Health Monitor multioperation schedule. Even though PE4 belongs to three VPNs, only one IP SLAs operation is being created.

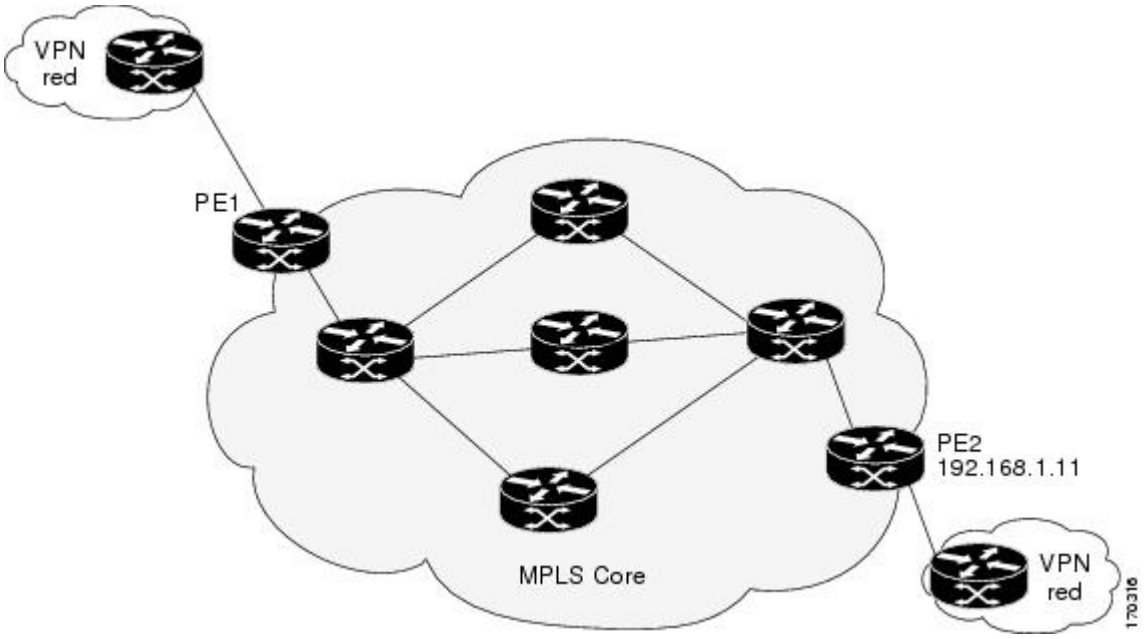
```
PE1# show ip sla mpls-lsp-monitor scan-queue 1
Next scan Time after: 23 Secs
Next Delete scan Time after: 23 Secs
BGP Next hop    Prefix            vrf                               Add/Delete?
10.10.10.8      10.10.10.8/32     red                               Add
10.10.10.8      10.10.10.8/32     blue                              Add
10.10.10.8      10.10.10.8/32     green                             Add
PE1# debug ip sla mpls-lsp-monitor
IP SLAs MPLSLM debugging for all entries is on
*Aug 19 19:59: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in AddQ
```

```
*Aug 19 19:59: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: IP SLAs MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding vrf red into tree entry 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding Probe 100005
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding ProbeID 100005 to tree entry 10.10.10.8 (1)
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding vrf blue into tree entry 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Duplicate in AddQ 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Adding vrf green into tree entry 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Duplicate in AddQ 10.10.10.8
*Aug 19 19:59: IP SLAs MPLSLM(1):Added Probe(s) 100005 will be scheduled after 26 secs over
 schedule period 60
```

Example Configuring and Verifying the LSP Health Monitor with LSP Discovery

The figure below illustrates a simple VPN scenario for an ISP. This network consists of a core MPLS VPN with two PE devices belonging to a VPN named red. From the perspective of device PE1, there are three equal-cost multipaths available to reach device PE2.

Figure 42: Network Used for LSP Health Monitor with LSP Discovery Example



The following example shows how to configure operation parameters, proactive threshold monitoring, and scheduling options on PE1 (see the figure above) using the LSP Health Monitor. In this example, the LSP discovery option is enabled for LSP Health Monitor operation 100. Operation 100 is configured to automatically create IP SLAs LSP ping operations for all equal-cost multipaths between PE1 and PE2. The BGP next hop neighbor process is enabled, and the time interval at which routing entries that are no longer valid are removed from the BGP next hop neighbor discovery database is set to 30 seconds. The time interval at which the LSP Health Monitor checks the scan queue for BGP next hop neighbor updates is set to 1 minute. The secondary frequency option is enabled for both connection loss and timeout events, and the secondary frequency is set to 5 seconds. The explicit null label option for echo request packets is enabled. The LSP rediscovery time period is set to 3 minutes. As specified by the proactive threshold monitoring configuration, an SNMP trap notification will be sent when an LSP discovery group status changes occurs. Multioperation scheduling and the generation of IP SLAs SNMP system logging messages are enabled.

PE1 Configuration

```
mpls discovery vpn next-hop
mpls discovery vpn interval 30
!
auto ip sla mpls-lsp-monitor 100
 type echo ipsla-vrf-all
 scan-interval 1
 secondary-frequency both 5
!
 path-discover
 force-explicit-null
 scan-period 3
!
auto ip sla mpls-lsp-monitor reaction-configuration 100 react lpd-group retry 3 action-type
 trapOnly
!
auto ip sla mpls-lsp-monitor schedule 100 schedule-period 30 start-time now
!
ip sla logging traps
snmp-server enable traps rtr
```

The following is sample output from the **show ip sla mpls-lsp-monitor configuration** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor configuration
Entry Number : 100
Modification time   : *21:50:16.411 GMT Tue Jun 20 2006
Operation Type      : echo
Vrf Name            : ipsla-vrf-all
Tag                 :
EXP Value           : 0
Timeout(ms)         : 5000
Threshold(ms)       : 50
Frequency(sec)      : Equals schedule period
ScanInterval(min)   : 1
Delete Scan Factor  : 1
Operations List     : 100002
Schedule Period(sec): 30
Request size        : 100
Start Time          : Start Time already passed
SNMP RowStatus      : Active
TTL value           : 255
Reply Mode          : ipv4
Reply Dscp Bits     :
Path Discover       : Enable
    Maximum sessions          : 1
    Session Timeout(seconds)  : 120
    Base LSP Selector         : 127.0.0.0
    Echo Timeout(seconds)     : 5
    Send Interval(msec)       : 0
    Label Shimming Mode       : force-explicit-null
    Number of Stats Hours     : 2
    Scan Period(minutes)      : 3
Secondary Frequency : Enabled on Connection Loss and Timeout
         Value(sec) : 5
Reaction Configs    :
    Reaction        : Lpd Group
    Retry Number    : 3
    Action Type     : Trap Only
```

The following is sample output from the **show mpls discovery vpn** command for PE1:

```
PE1# show mpls discovery vpn
Refresh interval set to 30 seconds.
Next refresh in 4 seconds
Next hop 192.168.1.11 (Prefix: 192.168.1.11/32)
        in use by: red
```

The following is sample output from the **show ip sla mpls-lsp-monitor neighbors** command for PE1:

```
PE1# show ip sla mpls-lsp-monitor neighbors
IP SLA MPLS LSP Monitor Database : 100
BGP Next hop 192.168.1.11 (Prefix: 192.168.1.11/32)  OK Paths: 3
  ProbeID: 100001 (red)
```

The following is sample output from the **show ip sla mpls-lsp-monitor lpd operational-state** command for LSP discovery group 100001:

```
PE1# show ip sla mpls-lsp-monitor lpd operational-state
Entry number: 100001
MPLSLM Entry Number: 100
Target FEC Type: LDP IPv4 prefix
Target Address: 192.168.1.11
Number of Statistic Hours Kept: 2
Last time LPD Stats were reset: *21:21:18.239 GMT Tue Jun 20 2006
Traps Type: 3
Latest Path Discovery Mode: rediscovery complete
Latest Path Discovery Start Time: *21:59:04.475 GMT Tue Jun 20 2006
Latest Path Discovery Return Code: OK
Latest Path Discovery Completion Time(ms): 3092
Number of Paths Discovered: 3
Path Information :
Path   Outgoing   Lsp             Link  Conn  Adj             Downstream
Index  Interface  Selector        Type  Id    Addr            Label Stack     Status
1      Et0/0      127.0.0.8       90    0     10.10.18.30     21              OK
2      Et0/0      127.0.0.2       90    0     10.10.18.30     21              OK
3      Et0/0      127.0.0.1       90    0     10.10.18.30     21              OK
```

The following is sample output from the **show ip sla mpls-lsp-monitor collection-statistics** command for LSP discovery group 100001:

```
PE1# show ip sla mpls-lsp-monitor collection-statistics
Entry number: 100001
Start Time Index: *21:52:59.795 GMT Tue Jun 20 2006
Path Discovery Start Time: *22:08:04.507 GMT Tue Jun 20 2006
Target Destination IP address: 192.168.1.11
Path Discovery Status: OK
Path Discovery Completion Time: 3052
Path Discovery Minimum Paths: 3
Path Discovery Maximum Paths: 3
LSP Group Index: 100002
LSP Group Status: up
Total Pass: 36
Total Timeout: 0        Total Fail: 0
Latest Probe Status: 'up,up,up'
Latest Path Identifier: '127.0.0.8-Et0/0-21,127.0.0.2-Et0/0-21,127.0.0.1-Et0/0-21'
Minimum RTT: 280        Maximum RTT: 324        Average RTT: 290
```

The following is sample output from the **show ip sla mpls-lsp-monitor summary** command for LSP Health Monitor operation 100:

```
PE1# show ip sla mpls-lsp-monitor summary 100
```

```
Index                    -  MPLS LSP Monitor probe index
Destination              -  Target IP address of the BGP next hop
Status                   -  LPD group status
LPD Group ID             -  Unique index to identify the LPD group
Last Operation Time      -  Last time an operation was attempted by
                            a particular probe in the LPD Group
Index  Destination     Status     LPD Group ID    Last Operation Time
100    192.168.1.11    up         100001          *22:20:29.471 GMT Tue Jun 20 2006
```

The following is sample output from the **show ip sla mpls-lsp-monitor summary** command for LSP discovery group 100001:

```
PE1#show ip sla mpls-lsp-monitor summary 100 group 100001
Group ID                 -  unique number to identify a LPD group
Lsp-selector             -  Unique 127/8 address used to identify a LPD
Last Operation status    -  Latest probe status
Last RTT                 -  Latest Round Trip Time
Last Operation Time      -  Time when the last operation was attempted
Group ID  Lsp-Selector    Status     Failures    Successes   RTT   Last Operation Time
100001    127.0.0.8       up         0           55          320   *22:20:29.471 GMT Tue
Jun 20 2006
100001    127.0.0.2       up         0           55          376   *22:20:29.851 GMT Tue
Jun 20 2006
100001    127.0.0.1       up         0           55          300   *22:20:30.531 GMT Tue
Jun 20 2006
```

Example Manually Configuring an IP SLAs LSP Ping Operation

The following example shows how to manually configure and schedule an IP SLAs LSP ping operation:

```
ip sla 1
mpls lsp ping ipv4 192.168.1.4 255.255.255.255 lsp-selector 127.1.1.1
frequency 120
secondary-frequency timeout 30
!
ip sla reaction-configuration 1 react connectionLoss threshold-type consecutive 3 action-type
 trapOnly
ip sla reaction-configuration 1 react timeout threshold-type consecutive 3 action-type
trapOnly
ip sla logging traps
!
ip sla schedule 1 start-time now life forever
```

Additional References

Related Documents

Related Topic	Document Title
MPLS LSP discovery management tool	"MPLS EM-MPLS LSP Multipath Tree Trace" chapter of the *Multiprotocol Label Switching Configuration Guide*
Configuring standard IP access lists	"Access Control Lists" chapter of the *Security Configuration Guide: Securing the Data Plane* guide

Related Topic	Document Title
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" chapter of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	" Configuring Proactive Threshold Monitoring of IP SLAs Operations" chapter of the *Cisco IOS IP SLAs Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards

Standard	Title
draft-ietf-mpls-lsp-ping-09.txt	Detecting MPLS Data Plane Failures
draft-ietf-mpls-oam-frmwk-03.txt	A Framework for MPLS Operations and Management (OAM)
draft-ietf-mpls-oam-requirements-06.txt	OAM Requirements for MPLS Networks

MIBs

MIB	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

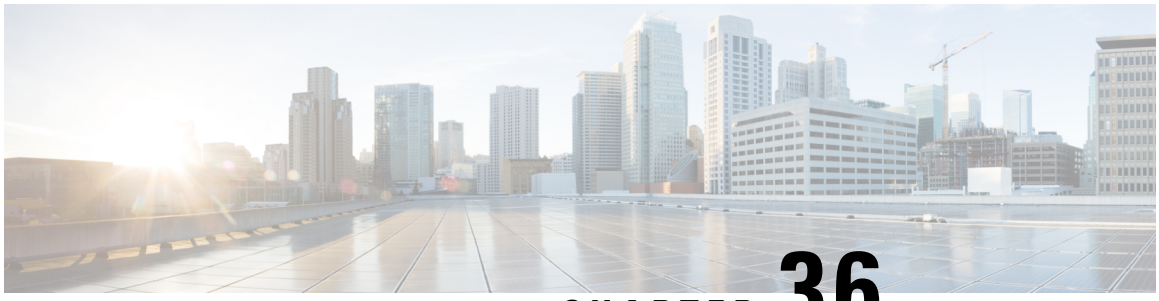
Feature Information for LSP Health Monitor Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 58: Feature Information for the LSP Health Monitor

Feature Name	Releases	Feature Information
IP SLAs--LSP Health Monitor		The IP SLAs LSP Health Monitor feature provides the capability to proactively monitor Layer 3 MPLS VPNs.
IP SLAs--LSP Health Monitor		For software releases in which this feature was already introduced, new command-line interface (CLI) was implemented that replaces the CLI introduced in the earlier releases
IP SLAs--LSP Health Monitor with LSP Discovery		The LSP discovery capability was added.



CHAPTER 36

IP SLAs for MPLS Psuedo Wire via VCCV

This module describes how to configure IP Service Level Agreements (SLAs) for MPLS Pseudo Wire (PWE3) via Virtual Circuit Connectivity Verification (VCCV) to schedule pseudo-wire ping operations and provide monitoring and alerts for round trip time (RTT), failure, and connection threshold violations via SNMP Traps.

Restrictions for IP SLAs for MPLS Pseudo Wire via VCCV

LSP discovery is not supported for IP SLAs VCCV operations.

Information About IP SLAs for MPLS Pseudo Wire via VCCV

IP SLAs VCCV Operation

The IP SLAs VCCV operation supports Virtual Circuit Connectivity Verification (VCCV) for Pseudo-Wire Emulation Edge-to-Edge (PWE3) services across MPLS networks. The IP SLAs VCCV operation type is based on the **ping mpls pseudowire** command, which checks MPLS LSP connectivity across an Any Transport over MPLS (AToM) virtual circuit (VC) by sending a series of pseudo-wire ping operations to the specified destination PE router.

When MPLS LSP connectivity checking is performed through an IP SLAs VCCV operation (rather than through the **ping mpls** command with the **pseudowire** keyword), you can use the IP SLA proactive threshold monitoring and multioperation scheduling capabilities:

The LSP discovery option does not support the IP SLAs VCCV operation.

Proactive Threshold Monitoring for the LSP Health Monitor

Proactive threshold monitoring support for the LSP Health Monitor feature provides the capability for triggering SNMP trap notifications and syslog messages when user-defined reaction conditions (such as a connection loss or timeout) are met. Configuring threshold monitoring for an LSP Health Monitor operation is similar to configuring threshold monitoring for a standard IP SLAs operation.

LSP Discovery Option Enabled

If the LSP discovery option for an LSP Health Monitor operation is enabled, SNMP trap notifications can be generated when one of the following events occurs:

- LSP discovery for a particular BGP next hop neighbor fails.
- Operational status of an LSP discovery group changes.

Possible reasons for which LSP discovery can fail for a particular BGP next hop neighbor are as follows:

- Expiration of time allowed for a BGP next hop neighbor to respond to an LSP discovery request.
- Return code is “Broken” or “Unexplorable” for all paths leading to the BGP next hop neighbor.

The table below describes the conditions for which the operational status of an LSP discovery group can change. Whenever an individual IP SLAs LSP ping operation of an LSP discovery group is executed, a return code is generated. Depending on the value of the return code and the current status of the LSP discovery group, the group status can change.

Table 59: Conditions for Which an LSP Discovery Group Status Changes

Individual IP SLAs Operation Return Code	Current Group Status = UP	Current Group Status = PARTIAL	Current Group Status = DOWN
OK	No group status change.	If return codes for all paths in the group are OK, then the group status changes to UP.	Group status changes to PARTIAL.
Broken or Unexplorable	Group status changes to PARTIAL.	If return codes for all paths in the group are Broken or Unexplorable, then the group status changes to DOWN.	No group status change.

The return code for an individual IP SLAs LSP ping operation can be one of the following:

- OK--Indicates that the LSP is working properly. The customer VPN traffic will be sent across this path.
- Broken--Indicates that the LSP is broken. Customer VPN traffic will not be sent across this path and may be discarded.
- Unexplorable--Indicates that not all the paths to this PE neighbor have been discovered. This may be due to a disruption along the LSP or because the number of 127/8 IP addresses used for LSP selection has been exhausted.

The status of an LSP discovery group can be one of the following:

- UNKNOWN--Indicates that group status has not yet been determined and that the paths belonging to the group are in the process of being tested for the first time. Once this initial test is complete, the group status will change to UP, PARTIAL, or DOWN.
- UP--Indicates that all the paths within the group are active and no operation failures have been detected.
- PARTIAL--Indicates that an operation failure has been detected for one or more, but not all, of the paths within the group.
- DOWN--Indicates that an operation failure has been detected for all the paths within the group.

Secondary Frequency Option

With the introduction of the LSP Health Monitor feature, a new threshold monitoring parameter has been added that allows you to specify a secondary frequency. If the secondary frequency option is configured and a failure (such as a connection loss or timeout) is detected for a particular path, the frequency at which the path is remeasured will increase to the secondary frequency value (testing at a faster rate). When the configured reaction condition is met (such as N consecutive connection losses or N consecutive timeouts), an SNMP trap and syslog message can be sent and the measurement frequency will return to its original frequency value.

How to Configure IP SLAs for MPLS Pseudo Wire via VCCM

Manually Configuring and Scheduling an IP SLAs VCCV Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **mpls lsp ping pseudowire** *peer-ipaddr vc-id* [**source-ipaddr** *source-ipaddr*]
5. **exp** *exp-bits*
6. **frequency** *seconds*
7. **request-data-size** *bytes*
8. **secondary-frequency** {**both** | **connection-loss** | **timeout**} *frequency*
9. **tag** *text*
10. **threshold** *milliseconds*
11. **timeout** *milliseconds*
12. **exit**
13. **ip sla reaction-configuration** *operation-number* [**react** *monitored-element*] [**threshold-type** {**never** | **immediate** | **consecutive** [*consecutive-occurrences*] | **xofy** [*x-value y-value*] | **average** [*number-of-probes*]}] [**threshold-value** *upper-threshold lower-threshold*] [**action-type** {**none** | **trapOnly** | **triggerOnly** | **trapAndTrigger**}]
14. **ip sla logging traps**
15. **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**]
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Router(config)# ip sla 777`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**mpls lsp ping pseudowire** *peer-ipaddr vc-id* [**source-ipaddr** *source-ipaddr*] **Example:** `Router(config-ip-sla)# mpls lsp ping pseudowire 192.168.1.103 123 source-ipaddr 192.168.1.102`	Configures the IP SLAs operation as an LSP pseudo-wire ping and enters VCCV configuration mode.
Step 5	**exp** *exp-bits* **Example:** **Example:** `Router(config-sla-vccv)# exp 5`	(Optional) Specifies the experimental field value in the header for an echo request packet of an IP SLAs operation.
Step 6	**frequency** *seconds* **Example:** `Router(config-sla-vccv)# frequency 120`	(Optional) Specifies the rate at which a specified IP SLAs operation repeats.
Step 7	**request-data-size** *bytes* **Example:** `Router(config-sla-vccv)# request-data-size 200`	(Optional) Specifies the protocol data size for a request packet of an IP SLAs operation.
Step 8	**secondary-frequency** {**both** | **connection-loss** | **timeout**} *frequency* **Example:** `Router(config-sla-vccv)# secondary-frequency connection-loss 10`	(Optional) Sets the faster measurement frequency (secondary frequency) to which an IP SLAs operation should change when a reaction condition occurs.

	Command or Action	Purpose
Step 9	**tag** *text* **Example:** `Router(config-sla-vccv)# tag testgroup`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 10	**threshold** *milliseconds* **Example:** **Example:** `Router(config-sla-vccv)# threshold 6000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 11	**timeout** *milliseconds* **Example:** `Router(config-sla-vccv)# timeout 7000`	(Optional) Specifies the amount of time the IP SLAs operation waits for a response from its request packet.
Step 12	**exit** **Example:** `Router(config-sla-vccv)# exit`	Exits VCCV configuration mode and returns to global configuration mode.
Step 13	**ip sla reaction-configuration** *operation-number* [**react** *monitored-element*] [**threshold-type** {**never** | **immediate** | **consecutive** [*consecutive-occurrences*] | **xofy** [*x-value y-value*] | **average** [*number-of-probes*]}] [**threshold-value** *upper-threshold lower-threshold*] [**action-type** {**none** | **trapOnly** | **triggerOnly** | **trapAndTrigger**}] **Example:** `Router(config)# ip sla reaction-configuration 777 react connectionLoss threshold-type consecutive 3 action-type traponly`	(Optional) Configures certain actions to occur based on events under the control of Cisco IOS IP SLAs.
Step 14	**ip sla logging traps** **Example:** `Router(config)# ip sla logging traps`	(Optional) Enables the generation of SNMP system logging messages specific to IP SLAs trap notifications.
Step 15	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**] **Example:** `Router(config)# ip sla schedule 777 life forever start-time now`	Configures the scheduling parameters for an IP SLAs operation.

	Command or Action	Purpose
Step 16	**exit** **Example:** `Router(config)# exit`	Exits global configuration submode and returns to privileged EXEC mode.

Troubleshooting Tips

Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an individual IP SLAs PWE3 service via VCCV operation.

What to Do Next

To display the results of an individual IP SLAs operation use the **show ip sla statistics** and **show ip sla statistics aggregated** commands. Checking the output for fields that correspond to criteria in your service level agreement will help you determine whether the service metrics are acceptable.

Configuration Examples for IP SLAs for MPLS Pseudo Wire via VCCM

Example Manually Configuring an IP SLAs VCCV Operation

The following example shows how to manually configure an IP SLAs VCCV operation in conjunction with the proactive threshold monitoring and multioperation scheduling capabilities of the LSP Health Monitor.

In this example, a VC with the identifier 123 has already been established between the PE device and its peer at IP address 192.168.1.103.

IP SLAs VCCV operation 777 is configured with operation parameters and reaction conditions, and it is scheduled to begin immediately and run indefinitely.

```
ip sla 777
 mpls lsp ping pseudowire 192.168.1.103 123
  exp 5
  frequency 120
  secondary-frequency timeout 30
  tag testgroup
  threshold 6000
  timeout 7000
  exit
!
 ip sla reaction-configuration 777 react rtt threshold-value 6000 3000 threshold-type
immediate 3 action-type traponly
 ip sla reaction-configuration 777 react connectionLoss threshold-type immediate action-type
 traponly
 ip sla reaction-configuration 777 react timeout threshold-type consecutive 3 action-type
traponly
 ip sla logging traps
!
 ip sla schedule 777 life forever start-time now
 exit
```

RTT Thresholds

The **threshold** command configures 6000 milliseconds as the amount of time for a rising threshold to be declared on the monitored pseudo-wire. The first **ip sla reaction-configuration** command specifies that an SNMP logging trap is to be sent immediately if the round-trip time violates the upper threshold of 6000 milliseconds or the lower threshold of 3000 milliseconds.

Connection Loss

The second **ip sla reaction-configuration** command specifies that an SNMP logging trap is to be sent immediately if a connection loss occurs for the monitored pseudo-wire.

Response Timeout

The **timeout** command configures 7000 seconds as the amount of time that VCCV operation 777 waits for a response from its request packet before a timeout is declared. The **secondary-frequency** command specifies that, if a timeout occurs, the measurement frequency of the operation repeats is to be increased from 120 seconds (the initial measurement frequency specified using the **frequency** command) to a faster rate of 30 seconds. The third **ip sla reaction-configuration** command specifies that an SNMP logging trap is to be sent if three consecutive timeouts occur.

Additional References

Related Documents

Related Topic	Document Title
MPLS LSP discovery management tool	"MPLS EM-MPLS LSP Multipath Tree Trace" chapter of the *Multiprotocol Label Switching Configuration Guide*
Configuring standard IP access lists	"Access Control Lists" chapter of the *Security Configuration Guide: Securing the Data Plane* guide
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" chapter of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	" Configuring Proactive Threshold Monitoring of IP SLAs Operations" chapter of the *Cisco IOS IP SLAs Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards

Standard	Title
draft-ietf-mpls-lsp-ping-09.txt	Detecting MPLS Data Plane Failures
draft-ietf-mpls-oam-frmwk-03.txt	A Framework for MPLS Operations and Management (OAM)

Standard	Title
draft-ietf-mpls-oam-requirements-06.txt	OAM Requirements for MPLS Networks

MIBs

MIB	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

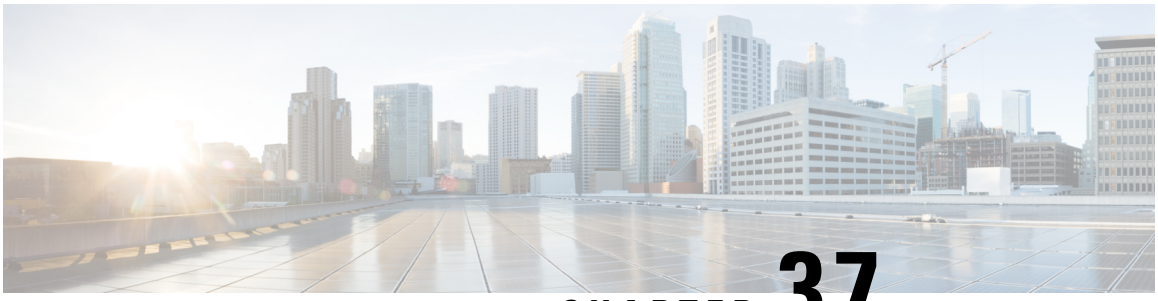
Feature Information for IP SLAs for MPLS PWE3 via VCCM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 60: IP SLAs for MPLS PWE3 via VCCM

Feature Name	Releases	Feature Information
IP SLAs for MPLS Pseudo Wire (PWE3) via VCCM	12(33)SB 12.2(33)SRC 15.0(1)S Cisco IOS XE 3.1.0SG	The IP SLAs VCCV operation was added to support Virtual Circuit Connectivity Verification (VCCV) for Pseudo-Wire Emulation Edge-to-Edge (PWE3) services across MPLS networks.



CHAPTER 37

Configuring IP SLAs for Metro-Ethernet

This module describes how to configure an IP Service Level Agreements (SLAs) for Metro-Ethernet to gather network performance metrics in service-provider Ethernet networks. Available statistical measurements for the IP SLAs Ethernet operation include round-trip time, jitter (interpacket delay variance), and packet loss.

Prerequisites for IP SLAs for Metro-Ethernet

It is recommended that the IEEE 802.1ag standard is supported on the destination devices in order to obtain complete error reporting and diagnostics information.

Restrictions for IP SLAs for Metro-Ethernet

- Memory and performance may be impacted for a given Ethernet CFM maintenance domain and Ethernet Virtual Circuit (EVC) or VLAN that has a large number of maintenance endpoints (MEPs).
- In case of PW redundancy, we need to have 2 different CFM/Y1731 sessions on active and backup PW. We cannot expect the same mpid and Y1731 session to work after PW switchover.
- Y1731 is not supported for port meps.
- CFM ans Y1731 is not supported for vpls cases, untagged EFP as well.

Information About IP SLAs for Metro-Ethernet

IP SLAs Ethernet Operation Basics

The IP SLAs for Metro-Ethernet integrates IP SLAs with the Ethernet Connectivity Fault Management (CFM) feature. Ethernet CFM is an end-to-end per-service-instance Ethernet-layer operation, administration, and management (OAM) protocol.

The IP SLAs for Metro-Ethernet feature provides the capability to gather statistical measurements by sending and receiving Ethernet data frames between Ethernet CFM maintenance endpoints (MEPs). The performance metrics for IP SLAs Ethernet operations are measured between a source MEP and a destination MEP. Unlike existing IP SLAs operations that provide performance metrics for the IP layer, the IP SLAs Ethernet operation provides performance metrics for Layer 2.

IP SLAs Ethernet operations may be configured using the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

You can manually configure individual Ethernet ping or Ethernet jitter operations by specifying the destination MEP identification number, name of the maintenance domain, and EVC or VLAN identifier or port level option.

You also have the option to configure an IP SLAs auto Ethernet operation (ping or jitter) that will query the Ethernet CFM database for all maintenance endpoints in a given maintenance domain and EVC or VLAN. When an IP SLAs auto Ethernet operation is configured, individual Ethernet ping or Ethernet jitter operations are automatically created based on the MEPs that were discovered. A notification mechanism exists between the IP SLAs and Ethernet CFM subsystems to facilitate the automatic creation of Ethernet ping or Ethernet jitter operations for applicable MEPs that are added to a given maintenance domain and EVC or VLAN while an auto Ethernet operation is running.

The IP SLAs for Metro-Ethernet feature supports multioperation scheduling of IP SLAs operations and proactive threshold violation monitoring through SNMP trap notifications and syslog messages.

Statistics Measured by the IP SLAs Ethernet Operation

The network performance metrics supported by the IP SLAs Ethernet operation is similar to the metrics supported by existing IP SLAs operations. The statistical measurements supported by the IP SLAs Ethernet jitter operation include the following:

- Round-trip time latency
- Unprocessed packets
- Packet loss (source-to-destination and destination-to-source)
- Out-of-sequence, tail-dropped, and late packets

How to Configure IP SLAs for Metro-Ethernet

Note There is no need to configure an IP SLAs responder on the destination device.

Configuring an IP SLAs Auto Ethernet Operation with Endpoint Discovery on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla ethernet-monitor** *operation-number*
4. **type echo domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} [**exclude-mpids** *mp-ids*]
5. **cos** *cos-value*
6. **owner** *owner-id*
7. **request-data-size** *bytes*
8. **tag** *text*
9. **threshold** *milliseconds*
10. **timeout** *milliseconds*
11. **end**
12. **show ip sla ethernet-monitor configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla ethernet-monitor** *operation-number* **Example:** `Device(config)# ip sla ethernet-monitor 1`	Begins configuration for an IP SLAs auto Ethernet operation and enters IP SLA Ethernet monitor configuration mode.
Step 4	**type echo domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} [**exclude-mpids** *mp-ids*]	• **domain** *domain-name*—Specify the name of the created domain.

	Command or Action	Purpose
	Example: `Device(config-ip-sla-ethernet-monitor)# type echo domain testdomain vlan 34`	• **vlan***vlan-id*—Enter the service provider VLAN ID or IDs as a VLAN-ID (1 to 4094), a range of VLAN-IDs separated by a hyphen, or a series of VLAN IDs separated by comma. • **exclude-mpids***mp-ids*—Enter a maintenance end point identifier (mpid). The identifier must be unique for each VLAN (service instance). The range is 1 to 8191. For Echo operations only: Configures an auto Ethernet operation for Ethernet ping operations. **Note** Depending on your release, the **evc** *evc-id* keyword and argument combination may not be available for this command.
Step 5	**cos** *cos-value* **Example:** `Device(config-ip-sla-ethernet-params)# cos 2`	(Optional) Sets the class of service for an IP SLAs Ethernet operation.
Step 6	**owner** *owner-id* **Example:** `Device(config-ip-sla-ethernet-params)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 7	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-ethernet-params)# request-data-size 64`	(Optional) Sets the padding size for the data frame of an IP SLAs Ethernet operation. • The default value for IP SLAs Ethernet ping operations is 66 bytes. • The default value for IP SLAs Ethernet jitter operations is 51 bytes.
Step 8	**tag** *text* **Example:** `Device(config-ip-sla-ethernet-params)# tag TelnetPollSever1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 9	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-ethernet-params)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 10	**timeout** *milliseconds* **Example:**	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

	Command or Action	Purpose
	`Device(config-ip-sla-ethernet-params)# timeout 10000`	
Step 11	**end** **Example:** `Device(config-ip-sla-ethernet-params)# end`	Exits to privileged EXEC configuration mode.
Step 12	**show ip sla ethernet-monitor configuration** [*operation-number*] **Example:** `Device# show ip sla ethernet-monitor configuration 1`	(Optional) Displays configuration settings for all IP SLAs auto Ethernet operations or a specified auto Ethernet operation.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Manually Configuring an IP SLAs Ethernet Ping or Jitter Operation on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ethernet echo mpid** *mp-id* **domain** *domain-name* {**evc** *evc-id* | **port** | **vlan** *vlan-id*}
5. **ethernet jitter mpid** *mp-id* **domain** *domain-name* {**evc** *evc-id* | **port** | **vlan** *vlan-id*} [**interval** *interframe-interval*] [**num-frames** *frames-number*]
6. **cos** *cos-value*
7. **frequency** *seconds*
8. **history** *history-parameter*
9. **owner** *owner-id*
10. **request-data-size** *bytes*
11. **tag** *text*
12. **threshold** *milliseconds*
13. **timeout** *milliseconds*
14. **end**
15. **show ip sla configuration** [*operation-number*]
16. **show ip sla application**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 1`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ethernet echo mpid** *mp-id* **domain** *domain-name* {**evc** *evc-id* | **port** | **vlan** *vlan-id*} **Example:** `Device(config-ip-sla)# ethernet echo mpid 23 domain testdomain vlan 34`	For a ping operation only: Configures the IP SLAs operation as an Ethernet ping operation and enters Ethernet echo configuration mode. **Note** Depending on your release, the **evc** evc-id keyword and argument combination may not be available for this command.
Step 5	**ethernet jitter mpid** *mp-id* **domain** *domain-name* {**evc** *evc-id* | **port** | **vlan** *vlan-id*} [**interval** *interframe-interval*] [**num-frames** *frames-number*] **Example:** `Device(config-ip-sla)# ethernet jitter mpid 23 domain testdomain evc testevc interval 20 num-frames 30`	For a jitter operation only: Configures the IP SLAs operation as an Ethernet jitter operation and enters Ethernet jitter configuration mode. **Note** Depending on your release, the **evc** evc-id keyword and argument combination may not be available for this command.
Step 6	**cos** *cos-value* **Example:** `Device(config-ip-sla-ethernet-echo)# cos 2`	(Optional) Sets the class of service for an IP SLAs Ethernet operation. **Note** For this and the remaining steps, the configuration mode shown in the example is for configuring an Ethernet echo operation. However, the commands are the same in the Ethernet jitter configuration mode.
Step 7	**frequency** *seconds* **Example:** `Device(config-ip-sla-ethernet-echo)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.

	Command or Action	Purpose
Step 8	**history** *history-parameter* **Example:** `Device(config-ip-sla-ethernet-echo)# history hours-of-statistics-kept 3`	(Optional) Specifies the parameters used for gathering statistical history information for an IP SLAs operation.
Step 9	**owner** *owner-id* **Example:** `Device(config-ip-sla-ethernet-echo)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 10	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-ethernet-echo)# request-data-size 64`	(Optional) Sets the padding size for the data frame of an IP SLAs Ethernet operation. The default value for IP SLAs Ethernet ping operations is 66 bytes. The default value for IP SLAs Ethernet jitter operations is 51 bytes.
Step 11	**tag** *text* **Example:** `Device(config-ip-sla-ethernet-echo)# tag TelnetPollSever1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 12	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-ethernet-echo)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 13	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-ethernet-echo)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 14	**end** **Example:** `Device(config-ip-sla-ethernet-echo)# end`	Exits to privileged EXEC mode.
Step 15	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 1`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.
Step 16	**show ip sla application** **Example:** `Device# show ip sla application`	(Optional) Displays global information about supported IP SLAs features.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Scheduling IP SLAs Operations

Note

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in an operation group must be the same unless you are enabling the random scheduler option for a multioperation scheduler.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip sla ethernet-monitor schedule** *operation-number* **schedule-period** *seconds* [**frequency** [*seconds*]] [**start-time** {**after** *hh* **:** *mm* **:** *ss* | *hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **now** | **pending**}]
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life**{**forever** | *seconds*}] [**start-time**{*hh:mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}]
4. **exit**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **ip sla ethernet-monitor schedule** *operation-number* **schedule-period** *seconds* [**frequency** [*seconds*]] [**start-time** {**after** *hh* **:** *mm* **:** *ss*	*hh* **:** *mm*[**:** *ss*] [*month day*	*day month*]	**now**	**pending**}]	• The first example shows how to configure scheduling parameters for an IP SLAs auto Ethernet operation. • The second example shows how to configure the scheduling parameters for an individual IP SLAs operation.

	Command or Action	Purpose

	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life**{**forever** | *seconds*}] [**start-time**{*hh:mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}]
	Example:
	`Device(config)# ip sla ethernet-monitor schedule 10 schedule-period 60 start-time now`
	`Device(config)# ip sla schedule 1 start-time now life forever`
	`Device(config)# ip sla group schedule 1 3,4,6-9`
		• The third example shows how to specifiy an IP SLAs operation group number and range of operation numbers to be scheduled for a multioperation scheduler.

Step 4	**exit** **Example:** `Device(config)# exit`	Exits to the privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays the IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays the IP SLAs configuration details.

Troubleshooting Tips

Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an individual IP SLAs Ethernet ping or Ethernet jitter operation. Use the **debug ip sla ethernet-monitor** command to help troubleshoot issues with an IP SLAs auto Ethernet operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP SLAs operation, see the “Configuring Proactive Threshold Monitoring” section.

operation)

To display and interpret the results of an IP SLAs operation, use the **show ip sla statistics** command. Check the output for fields that correspond to criteria in your service level agreement to determine whether the service metrics are acceptable.

Configuration Examples for IP SLAs for Metro-Ethernet

Example IP SLAs Auto Ethernet Operation with Endpoint Discovery

The following examples shows the operation parameters, proactive threshold monitoring, and scheduling options for an IP SLAs auto Ethernet operation. In Configuration A, operation 10 is configured to automatically create IP SLAs Ethernet ping operations for all the discovered maintenance endpoints in the domain named testdomain and VLAN identification number 34. In Configuration B, operation 20 is configured to automatically create IP SLAs Ethernet ping operations for all the discovered maintenance endpoints in the domain named testdomain and EVC identified as testevc. In both configurations, the proactive threshold monitoring configuration specifies that when three consecutive connection loss events occur, an SNMP trap notification should be sent. The schedule period for operation 10 and operation 20 is 60 seconds, and both operations are scheduled to start immediately.

Configuration A

```
ip sla ethernet-monitor 10
 type echo domain testdomain vlan 34
!
ip sla ethernet-monitor reaction-configuration 10 react connectionLoss threshold-type
consecutive 3 action-type trapOnly
!
ip sla ethernet-monitor schedule 10 schedule-period 60 start-time now
```

Configuration B

```
ip sla ethernet-monitor 20
 type echo domain testdomain evc testevc
!
ip sla ethernet-monitor reaction-configuration 20 react connectionLoss threshold-type
consecutive 3 action-type trapOnly
!
ip sla ethernet-monitor schedule 20 schedule-period 60 start-time now
```

Example Individual IP SLAs Ethernet Ping Operation

The following example show the configuration for an IP SLAs Ethernet ping operation. In Configuration C, the maintenance endpoint identification number is 23, the maintenance domain name is testdomain, and the VLAN identification number is 34. In Configuration D, the maintenance endpoint identification number is 23, the maintenance domain name is testdomain, and the EVC is identified as testevc. In both configurations, the proactive threshold monitoring configuration specifies that when three consecutive connection loss events occur, an SNMP trap notification should be sent. Operation 1 and operation 5 are scheduled to start immediately.

Configuration C

```
ip sla 1
 ethernet echo mpid 23 domain testdomain vlan 34
!
ip sla reaction-configuration 1 react connectionLoss threshold-type consecutive 3 action-type
 trapOnly
!
ip sla schedule 1 start-time now
```

Configuration D

```
ip sla 5
 ethernet echo mpid 23 domain testdomain evc testevc
!
ip sla reaction-configuration 5 react connectionLoss threshold-type consecutive 3 action-type
 trapOnly
!
ip sla schedule 5 start-time now
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference, All Releases
Cisco IOS IP SLAs: general information	"Cisco IOS IP SLAs Overview" module of the *Cisco IOS IP SLAs Configuration Guide*.
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" module of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

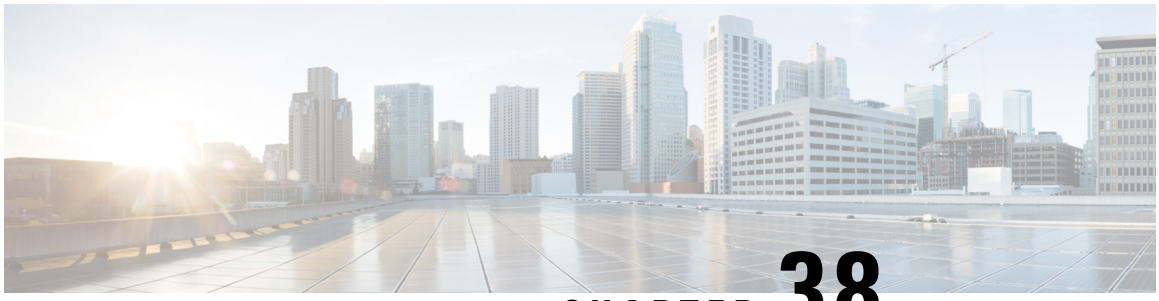
Feature Information for IP SLAs for Metro-Ethernet

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 61: Feature Information for IP SLAs for Metro-Ethernet

Feature Name	Releases	Feature Information
IP SLAs for Metro-Ethernet		The IP Service Level Agreements (SLAs) for Metro-Ethernet feature provides the capability to gather Ethernet-layer network performance metrics. Available statistical measurements for the IP SLAs Ethernet operation include round-trip time, jitter (interpacket delay variance), and packet loss.
IP SLAs Metro-Ethernet 2.0 (EVC)		Support for Ethernet Virtual Circuits (EVCs) was added.
IP SLAs Metro-Ethernet 3.0 (CFM d8.1)		Support for the Standards Based EOAM Performance Monitoring CFM base feature was added. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 900 Series.



CHAPTER 38

Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

This module describes how to configure an IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) operation to gather the following performance measurements for Ethernet service:

- Ethernet Delay
- Ethernet Delay Variation
- Ethernet Frame Loss Ratio

Prerequisites for ITU-T Y.1731 Operations

IEEE-compliant Connectivity Fault Management (CFM) must be configured and enabled for Y.1731 performance monitoring to function.

Note Y1731 is supported on Port Channel interfaces.

Restrictions for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731)

- SNMP is not supported for reporting threshold events or collecting performance statistics for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) operations.

  SNMP is partially supported; the results for DM/LM can be polled for some attributes. However MIB support for all parameters is not supported.

- Continuity Check Message (CCM)-based dual-ended Ethernet frame loss operations are not supported.

- In a single-ended Ethernet operation, performance measurement statistics can be retrieved only at the device on which the sender Ethernet Connectivity Fault Management (CFM) Maintenance End Point (MEP) is configured.
- To avoid losing the CoS value configured on the frames, do not configure **rewrite** on the EFPs throughout the Layer2 circuit. The CoS value is preserved, if the Y.1731 frames are marked with specific CoS value.
- CFM over cross-connect on the routers works only if the **control-word** is configured. To start DM timestamping, switch ON the control-word if the remote end is not switched ON.
- To avoid errors in RX and TX timestamping, ensure to have Y1731 sender as primary PTP, and the Y1731 responder as subordinate PTP.
- Reconfigure IP SLA Y1731 while doing online insertion removal (OIR) of IM or router reload because local MEP is deleted during the course.
- A delay may be observed after issuing the **ip sla schedule** command after a reload of the router is performed, to populate with the Y.1731 PM measurements.
- The dot1q tag contains class of service (CoS) bits, which are used by IPSLA Y.1731 PM session to test delay or loss of packets with a specific CoS. This CoS cannot be a non-zero value when using EPM over untagged EFPs.

How to Configure IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

Configuring a Dual-Ended Ethernet Delay or Delay Variation Operation

Perform the tasks for configuring a dual-ended operation in the order presented.

Note To remove the MEP configurations in an already-configured dual-ended operation, always remove the MEPs in the reverse order in which they were configured. That is, remove the scheduler first, then the threshold monitoring configuration, and then the sender MEP configuration on the source device before removing the scheduler, proactive threshold monitoring, and receiver MEP configuration on the destination device.

Configuring a Receiver MEP on the Destination Device

Before you begin

Time synchronization is required between the source and destination devices in order to provide accurate one-way delay (latency) or delay-variation measurements. Configure either Precision Time Protocol (PTP) or Network Time Protocol (NTP) on both the source and destination devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip sla** *operation-number*
4. **ethernet y1731 delay receive 1DM domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} **cos** *cos* {**mpid** *source-mp-id* | **mac-address** *source-address*}
5. **aggregate interval** *seconds*
6. **distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[**,...,***boundary*]
7. **frame offset** *offset-value*
8. **history interval** *intervals-stored*
9. **max-delay** *milliseconds*
10. **owner** *owner-id*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Router(config-term)# ip sla 501`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ethernet y1731 delay receive 1DM domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} **cos** *cos* {**mpid** *source-mp-id* | **mac-address** *source-address*} **Example:** `Router(config-ip-sla)# ethernet y1731 delay receive 1DM domain xxx evc yyy cos 3 mpid 101`	Begins configuring the receiver on the responder and enters IP SLA Y.1731 delay configuration mode. • The *source-mp-id* or *source-address* configured by this command corresponds to that of the MEP being configured. **Note** The session with mac-address will not be inactivated when there is CFM error.
Step 5	**aggregate interval** *seconds* **Example:** `Router(config-sla-y1731-delay)# aggregate interval 900`	(Optional) Configures the length of time during which the performance measurements are conducted and the results stored.

	Command or Action	Purpose
Step 6	**distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[**,...,***boundary*] **Example:** `Router(config-sla-y1731-delay)# distribution delay-variation one-way 5 5000,10000,15000,20000,-1`	(Optional) Specifies measurement type and configures bins for statistics distributions kept.
Step 7	**frame offset** *offset-value* **Example:** `Router(config-sla-y1731-delay)# frame offset 1`	(Optional) Sets the value for calculating delay variation rates.
Step 8	**history interval** *intervals-stored* **Example:** `Router(config-sla-y1731-delay)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.
Step 9	**max-delay** *milliseconds* **Example:** `Router(config-sla-y1731-delay)# max-delay 5000`	(Optional) Sets the amount of time an MEP waits for a frame.
Step 10	**owner** *owner-id* **Example:** `Router(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.
Step 11	**end** **Example:** `Router(config-sla-y1731-delay)# end`	Exits to privileged EXEC mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, see the "Configuring Proactive Threshold Monitoring" module of the *IP SLAs Configuration Guide*.

When you are finished configuring proactive threshold monitoring for this MEP, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Configuring the Sender MEP on the Source Router

Before you begin

- Time synchronization is required between the source and destination devices in order to provide accurate one-way delay (latency) or delay-variation measurements. Configure either Precision Time Protocol (PTP) or Network Time Protocol (NTP) on both the source and destination devices.
- The receiver MEP must be configured, including proacive threshold monitoring, and scheduled before you configure the sender MEP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla operation-number**
4. **ethernet y1731 delay 1DM domain domain-name** {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
5. **aggregate interval** *seconds*
6. **frame interval** *milliseconds*
7. **frame size** *bytes*
8. **history interval** *intervals-stored*
9. **owner** *owner-id*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla operation-number** **Example:** `Router(config)# ip sla 500`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ethernet y1731 delay 1DM domain domain-name** {**evc** *evc-id*	**vlan** *vlan-id*} {**mpid** *target-mp-id*	**mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id*	**mac-address** *source-address*}}	Begins configuring a dual-ended Ethernet delay operation and enters IP SLA Y.1731 delay configuration mode. **Note** The session with mac-address will not be inactivated when there is CFM error.

	Command or Action	Purpose
	Example: `Router(config-ip-sla)# ethernet y1731 delay 1DM domain xxx evc yyy mpid 101 cos 3 source mpid 100`	
Step 5	**aggregate interval** *seconds* **Example:** `Router(config-sla-y1731-delay)# aggregate interval 900`	(Optional) Configures the length of time during which the performance measurements are conducted and the results stored.
Step 6	**frame interval** *milliseconds* **Example:** `Router(config-sla-y1731-delay)# frame interval 100`	(Optional) Sets the gap between successive frames.
Step 7	**frame size** *bytes* **Example:** `Router(config-sla-y1731-delay)# frame size 64`	(Optional) Sets the padding size for frames.
Step 8	**history interval** *intervals-stored* **Example:** `Router(config-sla-y1731-delay)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.
Step 9	**owner** *owner-id* **Example:** `Router(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.
Step 10	**end** **Example:** `Router(config-sla-y1731-delay)# end`	Exits to privileged EXEC mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, see the "Configuring Proactive Threshold Monitoring" module of the *IP SLAs Configuration Guide*.

When you are finished configuring proactive threshold monitoring for this MEP, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Configuring a Sender MEP for a Single-Ended Ethernet Delay or Delay Variation Operation

Perform this task to configure a sender MEP on the source device.

Before you begin

- Time synchronization is required between the source and destination devices in order to provide accurate one-way delay (latency) or delay-variation measurements. Configure either Precision Time Protocol (PTP) or Network Time Protocol (NTP) on both the source and destination devices.

Note To display information about remote (target) MEPs on destination devices, use the **show ethernet cfm maintenance-points remote** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ethernet y1731 delay** {**DMM** | **DMMv1**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
5. **clock sync**
6. **aggregate interval** *seconds*
7. **distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[**,...,**boundary*]
8. **frame interval** *milliseconds*
9. **frame offset** *offset-value*
10. **frame size** *bytes*
11. **history interval** *intervals-stored*
12. **max-delay** *milliseconds*
13. **owner** *owner-id*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config-term)# ip sla 10`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ethernet y1731 delay** {**DMM** | **DMMv1**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}} **Example:** `Device(config-ip-sla)# ethernet y1731 delay dmm domain xxx evc yyy mpid 101 cos 4 source mpid 100`	Begins configuring a single-ended Ethernet delay operation and enters IP SLA Y.1731 delay configuration mode. • To configure concurrent operations, use the **DMMv1** keyword with this command. Repeat the preceding two steps to each concurrent operation, to be added to a single IP SLA operation number. Concurrent operations are supported for a given EVC, CoS, and remote MEP combination, or for multiple MEPs for a given multipoint EVC. **Note** The session with mac-address will not be inactivated when there is CFM error.
Step 5	**clock sync** **Example:** `Device(config-sla-y1731-delay)# clock sync`	(Optional) Indicates that the end points are synchronized and thus allows the operation to calculate one-way delay measurements.
Step 6	**aggregate interval** *seconds* **Example:** `Device(config-sla-y1731-delay)# aggregate interval 900`	(Optional) Configures the length of time during which the performance measurements are conducted and the results stored.
Step 7	**distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[**,...,***boundary*] **Example:** `Device(config-sla-y1731-delay)# distribution delay-variation one-way 5 5000, 10000,15000,20000,-1`	(Optional) Specifies measurement type and configures bins for statistics distributions kept.
Step 8	**frame interval** *milliseconds* **Example:**	(Optional) Sets the gap between successive frames.

	Command or Action	Purpose
	`Device(config-sla-y1731-delay)# frame interval 100`	
Step 9	**frame offset** *offset-value* **Example:** `Device(config-sla-y1731-delay)# frame offset 1`	(Optional) Sets value for calculating delay variation values.
Step 10	**frame size** *bytes* **Example:** `Device(config-sla-y1731-delay)# frame size 32`	(Optional) Configures padding size for frames.
Step 11	**history interval** *intervals-stored* **Example:** `Device(config-sla-y1731-delay)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.
Step 12	**max-delay** *milliseconds* **Example:** `Device(config-sla-y1731-delay)# max-delay 5000`	(Optional) Sets the amount of time an MEP waits for a frame.
Step 13	**owner** *owner-id* **Example:** `Device(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.
Step 14	**end** **Example:** `Device(config-sla-y1731-delay)# end`	Exits to privileged EXEC mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, see the "Configuring Proactive Threshold Monitoring" module of the *IP SLAs Configuration Guide*.

When you are finished configuring proactive threshold monitoring for this operation, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Configuring a Sender MEP for a Single-Ended Ethernet Frame Loss Ratio Operation

Note To display information about remote (target) MEPs on destination devices, use the **show ethernet cfm maintenance-points remote** command.

Perform this task to configure a sender MEP on the source device.

Before you begin

- Class of Service (CoS)-level monitoring must be enabled on MEPs associated to the Ethernet frame loss operation by using the **monitor loss counter** command on the devices at both ends of the operation. See the *Cisco IOS Carrier Ethernet Command Reference* for command information. See the "Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" section for configuration information.

Note Cisco IOS Y.1731 implementation allows monitoring of frame loss for frames on an EVC regardless of the CoS value (any CoS or Aggregate CoS cases). See the "Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" section for configuration information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ethernet y1731 loss** {**LMM** | **SLM**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **CoS** *CoS* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
5. **aggregate interval** *seconds*
6. **availability algorithm** {**sliding-window** | **static-window**}
7. **frame consecutive** *value*
8. **frame interval** *milliseconds*
9. **history interval** *intervals-stored*
10. **owner** *owner-id*
11. **exit**
12. **exit**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config-term)# ip sla 11`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ethernet y1731 loss** {**LMM** | **SLM**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **CoS** *CoS* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}} **Example:** `Device(config-ip-sla)# ethernet y1731 loss LMM domain xxx vlan 12 mpid 34 CoS 4 source mpid 23`	Begins configuring a single-ended Ethernet frame loss ratio operation and enters IP SLA Y.1731 loss configuration mode. • To configure concurrent operations, use the **SLM** keyword with this command. Repeat the preceding two steps to configure each concurrent operation to be added to a single IP SLA operation number. Concurrent operations are supported for a given EVC, CoS, and remote-MEP combination, or for multiple MEPs for a given multipoint EVC. **Note** The session with mac-address will not be inactivated when there is CFM error.
Step 5	**aggregate interval** *seconds* **Example:** `Device(config-sla-y1731-loss)# aggregate interval 900`	(Optional) Configures the length of time during which performance measurements are conducted and the results stored.
Step 6	**availability algorithm** {**sliding-window** | **static-window**} **Example:** `Device(config-sla-y1731-loss)# availability algorithm static-window`	(Optional) Specifies availability algorithm used.
Step 7	**frame consecutive** *value* **Example:** `Device(config-sla-y1731-loss)# frame consecutive 10`	(Optional) Specifies number of consecutive measurements to be used to determine availability or unavailability status.

	Command or Action	Purpose
Step 8	**frame interval** *milliseconds* **Example:** `Device(config-sla-y1731-loss)# frame interval 100`	(Optional) Sets the gap between successive frames.
Step 9	**history interval** *intervals-stored* **Example:** `Device(config-sla-y1731-loss)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.
Step 10	**owner** *owner-id* **Example:** `Device(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.
Step 11	**exit** **Example:** `Device(config-sla-y1731-delay)# exit`	Exits to IP SLA configuration mode.
Step 12	**exit** **Example:** `Device(config-ip-sla)# exit`	Exits to global configuration mode.
Step 13	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.

What to do next

When you are finished configuring this MEP, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.

- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range*	**schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever**	*seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm* [:*ss*]}] **Example:** `Device(config)# ip sla schedule 10 life forever start-time now`	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose

	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

Example: Dual-Ended Ethernet Delay Operation

The following sample output shows the configuration, including default values, of a receiver MEP on the responder device for a dual-ended Ethernet delay or delay variation operation:

```
Device# show ip sla configuration 501

IP SLAs Infrastructure Engine-III
Entry number: 501
Owner: admin
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Delay Operation
Frame Type: 1DM
Domain: xxx
ReceiveOnly: TRUE
Evc: yyy
Local Mpid: 101
CoS: 3
   Max Delay: 5000
Threshold (milliseconds): 5000
```

```
.
.
.
Statistics Parameters
  Aggregation Period: 900
  Frame offset: 1
  Distribution Delay One-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
  Distribution Delay-Variation One-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
History
  Number of intervals: 2
```

The following sample output shows the configuration, including default values, of the sender MEP for a dual-ended IP SLAs Ethernet delay or delay variation operation:

```
Device# show ip sla configuration 500

IP SLAs Infrastructure Engine-III
Entry number: 500
Owner:
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Delay Operation
Frame Type: 1DM
Domain: yyy
ReceiveOnly: FALSE
Evc: xxx
Target Mpid: 101
Source Mpid: 100
CoS: 3
   Request size (Padding portion): 64
   Frame Interval: 1000
Threshold (milliseconds): 5000
.
.
.
Statistics Parameters
  Aggregation Period: 900
  Frame offset: 1
History
  Number of intervals: 22
```

Example: Frame Delay and Frame Delay Variation Measurement Configuration

The following sample output shows the performance monitoring session summary:

```
Device# show ethernet cfm pm session summary

Number of Configured Session : 2
Number of Active Session: 2
Number of Inactive Session: 0
```

The following sample output shows the active performance monitoring session:

```
Device# show ethernet cfm pm session active

Display of Active Session
```

```
-------------------------------------------------------------------------------
EPM-ID   SLA-ID    Lvl/Type/ID/Cos/Dir     Src-Mac-address Dst-Mac-address
-------------------------------------------------------------------------------
 0     10           3/BD-V/10/2/Down     d0c2.8216.c9d7  d0c2.8216.27a3
 1     11           3/BD-V/10/3/Down     d0c2.8216.c9d7  d0c2.8216.27a3
Total number of Active Session: 2
```

```
Device# show ethernet cfm pm session db 0
```

```
-------------------------------------------------------------------------------
      TX Time FWD                   RX Time FWD
      TX Time BWD                   RX Time BWD              Frame Delay
      Sec:nSec                      Sec:nSec                  Sec:nSec
-------------------------------------------------------------------------------
Session ID: 0
*******************************************************************************
       234:526163572                 245:305791416
       245:306761904                 234:527134653            0:593
*******************************************************************************
       235:528900628                 246:308528744
       246:309452848                 235:529825333            0:601
*******************************************************************************
       236:528882716                 247:308511128
       247:309450224                 236:529822413            0:601
*******************************************************************************
       237:526578788                 248:306207432
       248:307157936                 237:527529885            0:593
*******************************************************************************
       238:527052156                 249:306681064
       249:307588016                 238:527959717            0:609
*******************************************************************************
       239:526625044                 250:306254200
       250:307091888                 239:527463325            0:593
*******************************************************************************
       240:528243204                 251:307872648
       251:308856880                 240:529228021            0:585
```

Example: Sender MEP for a Single-Ended Ethernet Delay Operation

The following sample output shows the configuration, including default values, of the sender MEP for a single-ended IP SLAs Ethernet delay operation:

```
Router# show ip sla configuration 10
```

```
IP SLAs Infrastructure Engine-III
Entry number: 10
Owner:
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Delay Operation
Frame Type: DMM
Domain: xxx
Vlan: yyy
Target Mpid: 101
Source Mpid: 100
CoS: 4
   Max Delay: 5000
   Request size (Padding portion): 64
   Frame Interval: 1000
   Clock: Not In Sync
Threshold (milliseconds): 5000
```

```
.
.
.
Statistics Parameters
  Aggregation Period: 900
  Frame offset: 1
  Distribution Delay Two-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
  Distribution Delay-Variation Two-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
History
  Number of intervals: 2
```

Example: Sender MEP for a Single-Ended Ethernet Frame Loss Operation

The following output shows the configuration, including default values, of the sender MEP in a basic single-ended IP SLAs Ethernet frame loss ratio operation with a start-time of now:

```
Router# show ip sla configuration 11

IP SLAs Infrastructure Engine-III
Entry number: 11
Owner:
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Loss Operation
Frame Type: LMM
Domain: xxx
Vlan: 12
Target Mpid: 34
Source Mpid: 23
CoS: 4
   Request size (Padding portion): 0
   Frame Interval: 1000
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): ActiveThreshold (milliseconds): 5000
Statistics Parameters
  Aggregation Period: 900
  Frame consecutive: 10
  Availability algorithm: static-window
History
  Number of intervals: 2
```

Additional References for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

Related Documents

Related Topic	Document Title
Cisco IOS Carrier Ethernet commands	Cisco IOS Carrier Ethernet Command Reference
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference
Ethernet CFM	"Configuring Ethernet Connectivity Fault Management in a Service Provider Network" module of the *Cisco IOS Carrier Ethernet Configuration Guide*
Network Time Protocol (NTP)	"Configuring NTP" module of the *Cisco IOS Network Management Configuration Guide*
Proactive threshold monitoring for Cisco IOS IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
ITU-T Y.1731	*OAM functions and mechanisms for Ethernet-based networks*
No specific RFCs are supported by the features in this document.	--

MIBs

MIB	MIBs Link
• CISCO-IPSLA-ETHERNET-MIB • CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

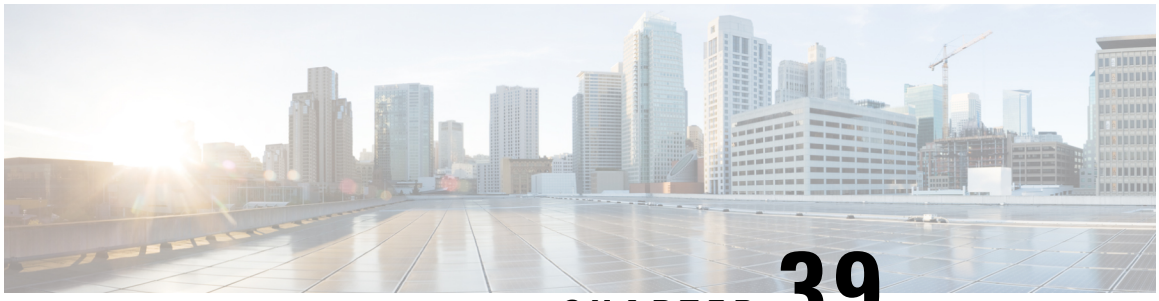
Feature Information for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 62: Feature Information for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731)

Feature Name	Releases	Feature Information
IP SLA Support for ETH-SLM (Ethernet Synthetic Loss Measurement in Y1731)		Y.1731 Performance Monitoring (PM) provides a standard Ethernet PM function that includes measurement of Ethernet frame delay, frame delay variation, frame loss, and frame throughput measurements specified by the ITU-T Y-1731 standard and interpreted by the Metro Ethernet Forum (MEF) standards group.
Y1731 MIB Support through existing IPSLA MIBs		Support was added for reporting threshold events and collecting performance statistics for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) operations using SNMP.



CHAPTER 39

IPSLA Y1731 On-Demand and Concurrent Operations

This module describes how to configure the IPSLA Y1731 SLM Feature Enhancements feature for enabling real-time Ethernet service troubleshooting for users without configuration privileges. This feature supports on-demand Synthetic Loss Measurement (SLM) operations that can be run by issuing a single command in privileged EXEC mode.

Prerequisites for ITU-T Y.1731 Operations

IEEE-compliant Connectivity Fault Management (CFM) must be configured and enabled for Y.1731 performance monitoring to function.

Note Y1731 is supported on Port Channel interfaces.

Restrictions for IP SLAs Y.1731 On-Demand Operations

- SNMP is not supported for reporting threshold events or collecting performance statistics for on-demand operations.
- On-demand operation statistics are not stored and are not supported by the statistic history and aggregation functions.

Information About IP SLAs Y.1731 On-Demand and Concurrent Operations

IPSLA Y1731 SLM Feature Enhancements

On-demand IP SLAs Synthetic Loss Measurement (SLM) operations, in the IPSLA Y1731 SLM Feature Enhancements feature, enable users without configuration access to perform real-time troubleshooting of Ethernet services. There are two operational modes for on-demand operations: direct mode that creates and runs an operation immediately and referenced mode that starts and runs a previously configured operation.

- In the direct mode, a single command can be used to create multiple pseudo operations for a range of class of service (CoS) values to be run, in the background, immediately. A single command in privileged EXEC mode can be used to specify frame size, interval, frequency, and duration for the direct on-demand operation. Direct on-demand operations start and run immediately after the command is issued.
- In the referenced mode, you can start one or more already-configured operations for different destinations, or for the same destination, with different CoS values. Issuing the privileged EXEC command creates a pseudo version of a proactive operation that starts and runs in the background, even while the proactive operation is running.
- Once an on-demand operation is completed, statistical output is displayed on the console. On-demand operation statistics are not stored and are not supported by the statistic history and aggregation functions.
- After an on-demand operation is completed, and the statistics handled, the direct and referenced on-demand operation is deleted. The proactive operations are not deleted and continue to be available to be run in referenced mode, again.

A concurrent operation consists of a group of operations, all configured with the same operation ID number, that run concurrently. Concurrent operations are supported for a given Ethernet Virtual Circuit (EVC), CoS, and remote Maintenance End Point (MEP) combination, or for multiple MEPs for a given multipoint EVC, for delay or loss measurements. A new keyword was added to the appropriate commands to specify that concurrent Ethernet frame Delay Measurement (ETH-DM) synthetic frames are sent during the operation.

The IPSLA Y.1731 SLM Feature Enhancements feature also supports burst mode for concurrent operations, one-way dual-ended, and single-ended delay and delay variation operations, as well as for single-ended loss operations. A new keyword was added to the appropriate commands to support bursts of PDU transmission during an aggregation interval. The maximum number of services monitored is 50 every 30 minutes, with an average of 25 services every 2 hours.

How to Configure IP SLAs Y.1731 On-Demand and Concurrent Operations

Configuring a Direct On-Demand Operation on a Sender MEP

Before you begin

Class of Service (CoS)-level monitoring must be enabled on MEPs associated to the Ethernet frame loss operation by using the **monitor loss counter** command on the devices at both ends of the operation. See the *Cisco IOS Carrier Ethernet Command Reference* for command information. See the "Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" section for configuration information.

Note The Cisco IOS Y.1731 implementation allows monitoring of frame loss on an EVC regardless of the CoS value (any CoS or aggregate CoS cases). See the "Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" section for configuration information.

SUMMARY STEPS

1. **enable**
2. **ip sla on-demand ethernet** {**DMMv1** | **SLM**} **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}} {**continuous** [**interval** *milliseconds*] | **burst** [**interval** *milliseconds*] [**number** *number-of-frames*] [**frequency** *seconds*]} [**size** *bytes*] **aggregation** *seconds* {**duration** *seconds* | **max** *number-of-packets*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ip sla on-demand ethernet** {**DMMv1** | **SLM**} **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}} {**continuous** [**interval** *milliseconds*] | **burst** [**interval** *milliseconds*] [**number** *number-of-frames*] [**frequency** *seconds*]} [**size** *bytes*] **aggregation** *seconds* {**duration** *seconds* | **max** *number-of-packets*} **Example:**	Creates and runs an on-demand operation in direct mode. • To create and run concurrent on-demand operations, configure this command using the **DMMv1** keyword. • Statistical output is posted on the console after the operation is finished. • Repeat this step for each on-demand operation to be run. • After an on-demand operation is finished and the statistics handled, the operation is deleted.

	Command or Action	Purpose
	`Device# ip sla on-demand ethernet SLM domain xxx vlan 12 mpid 34 cos 4 source mpid 23 continuous aggregation 10 duration 60`	

Configuring a Referenced On-Demand Operation on a Sender MEP

Note After an on-demand operation is finished and the statistics handled, the on-demand version of the operation is deleted.

Before you begin

- Single-ended and concurrent Ethernet delay, or delay variation, and frame loss operations to be referenced must be configured. See the "Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" module of the *IP SLAs Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **ip sla on-demand ethernet** [**dmmv1** | **slm**] *operation-number* {**duration** *seconds* | **max** *number-of-packets*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ip sla on-demand ethernet** [**dmmv1**	**slm**] *operation-number* {**duration** *seconds*	**max** *number-of-packets* **Example:** `Device# ip sla on-demand ethernet slm 11 duration 38`	Creates and runs a pseudo operation of the operation being referenced, in the background. • Statistical output is posted on the console after the operation is finished. • Repeat this step for each on-demand operation to be run.

Configuring an IP SLAs Y.1731 Concurrent Operation on a Sender MEP

To configure concurrent Ethernet delay, delay variation, and frame loss operations, see the "Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" module of the

IP SLAs Configuration Guide.

Configuration Examples for IP SLAs Y.1731 On-Demand and Concurrent Operations

Example: On-Demand Operation in Direct Mode

```
Device# ip sla on-demand ethernet SLM domain xxx vlan 10 mpid 3 cos 1 source mpid 1 continuous
 aggregation 35 duration 38

Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK

Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK
```

```
Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
```

Example: On-Demand Operation in Referenced Mode

```
Device(config)# ip sla 11
Device(config-ip-sla)# ethernet y1731 loss SLM domain xxx vlan 10 mpid 3 cos 1 source mpid
 1
Device(config-sla-y1731-loss)# end
Device# ip sla on-demand ethernet slm 11 duration 38

Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK

Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
```

```
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK

Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
```

IP SLA Reconfiguration Scenarios

IP SLA Reconfiguration Scenarios

IP SLA must be reconfigured in the following scenarios:

- When an Ethernet service instance is disabled on the interface using the **service instance ethernet** command.
- When the local MEP is removed using the **no cfm mep domain** *domain-name* **mpid** *mpid* command.

- When the configuration of an interface is reset to its default values, using the **default interface** command.
- When an interface configuration is removed using the **no interface** command.
- When the Ethernet Connectivity Fault Management (CFM) distribution is disabled using the **no ethernet cfm global** and **no ethernet cfm ieee** commands.

Additional References for IP SLAs Y.1731 On-Demand and Concurrent Operations

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS Carrier Ethernet commands	Cisco IOS Carrier Ethernet Command Reference
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference
Ethernet CFM for ITU-T Y.1731	"ITU-T Y.1731 Performance Monitoring in a Service Provider Network" module of the *Carrier Ethernet Configuration Guide*
Ethernet operations	"Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" module of the *IP SLAs Configuration Guide*
Network Time Protocol (NTP)	"Configuring NTP" module of the *Network Management Configuration Guide*

Standards and RFCs

Standard/RFC	Title
ITU-T Y.1731	*OAM functions and mechanisms for Ethernet-based networks*

MIBs

MIB	MIBs Link
• CISCO-IPSLA-ETHERNET-MIB • CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

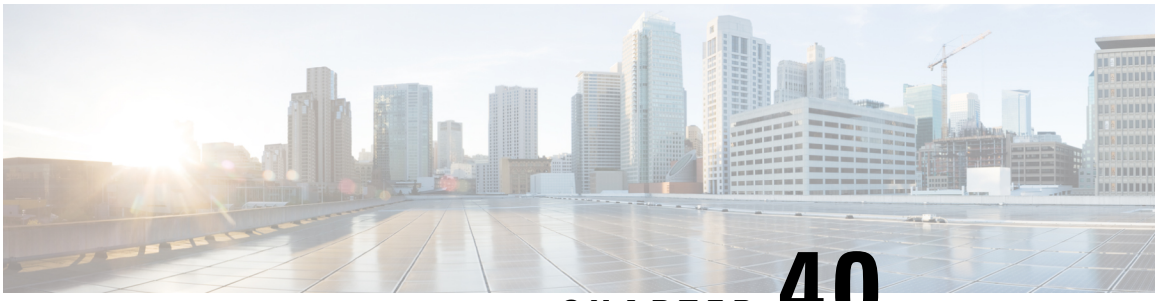
Feature Information for IP SLAs Y.1731 On-Demand and Concurrent Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 63: Feature Information for IP SLAs Y.1731 On-Demand and Concurrent Operations

Feature Name	Releases	Feature Information
IPSLA Y1731 SLM Feature Enhancements		This feature enhancement allows you to run on-demand Synthetic Loss Measurement (SLM) operations, independent from previously scheduled operations, for the purpose of troubleshooting Etherent services in your network. The following commands were introduced or modified: **ethernet y1731 delay**, **ethernet y1737 loss**, **ip sla on-demand ethernet**.



CHAPTER 40

Configuring IP SLAs UDP Echo Operations

This module describes how to configure an IP Service Level Agreements (SLAs) User Datagram Protocol (UDP) Echo operation to monitor end-to-end response time between a Cisco device and devices using IPv4 or IPv6. UDP echo accuracy is enhanced by using the Cisco IP SLAs Responder at the destination Cisco device. This module also demonstrates how the results of the UDP echo operation can be displayed and analyzed to determine how a UDP application is performing.

Restrictions for IP SLAs UDP Echo Operations

We recommend using a Cisco networking device as the destination device, although any networking device that supports RFC 862, *Echo Protocol* , can be used.

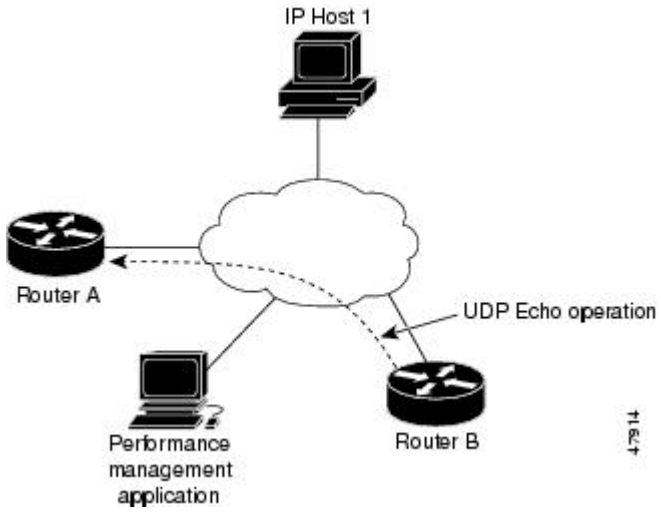
Information About IP SLAs UDP Echo Operations

UDP Echo Operation

The UDP echo operation measures end-to-end response time between a Cisco device and devices using IP. UDP is a transport layer (Layer 4) Internet protocol that is used for many IP services. UDP echo is used to measure response times and test end-to-end connectivity.

In the figure below Device A has been configured as an IP SLAs Responder and Device B is configured as the source IP SLAs device.

Figure 43: UDP Echo Operation



Response time (round-trip time) is computed by measuring the time taken between sending a UDP echo request message from Device B to the destination device--Device A--and receiving a UDP echo reply from Device A. UDP echo accuracy is enhanced by using the IP SLAs Responder at Device A, the destination Cisco device. If the destination device is a Cisco device, then IP SLAs sends a UDP datagram to any port number that you specified. Using the IP SLAs Responder is optional for a UDP echo operation when using Cisco devices. The IP SLAs Responder cannot be configured on non-Cisco devices.

The results of a UDP echo operation can be useful in troubleshooting issues with business-critical applications by determining the round-trip delay times and testing connectivity to both Cisco and non-Cisco devices.

How to Configure IP SLAs UDP Echo Operations

Configuring the IP SLAs Responder on a Destination Device

Note A responder should not configure a permanent port for a sender. If the responder configures a permanent port for a sender, even if the packets are successfully sent (no timeout or packet-loss issues), the jitter value is zero.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla responder**
   - **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla responder**
	• **ip sla responder udp-echo ipaddress** *ip-address* **port** *port***vrf** *vrf*
	Example:
	`Device(config)# ip sla responder`
	`Device(config)# ip sla responder udp-echo ipaddress 192.0.2.132 port 5000 vrf vrf1`
		(Optional) Temporarily enables IP SLAs responder functionality on a Cisco device in response to control messages from the source.
		(Optional; required only if protocol control is disabled on the source.) Enables IP SLAs responder functionality on the specified IP address, port and VRF.
		• Protocol control is enabled by default.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a UDP Echo Operation on the Source Device

Perform only one of the following tasks:

Configuring a Basic UDP Echo Operation on the Source Device

Before you begin

If you are using the IP SLAs Responder, ensure that you have completed the "Configuring the IP SLAs Responder on the Destination Device" section before you start this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-echo** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}]
5. **data-pattern** *hex value*

6. **frequency** *seconds*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-echo** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `Device(config-ip-sla)# udp-echo 172.29.139.134 5000`	Defines a UDP echo operation and enters IP SLA UDP configuration mode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices.
Step 5	**data-pattern** *hex value* **Example:** `Device(config-ip-sla-udp)# data-pattern FFFFFFFF`	(Optional) Sets a hexadecimal value for data pattern. The range is 0 to FFFFFFFF.
Step 6	**frequency** *seconds* **Example:** `Device(config-ip-sla-udp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 7	**end** **Example:** `Device(config-ip-sla-udp)# end`	Returns to prileged EXEC mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Configuring a UDP Echo Operation with Optional Parameters on the Source Device

Before you begin

If you are using an IP SLAs Responder in this operation, the responder must be configured on the destination device. See the "Configuring the IP SLAs Responder on the Destination Device."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **udp-echo** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}]
5. **history buckets-kept** *size*
6. **data-pattern** *hex-pattern*
7. **history distributions-of-statistics-kept** *size*
8. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
9. **history filter** {**none** | **all** | **overThreshold** | **failures**}
10. **frequency** *seconds*
11. **history hours-of-statistics-kept** *hours*
12. **history lives-kept** *lives*
13. **owner** *owner-id*
14. **request-data-size** *bytes*
15. **history statistics-distribution-interval** *milliseconds*
16. **tag** *text*
17. **threshold** *milliseconds*
18. **timeout** *milliseconds*
19. Do one of the following:
    - **tos** *number*
    - **traffic-class** *number*
20. **flow-label** *number*
21. **verify-data**
22. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**udp-echo** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `Device(config-ip-sla)# udp-echo 172.29.139.134 5000`	Defines a UDP echo operation and enters IP SLA UDP configuration mode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices.
Step 5	**history buckets-kept** *size* **Example:** `Device(config-ip-sla-udp)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**data-pattern** *hex-pattern* **Example:** `Device(config-ip-sla-udp)# data-pattern`	(Optional) Specifies the data pattern in an IP SLAs operation to test for data corruption.
Step 7	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-udp)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 8	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-udp)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 9	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Device(config-ip-sla-udp)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 10	**frequency** *seconds* **Example:** `Device(config-ip-sla-udp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.

	Command or Action	Purpose
Step 11	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-udp)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 12	**history lives-kept** *lives* **Example:** `Device(config-ip-sla-udp)# history lives-kept 2`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 13	**owner** *owner-id* **Example:** `Device(config-ip-sla-udp)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 14	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-udp)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 15	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-udp)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 16	**tag** *text* **Example:** `Device(config-ip-sla-udp)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 17	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-udp)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 18	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-udp)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 19	Do one of the following: • **tos** *number* • **traffic-class** *number*	(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation. or

	Command or Action	Purpose

	Example:
	`Device(config-ip-sla-jitter)# tos 160`
	Example:
	`Device(config-ip-sla-jitter)# traffic-class 160`
		(Optional) In an IPv6 network only, defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

Step 20	**flow-label** *number* **Example:** `Device(config-ip-sla-udp)# flow-label 112233`	(Optional) In an IPv6 network only, defines the flow label field in the IPv6 header for a supported IP SLAs operation.
Step 21	**verify-data** **Example:** `Device(config-ip-sla-udp)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 22	**exit** **Example:** `Device(config-ip-sla-udp)# exit`	Exits UDP configuration submode and returns to global configuration mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]

4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`
		• Configures the scheduling parameters for an individual IP SLAs operation.
		• Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:**	(Optional) Displays IP SLAs group schedule details.

	Command or Action	Purpose
	`Device# show ip sla group schedule`	
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Configuration Examples for IP SLAs UDP Echo Operations

Example Configuring a UDP Echo Operation

The following example configures an IP SLAs operation type of UDP echo that will start immediately and run indefinitely.

```
ip sla 5
 udp-echo 172.29.139.134 5000
 frequency 30
 request-data-size 160
 tos 128
 timeout 1000
 tag FLL-RO
ip sla schedule 5 life forever start-time now
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 862	Echo Protocol

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

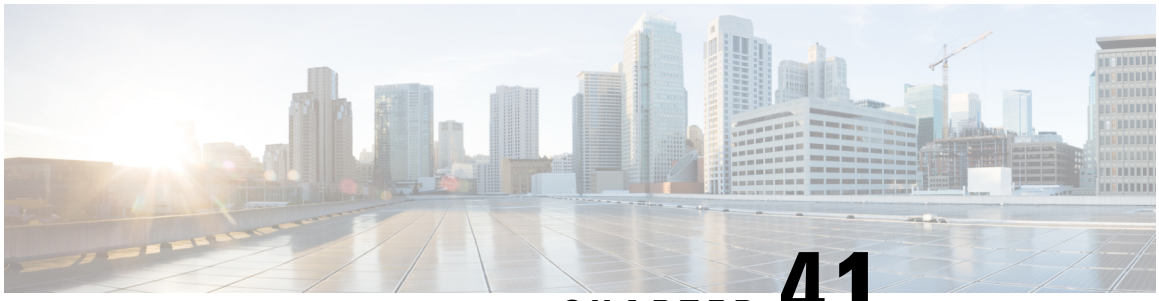
Feature Information for the IP SLAs UDP Echo Operation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 64: Feature Information for the IP SLAs UDP Echo Operation

Feature Name	Releases	Feature Information
IP SLAs - UDP Echo Operation		The Cisco IOS IP SLAs User Datagram Protocol (UDP) jitter operation allows you to measure round-trip delay, one-way delay, one-way jitter, one-way packet loss, and connectivity in networks that carry UDP traffic.
IPv6 - IP SLAs (UDP Jitter, UDP Echo, ICMP Echo, TCP Connect)		Support was added for operability in IPv6 networks.



CHAPTER 41

Configure IP SLAs HTTPS Operations

This module describes how to configure an IP Service Level Agreements (SLAs) HTTPS operation to monitor the response time between a Cisco device and an HTTPS server to retrieve a web page. The IP SLAs HTTPS operation supports both the normal GET requests and customer RAW requests. This module also demonstrates how the results of the HTTPS operation can be displayed and analyzed to determine how an HTTPS server is performing.

Restrictions for IP SLAs HTTP Operations

- IP SLAs HTTP operations support only HTTP/1.0.
- HTTP/1.1 is not supported for any IP SLAs HTTP operation, including HTTP RAW requests.
- If IP SLA probe fails while configuring IP SLA HTTP operation using a public server, install your target server's certificate authority (CA) certificate as a trusted CA in the router.

Information About IP SLAs HTTPS Operations

HTTPS Operation

The HTTPS operation measures the round-trip time (RTT) between a Cisco device and an HTTPS server to retrieve a web page. The HTTPS server response time measurements consist of three types:

The HTTPS operation measures the round-trip time (RTT) between a Cisco device and an HTTPS server to retrieve a web page.

The IPSLA HTTPS operation uses the Cisco IOS XE HTTPS secure client to send the HTTPS request, process the response from the HTTPS server and pass the response back to IPSLA.

The HTTPS server response time measurements consist of two types:

DNS lookup--RTT taken to perform domain name lookup.

HTTPS transaction time-- RTT taken by the Cisco IOS XE HTTPS secure client to send HTTPS request to the HTTPS server, get the response from the server.

The DNS operation is performed first and the DNS RTT is measured. Once the domain name is found, request with GET or HEAD method is sent to the Cisco IOS XE HTTPS secure client to send HTTPS request to the HTTPS server and RTT taken to retrieve the home HTML page from the HTTPS server is measured. This RTT includes the time taken for SSL handshake, TCP connection to the server and HTTPS transactions.

The total RTT is a sum of the DNS RTT and the HTTPS transaction RTT.

Currently, the error codes are determined, and the IP SLA HTTPS operation goes down only if the return code is not 200. Use http-status-code-ignore command to ignore the HTTPS status code and consider the operation's status as OK.

How to Configure IP SLAs HTTP Operations

Configure an HTTPS GET Operation on the Source Device

Note This operation does not require an IP SLAs Responder on the destination device.

Perform only one of the following tasks:

Configure a Basic HTTPS GET Operation on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **http secure** {**get** | **head**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** {**interface-name**}]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**http secure** {**get** | **head**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** {**interface-name**}] **Example:** `Device(config-ip-sla)# http secure get https://www.cisco.com/index.html`	Defines anHTTPs operation and enters IP SLA configuration mode.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-http)# frequency 90`	(Optional) Sets the rate at which a specified IP SLAs HTTPS operation repeats. The default and minimum frequency value for an IP SLAs HTTPS operation is 60 seconds.
Step 6	**end** **Example:** `Device(config-ip-sla-http)# end`	Exits to privileged EXEC mode.

Configure an HTTPS GET Operation with Optional Parameters on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **http secure** {**get** | **raw**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** *ip-address* {*interface-name*}]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**http secure** {**get** | **raw**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** *ip-address* {*interface-name*}] **Example:** `Device(config-ip-sla)# http secure get https://www.cisco.com/index.html`	Defines an HTTPS operation and enters IP SLA configuration mode.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-http)# frequency 90`	(Optional) Sets the rate at which a specified IP SLAs HTTP operation repeats. The default and minimum frequency value for an IP SLAs HTTP operation is 60 seconds.
Step 6	**end** **Example:** `Device(config-ip-sla-http)# end`	Exits to privileged EXEC mode.

Configuring an HTTP RAW Operation on the Source Device

Note This operation does not require an IP SLAs Responder on the destination device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **http** {**get** | **raw**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**cache** {**enable** | **disable**}] [**proxy** *proxy-url*]
5. **http-raw-request**
6. Enter the required HTTP 1.0 command syntax.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**http** {**get** | **raw**} *url* [**name-server** *ip-address*] [**version** *version-number*] [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**cache** {**enable** | **disable**}] [**proxy** *proxy-url*] **Example:** `Device(config-ip-sla)# http raw http://198.133.219.25`	Defines an HTTP operation.
Step 5	**http-raw-request** **Example:** `Device(config-ip-sla)# http-raw-request`	Enters HTTP RAW configuration mode.
Step 6	Enter the required HTTP 1.0 command syntax. **Example:** `Device(config-ip-sla-http)# GET /en/US/hmpgs/index.html HTTP/1.0\r\n\r\n`	Specifies all the required HTTP 1.0 commands.
Step 7	**end** **Example:** `Device(config-ip-sla-http)# end`	Exits to privileged EXEC mode.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range*	**schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever**	*seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm* [:*ss*]}] **Example:** `Device(config)# ip sla schedule 10 life forever start-time now` `Device(config)# ip sla group schedule 10 schedule-period frequency`	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose

	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Configuration Examples for IP SLAs HTTPS Operations

Example Configuring an HTTPS GET Operation

```
ip sla 1
http secure get https://www.cisco.com name-server 8.8.8.8 version 1.1
ip sla schedule 1 life forever start-time now
```

Example Configuring an HTTPS HEAD Operation

```
ip sla 1
http secure head  https://www.cisco.com name-server 8.8.8.8 version 1.1
ip sla schedule 1 life forever start-time now
```

Example Configuring an HTTP RAW Operation Through a Proxy Server

The following example shows how to configure an HTTP RAW operation through a proxy server. The proxy server is www.proxy.cisco.com and the HTTP server is www.yahoo.com.

```
ip sla 8
 http raw url http://www.proxy.cisco.com
 http-raw-request
 GET http://www.yahoo.com HTTP/1.0\r\n
 \r\n
 end
ip sla schedule 8 life forever start-time now
```

Example Configuring an HTTP RAW Operation with Authentication

The following example shows how to configure an HTTP RAW operation with authentication.

```
ip sla 8
 http raw url http://site-test.cisco.com
 http-raw-request
 GET /lab/index.html HTTP/1.0\r\n
 Authorization: Basic btNpdGT4biNvoZe=\r\n
 \r\n
 end
ip sla schedule 8 life forever start-time now
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

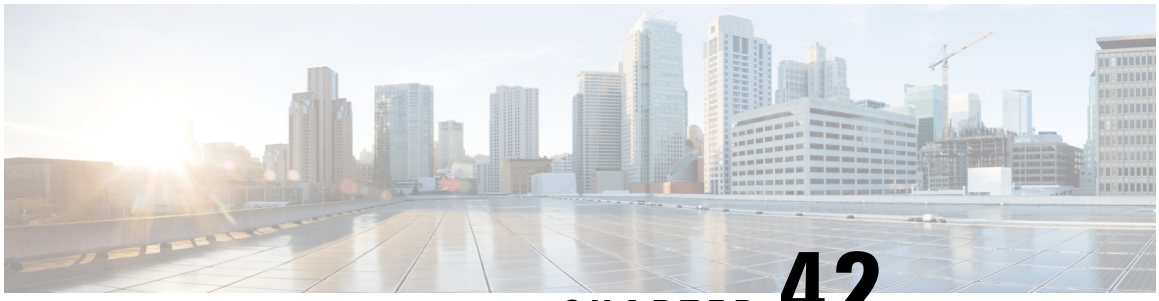
Feature Information for IP SLAs HTTP Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 65: Feature Information for IP SLAs HTTP Operations

Feature Name	Releases	Feature Information
IP SLAs HTTP Operation		The Cisco IOS IP SLAs Hypertext Transfer Protocol (HTTP) operation allows you to measure the network response time between a Cisco device and an HTTP server to retrieve a web page.
IPSLA 4.0 - IP v6 phase2		Support was added for operability in IPv6 networks. The following commands are introduced or modified: **http (IP SLA)**, **show ip sla configuration**, **show ip sla summary**.
IP SLAs VRF Aware 2.0		Support was added for IP SLAs VRF-aware capabilities for TCP connect, FTP, HTTP and DNS client operation types.



CHAPTER 42

Configuring IP SLAs TCP Connect Operations

This module describes how to configure an IP Service Level Agreements (SLAs) TCP Connect operation to measure the response time taken to perform a TCP Connect operation between a Cisco router and devices using IPv4 or IPv6. TCP Connect accuracy is enhanced by using the IP SLAs Responder at the destination Cisco router. This module also demonstrates how the results of the TCP Connect operation can be displayed and analyzed to determine how the connection times to servers and hosts within your network can affect IP service levels. The TCP Connect operation is useful for measuring response times for a server used for a particular application or connectivity testing for server availability.

Information About the IP SLAs TCP Connect Operation

TCP Connect Operation

The IP SLAs TCP Connect operation measures the response time taken to perform a TCP Connect operation between a Cisco device and devices using IP. TCP is a transport layer (Layer 4) Internet protocol that provides reliable full-duplex data transmission. The destination device can be any device using IP or an IP SLAs Responder.

In the figure below Device B is configured as the source IP SLAs device and a TCP Connect operation is configured with the destination device as IP Host 1.

Connection response time is computed by measuring the time taken between sending a TCP request message from Device B to IP Host 1 and receiving a reply from IP Host 1.

TCP Connect accuracy is enhanced by using the IP SLAs Responder at the destination Cisco device. If the destination device is a Cisco device, then IP SLAs makes a TCP connection to any port number that you specified. If the destination is not a Cisco IP host, then you must specify a known destination port number such as 21 for FTP, 23 for Telnet, or 80 for an HTTP server.

Using the IP SLAs Responder is optional for a TCP Connect operation when using Cisco devices. The IP SLAs Responder cannot be configured on non-Cisco devices.

TCP Connect is used to test virtual circuit availability or application availability. Server and application connection performance can be tested by simulating Telnet, SQL, and other types of connection to help you verify your IP service levels.

How to Configure the IP SLAs TCP Connect Operation

Configuring the IP SLAs Responder on the Destination Device

Before you begin

If you are using the IP SLAs Responder, ensure that the networking device to be used as the responder is a Cisco device and that you have connectivity to that device through the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip sla responder**
   - **ip sla responder tcp-connect ipaddress** *ip-address* **port** *port* **vrf** *vrf*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **ip sla responder** • **ip sla responder tcp-connect ipaddress** *ip-address* **port** *port* **vrf** *vrf* **Example:** `Device(config)# ip sla responder` **Example:**	(Optional) Temporarily enables IP SLAs responder functionality on the Cisco device in response to control messages from source. or (Optional) Required only if protocol control is explicitly disabled on the source device. Permanently enables IP SLAs responder functionality on the specified IP address and port and the VRF. • Control is enabled by default.

	Command or Action	Purpose
	`Device(config)# ip sla responder tcp-connect ipaddress 172.29.139.132 port 5000 vrf vrf1`	
Step 4	**exit** **Example:** `Device(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring and Scheduling a TCP Connect Operation on the Source Device

Perform only one of the following tasks:

Prerequisites

If you are using the IP SLAs Responder, complete the "Configuring the IP SLAs Responder on the Destination Device" section before you start this task.

Configuring a Basic TCP Connect Operation on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **tcp-connect** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.

	Command or Action	Purpose
Step 4	**tcp-connect** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `Device(config-ip-sla)# tcp-connect 172.29.139.132 5000`	Defines a TCP Connect operation and enters IP SLA TCP configuration mode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-tcp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-tcp)# end`	Returns to global configuration mode.

Configuring a TCP Connect Operation with Optional Parameters on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **tcp-connect** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}]
5. **history buckets-kept** *size*
6. **history distributions-of-statistics-kept** *size*
7. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
8. **history filter** {**none** | **all** | **overThreshold** | **failures**}
9. **frequency** *seconds*
10. **history hours-of-statistics-kept** *hours*
11. **history lives-kept** *lives*
12. **owner** *owner-id*
13. **history statistics-distribution-interval** *milliseconds*
14. **tag** *text*
15. **threshold** *milliseconds*
16. **timeout** *milliseconds*
17. Do one of the following:
    - **tos** *number*
    - **traffic-class** *number*
18. **flow-label** *number*

19. **exit**
20. **show ip sla configuration** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**tcp-connect** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] [**control** {**enable** | **disable**}] **Example:** `Device(config-ip-sla)# tcp-connect 172.29.139.132 5000`	Defines a TCP Connect operation and enters IP SLA TCP configuration mode. • Use the **control disable** keyword combination only if you disable the IP SLAs control protocol on both the source and target devices.
Step 5	**history buckets-kept** *size* **Example:** `Device(config-ip-sla-tcp)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-tcp)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 7	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-tcp)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 8	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Device(config-ip-sla-tcp)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.

	Command or Action	Purpose
Step 9	**frequency** *seconds* **Example:** `Device(config-ip-sla-tcp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 10	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-tcp)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 11	**history lives-kept** *lives* **Example:** `Device(config-ip-sla-tcp)# history lives-kept 2`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 12	**owner** *owner-id* **Example:** `Device(config-ip-sla-tcp)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 13	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-tcp)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 14	**tag** *text* **Example:** `Device(config-ip-sla-tcp)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 15	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-tcp)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 16	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-tcp)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 17	Do one of the following:
	• **tos** *number*
	• **traffic-class** *number*
	Example:
	`Device(config-ip-sla-jitter)# tos 160`
	Example:
	`Device(config-ip-sla-jitter)# traffic-class 160`
		(Optional) For IPv4: Defines the ToS byte in the IPv4 header of an IP SLAs operation.
		or
		(Optional) For IPv6: Defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

	Command or Action	Purpose
Step 18	**flow-label** *number* **Example:** `Device(config-ip-sla-tcp)# flow-label 112233`	(Optional) For IPv6: Defines the flow label field in the IPv6 header for a supported IP SLAs operation.
Step 19	**exit** **Example:** `Device(config-ip-sla-tcp)# exit`	Exits TCP configuration submode and returns to global configuration mode.
Step 20	**show ip sla configuration** [*operation-number*] **Example:** `Device# show ip sla configuration 10`	(Optional) Displays configuration values including all defaults for all IP SLAs operations or a specified operation.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`
		• Configures the scheduling parameters for an individual IP SLAs operation.
		• Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Configuration Examples for IP SLAs TCP Connect Operations

Example Configuring a TCP Connect Operation

The following example shows how to configure a TCP Connect operation from Device B to the Telnet port (TCP port 23) of IP Host 1 (IP address 10.0.0.1), as shown in the "TCP Connect Operation" figure in the "Information About the IP SLAs TCP Connect Operation" section. The operation is scheduled to start immediately. In this example, the control protocol is disabled on the source (Device B). IP SLAs uses the control protocol to notify the IP SLAs responder to enable the target port temporarily. This action allows the responder to reply to the TCP Connect operation. In this example, because the target is not a Cisco device and a well-known TCP port is used, there is no need to send the control message.

Device A (target device) Configuration

```
configure terminal
 ip sla responder tcp-connect ipaddress 10.0.0.1 port 23
```

Device B (source device) Configuration

```
ip sla 9
 tcp-connect 10.0.0.1 23 control disable
 frequency 30
 tos 128
 timeout 1000
 tag FLL-RO
ip sla schedule 9 start-time now
```

The following example shows how to configure a TCP Connect operation with a specific port, port 23, and without an IP SLAs responder. The operation is scheduled to start immediately and run indefinitely.

```
ip sla 9
 tcp-connect 173.29.139.132 21 control disable
 frequency 30
ip sla schedule 9 life forever start-time now
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference, All Releases
Cisco IOS IP SLAs: general information	"Cisco IOS IP SLAs Overview" module of the *Cisco IOS IP SLAs Configuration Guide*.
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" module of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

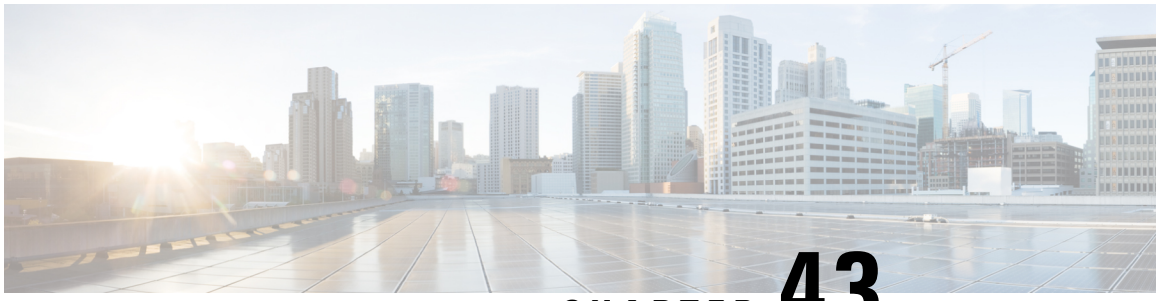
Feature Information for the IP SLAs TCP Connect Operation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 66: Feature Information for the IP SLAs TCP Connect Operation

Feature Name	Releases	Feature Information
IP SLAs TCP Connect Operation		The Cisco IOS IP SLAs Transmission Control Protocol (TCP) connect operation allows you to measure the network response time taken to perform a TCP Connect operation between a Cisco device and other devices using IP.
IPv6 - IP SLAs (UDP Jitter, UDP Echo, ICMP Echo, TCP Connect)		Support was added for operability in IPv6 networks.
IP SLAs VRF Aware 2.0		Support was added for IP SLAs VRF-aware capabilities for TCP connect, FTP, HTTP and DNS client operation types.



CHAPTER 43

Configuring Cisco IP SLAs ICMP Jitter Operations

This module describes how to configure a Cisco IOS IP Service Level Agreements (SLAs) Internet Control Message Protocol (ICMP) Jitter operation for generating a stream of ICMP packets between a Cisco IOS device (source) and any other IP device (destination) to gather network performance-related statistics. The destination device can be any network device that supports ICMP such as a server or workstation. Available statistical measurements for IP SLAs ICMP jitter operations include latency, round-trip time, jitter (interpacket delay variance), and packet loss. The IP SLAs ICMP jitter operation does not require an IP SLAs Responder on the destination device.

Restrictions for IP SLAs ICMP Jitter Operations

- Cisco IOS-XR devices do not support ICMP Timestamp and hence all ICMP jitter operations to these devices fail.
- When compared to the IP SLAs User Datagram Protocol (UDP) jitter operation, the IP SLAs ICMP jitter operation may provide less accurate measurements because the accuracy of the measurements provided by a non-Cisco destination device cannot be determined.
- Because ICMP packets do not support voice technology, the IP SLAs ICMP jitter operation does not support Mean Opinion Score (MOS), Calculated Planning Impairment Factor (ICPIF), or estimated transmission rating factor (R) reaction configuration capabilities.

Information About IP SLAs ICMP Jitter Operations

Benefits of the IP SLAs ICMP Jitter Operation

The IP SLAs ICMP Jitter Operation feature provides the following key benefits:

- End-to-end performance measurements between a Cisco device (source) and any other IP device (destination) using ICMP.
- Proactive threshold violation monitoring through Simple Network Management Protocol (SNMP) trap notifications and syslog messages.

Statistics Measured by the IP SLAs ICMP Jitter Operation

The IP SLAs ICMP jitter operation supports the following statistical measurements:

- Jitter (source-to-destination and destination-to-source)
- Latency (source-to-destination and destination-to-source)
- Round-trip time latency
- Packet loss
- Successive packet loss
- Out-of-sequence packets (source-to-destination, destination-to-source, and round-trip)
- Late packets

IP SLAs ICMP jitter uses a two ICMP time stamp messages, an ICMP Timestamp Request (Type 13) and an ICMP Timestamp Reply (Type 14), to provide jitter, packet loss, and latency. IP SLAs ICMP jitter operations differ from IP SLAs ICMP echo operations in that ICMP echo uses ICMP Echo request and reply (ping). Devices that are fully compliant with RFC 792, *Internet Control Message Protocol* , must be able to respond to the time stamp messages without requiring an IP SLA responder at the destination.

Note Cisco IOS devices support RFC 792's timestamp requests and replies, but Cisco IOS-XR devices do not support this.

The ICMP API sends a configurable number of request message packets out of the interface. The data (time stamp) that is received in the request is returned in a reply message packet along with another time stamp. Every packet includes three time stamps: an Originate (sent) Timestamp, a Receive Timestamp, and a Transmit (reply) Timestamp.

IP SLAs utilizes the time stamps to calculate jitter for each direction, based on the difference between interarrival and interdeparture delay for two successive packets. If the difference is positive, it is counted in positive jitter. A negative value is counted in negative jitter. Separate measurements for the source-to-destination and destination-to-source data paths can be used to identify problems in your network because the paths can be different (asymmetric).

Each ICMP packet includes a sequence number in its header that is used to count the number of packets received out of sequence on the sender. Both the sequence number and the receive timestamps can be used to calculate out-of-sequence packets on the source-to-destination path. If the receive time stamp for a packet is greater than that of the next packet, the first packet was delivered out of order on the source-to-destination path. For the destination-to-source path, the same method can be applied. Note that if the packet is out of order on the source-to-destination path, it should be returned out of order to the sender unless there is also misordering on the destination-to-source path.

If any packet cannot be sent due to an internal or unexpected error, or because the timerwheel slot containing the packet is missed, it is counted as Packet Skipped. This metric is very important because statistics are measured on sent packets only.

All timed-out packets are counted towards Packet Loss. Successive packet loss is calculated by counting, and adding, the number of successive dropped packets. Successive packet loss is reported as minimum of successive packet drop and maximum of successive packet drop.

All other statistics are calculated using the same logic as a UDP jitter operation.

How to Configure IP SLAs ICMP Jitter Operations

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	

Step 3	Enter one of the following commands:
	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`
		• Configures the scheduling parameters for an individual IP SLAs operation.
		• Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

• If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for

corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.

- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the "Configuring Proactive Threshold Monitoring" section.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS IP SLAs commands	IP SLAs Command Reference
Cisco IOS IP SLAs: general information	Cisco IOS IP SLAs Overview chapter of the *Cisco IOS IP SLAs Configuration Guide*.

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
• CISCO-RTTMON-MIB • CISCO-RTTMON-ICMP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 792	*Internet Control Message Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

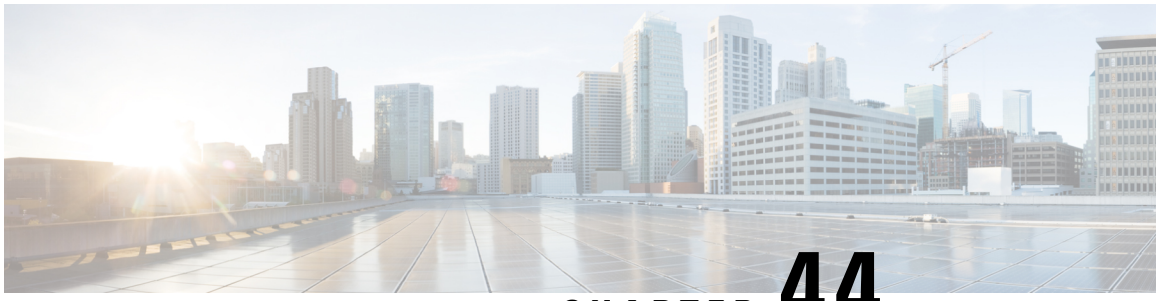
Feature Information for IP SLAs - ICMP Jitter Operation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 67: Feature Information for IP SLAs - ICMP Jitter Operation

Feature Name	Releases	Feature Information
IP SLAs ICMP Jitter Operation		The Cisco IOS IP Service Level Agreements (SLAs) Internet Control Message Protocol (ICMP) jitter operation provides the capability to generate a stream of ICMP packets between a Cisco IOS device (source) and any other IP device (destination) to gather network performance-related statistics. Available statistical measurements for the IP SLAs ICMP jitter operation include latency, round-trip time, jitter (interpacket delay variance), and packet loss.



CHAPTER 44

Configuring IP SLAs ICMP Echo Operations

This module describes how to configure an IP Service Level Agreements (SLAs) Internet Control Message Protocol (ICMP) Echo operation to monitor end-to-end response time between a Cisco router and devices using IPv4 or IPv6. ICMP Echo is useful for troubleshooting network connectivity issues. This module also demonstrates how the results of the ICMP Echo operation can be displayed and analyzed to determine how the network IP connections are performing.

Restrictions for IP SLAs ICMP Echo Operations

We recommend using a Cisco networking device as the destination device although any networking device that supports RFC 862, Echo protocol, can be used.

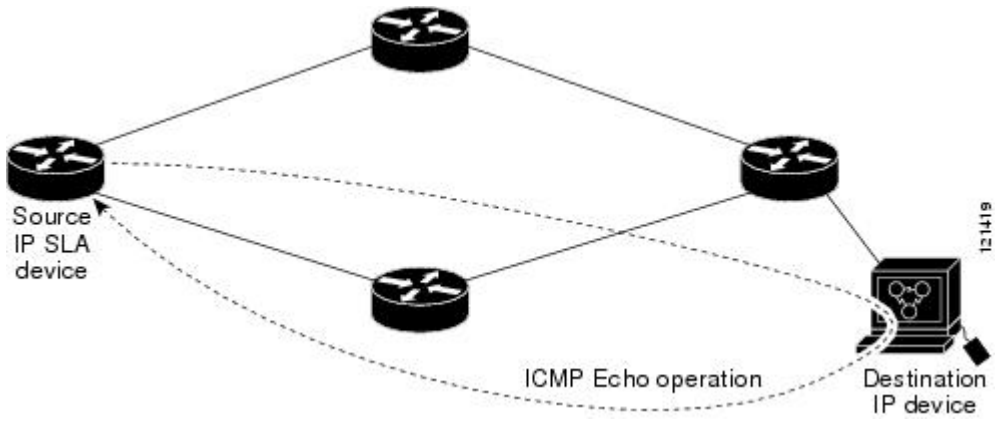
Information About IP SLAs ICMP Echo Operations

ICMP Echo Operation

The ICMP Echo operation measures end-to-end response time between a Cisco router and any devices using IP. Response time is computed by measuring the time taken between sending an ICMP Echo request message to the destination and receiving an ICMP Echo reply.

In the figure below ping is used by the ICMP Echo operation to measure the response time between the source IP SLAs device and the destination IP device. Many customers use IP SLAs ICMP-based operations, in-house ping testing, or ping-based dedicated probes for response time measurements.

Figure 44: ICMP Echo Operation



The IP SLAs ICMP Echo operation conforms to the same IETF specifications for ICMP ping testing and the two methods result in the same response times.

How to Configure IP SLAs ICMP Echo Operations

Configuring an ICMP Echo Operation

Note There is no need to configure an IP SLAs responder on the destination device.

Perform one of the following tasks:

Configuring a Basic ICMP Echo Operation on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-name*]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 6`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-name*] **Example:** `Device(config-ip-sla)# icmp-echo 172.29.139.134`	Defines an ICMP Echo operation and enters IP SLA ICMP Echo configuration mode.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-echo)# frequency 300`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-echo)# end`	Exits to privileged EXEC mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Configuring an ICMP Echo Operation with Optional Parameters

Perform this task on the source device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-name*]
5. **data-pattern** *hex value*
6. **history buckets-kept** *size*

7. **history distributions-of-statistics-kept** *size*
8. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
9. **history filter** {**none** | **all** | **overThreshold** | **failures**}
10. **frequency** *seconds*
11. **history hours-of-statistics-kept** *hours*
12. **history lives-kept** *lives*
13. **owner** *owner-id*
14. **request-data-size** *bytes*
15. **history statistics-distribution-interval** *milliseconds*
16. **tag** *text*
17. **threshold** *milliseconds*
18. **timeout** *milliseconds*
19. Do one of the following:
    - **tos** *number*
    - **traffic-class** *number*
20. **flow-label** *number*
21. **verify-data**
22. **vrf** *vrf-name*
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 6`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-name*] **Example:** `Device(config-ip-sla)# icmp-echo 172.29.139.134 source-ip 172.29.139.132`	Defines an Echo operation and enters IP SLA Echo configuration mode.

	Command or Action	Purpose
Step 5	**data-pattern** *hex value* **Example:** `Device(config-ip-sla-echo)# data pattern FFFFFFFF`	(Optional) Sets the hexadecimal value for data pattern. The range is 0 to FFFFFFFF.
Step 6	**history buckets-kept** *size* **Example:** `Device(config-ip-sla-echo)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 7	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-echo)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 8	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-echo)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 9	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Device(config-ip-sla-echo)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 10	**frequency** *seconds* **Example:** `Device(config-ip-sla-echo)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 11	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-echo)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 12	**history lives-kept** *lives* **Example:** `Device(config-ip-sla-echo)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 13	**owner** *owner-id* **Example:**	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.

	Command or Action	Purpose
	`Device(config-ip-sla-echo)# owner admin`	
Step 14	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-echo)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 15	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-echo)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 16	**tag** *text* **Example:** `Device(config-ip-sla-echo)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 17	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-echo)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 18	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-echo)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.

Step 19	Do one of the following:
	• **tos** *number*
	• **traffic-class** *number*
	Example:
	`Device(config-ip-sla-jitter)# tos 160`
	Example:
	`Device(config-ip-sla-jitter)# traffic-class 160`
		(Optional) In an IPv4 network only, defines the ToS byte in the IPv4 header of an IP SLAs operation.
		or
		(Optional) In an IPv6 network only, defines the traffic class byte in the IPv6 header for a supported IP SLAs operation.

Step 20	**flow-label** *number* **Example:** `Device(config-ip-sla-echo)# flow-label 112233`	(Optional) In an IPv6 network only, defines the flow label field in the IPv6 header for a supported IP SLAs operation.
Step 21	**verify-data** **Example:**	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.

	Command or Action	Purpose
	`Device(config-ip-sla-echo)# verify-data`	
Step 22	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-echo)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 23	**end** **Example:** `Device(config-ip-sla-echo)# end`	Exits to privileged EXEC mode.

What to do next

To add proactive threshold conditions and reactive triggering for generating traps, or for starting another operation, to an IP SLAs operation, see the "Configuring Proactive Threshold Monitoring" section.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`
		• Configures the scheduling parameters for an individual IP SLAs operation.
		• Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the "Configuring Proactive Threshold Monitoring" section.

Configuration Examples for IP SLAs ICMP Echo Operations

Example Configuring an ICMP Echo Operation

The following example shows how to configure an IP SLAs operation type of ICMP Echo that will start immediately and run indefinitely.

```
ip sla 6
 icmp-echo 172.29.139.134 source-ip 172.29.139.132
 frequency 300
 request-data-size 28
 tos 160
 timeout 2000
 tag SFO-RO
ip sla schedule 6 life forever start-time now
```

Additional References for IP SLAs ICMP Echo Operations

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP SLAs commands	Cisco IOS IP SLAs Command Reference
Information about Cisco IP SLAs	"Cisco IOS IP SLAs Overview" module of the *IP SLAs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 862	Echo Protocol

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

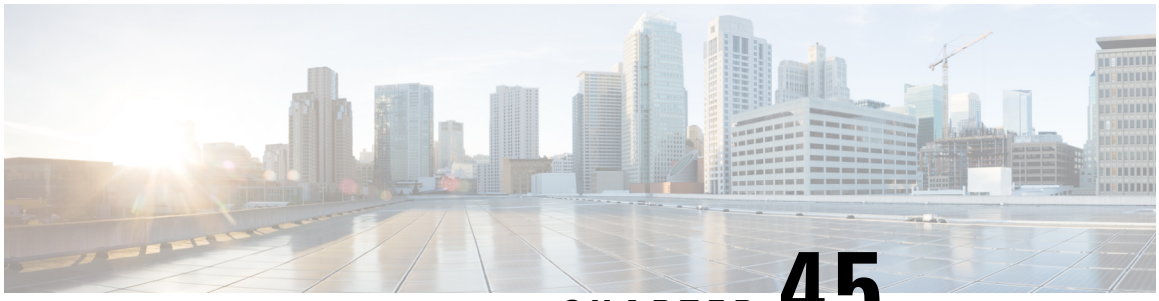
Feature Information for IP SLAs ICMP Echo Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 68: Feature Information for IP SLAs ICMP Echo Operations

Feature Name	Releases	Feature Information
IP SLAs ICMP Echo Operation		The Cisco IOS IP SLAs Internet Control Message Protocol (ICMP) echo operation allows you to measure end-to-end network response time between a Cisco device and other devices using IP.
IPv6 - IP SLAs (UDP Jitter, UDP Echo, ICMP Echo, TCP Connect)		Support was added for operability in IPv6 networks.



CHAPTER 45

Configuring IP SLAs ICMP Path Echo Operations

This module describes how to configure an IP Service Level Agreements (SLAs) Internet Control Message Protocol (ICMP) Path Echo operation to monitor end-to-end and hop-by-hop response time between a Cisco device and other devices using IP. ICMP Path Echo is useful for determining network availability and for troubleshooting network connectivity issues. The results of the ICMP Path Echo operation can be displayed and analyzed to determine how ICMP is performing.

Restrictions for IP SLAs ICMP Path Echo Operations

We recommend using a Cisco networking device as the destination device although any networking device that supports RFC 862, Echo protocol, can be used.

Information About IP SLAs ICMP Path Echo Operations

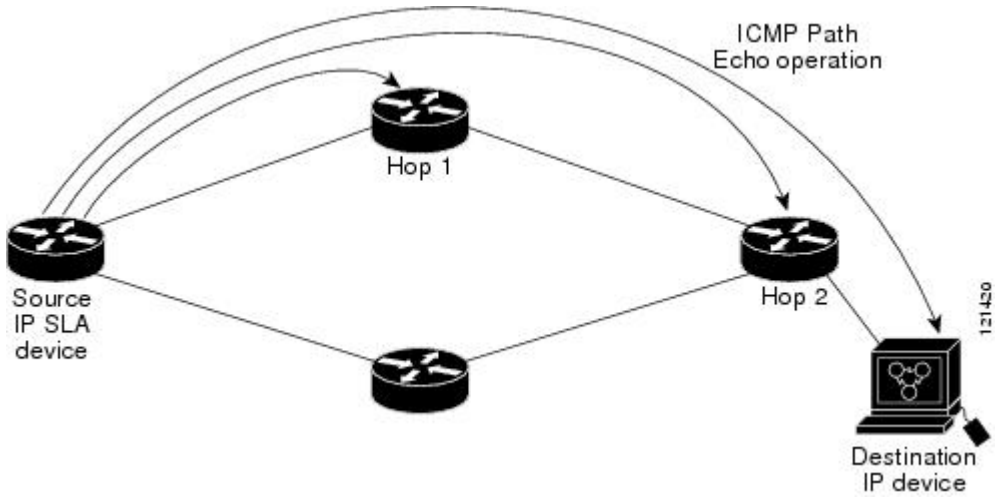
ICMP Path Echo Operation

To monitor ICMP Path Echo performance on a device, use the IP SLAs ICMP Path Echo operation. An ICMP Path Echo operation measures end-to-end and hop-by-hop response time between a Cisco device and other devices using IP. ICMP Path Echo is useful for determining network availability and for troubleshooting network connectivity issues.

The IP SLAs ICMP Path Echo operation records statistics for each hop along the path that the IP SLAs operation takes to reach its destination. The ICMP Path Echo operation determines this hop-by-hop response time between a Cisco device and any IP device on the network by discovering the path using the traceroute facility.

In the figure below the source IP SLAs device uses traceroute to discover the path to the destination IP device. A ping is then used to measure the response time between the source IP SLAs device and each subsequent hop in the path to the destination IP device.

Figure 45: ICMP Path Echo Operation



Using the statistics recorded for the response times and availability, the ICMP Path Echo operation can identify a hop in the path that is causing a bottleneck.

How to Configure IP SLAs ICMP Path Echo Operations

Configuring an ICMP Path Echo Operation on the Source Device

Note This operation does not require an IP SLAs Responder on the destination device.

Perform only one of the following tasks:

Configuring a Basic ICMP Path Echo Operation on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-id*
4. **path-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-id* **Example:** `Device(config)# ip sla 7`	Specifies an ID number for the operation being configured, and enters IP SLA configuration mode.
Step 4	**path-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] **Example:** `Device(config-ip-sla)# path-echo 172.29.139.134`	Defines a Path Echo operation and enters IP SLA Path Echo configuration mode.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-pathEcho)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-pathEcho)# end`	Exits to privileged EXEC mode.

Example

The following example shows the configuration of the IP SLAs ICMP Path Echo operation number 7 that will start in 30 seconds and run for 5 minutes.

```
ip sla 7
 path-echo 172.29.139.134
 frequency 30
!
ip sla schedule 7 start-time after 00:00:30 life 300
```

Configuring an ICMP Path Echo Operation with Optional Parameters on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip sla** *operation-number*
4. **path-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]
5. **history buckets-kept** *size*
6. **history distributions-of-statistics-kept** *size*
7. **history filter** {**none** | **all** | **overThreshold** | **failures**}
8. **frequency** *seconds*
9. **history hours-of-statistics-kept** *hours*
10. **history lives-kept** *lives*
11. **owner** *owner-id*
12. **paths-of-statistics-kept** *size*
13. **request-data-size** *bytes*
14. **samples-of-history-kept** *samples*
15. **history statistics-distribution-interval** *milliseconds*
16. **tag** *text*
17. **threshold** *milliseconds*
18. **timeout** *milliseconds*
19. **tos** *number*
20. **verify-data**
21. **vrf** *vrf-name*
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**path-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] **Example:** `Device(config-ip-sla)# path-echo 172.29.139.134`	Defines a Path Echo operation and enters IP SLA Path Echo configuration mode.

	Command or Action	Purpose
Step 5	**history buckets-kept** *size* **Example:** `Device(config-ip-sla-pathEcho)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-pathEcho)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 7	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Device(config-ip-sla-pathEcho)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 8	**frequency** *seconds* **Example:** `Device(config-ip-sla-pathEcho)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 9	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-pathEcho)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 10	**history lives-kept** *lives* **Example:** `Device(config-ip-sla-pathEcho)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 11	**owner** *owner-id* **Example:** `Device(config-ip-sla-pathEcho)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 12	**paths-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-pathEcho)# paths-of-statistics-kept 3`	(Optional) Sets the number of paths for which statistics are maintained per hour for an IP SLAs operation.
Step 13	**request-data-size** *bytes* **Example:**	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.

	Command or Action	Purpose
	`Device(config-ip-sla-pathEcho)# request-data-size 64`	
Step 14	**samples-of-history-kept** *samples* **Example:** `Device(config-ip-sla-pathEcho)# samples-of-history-kept 10`	(Optional) Sets the number of entries kept in the history table per bucket for an IP SLAs operation.
Step 15	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-pathEcho)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 16	**tag** *text* **Example:** `Device(config-ip-sla-pathEcho)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 17	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-pathEcho)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 18	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-pathEcho)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 19	**tos** *number* **Example:** `Device(config-ip-sla-pathEcho)# tos 160`	(Optional) Defines a type of service (ToS) byte in the IP header of an IP SLAs operation.
Step 20	**verify-data** **Example:** `Device(config-ip-sla-pathEcho)# verify-data`	(Optional) Causes an IP SLAs operation to check each reply packet for data corruption.
Step 21	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-pathEcho)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 22	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-ip-sla-pathEcho)# end`	

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose

	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

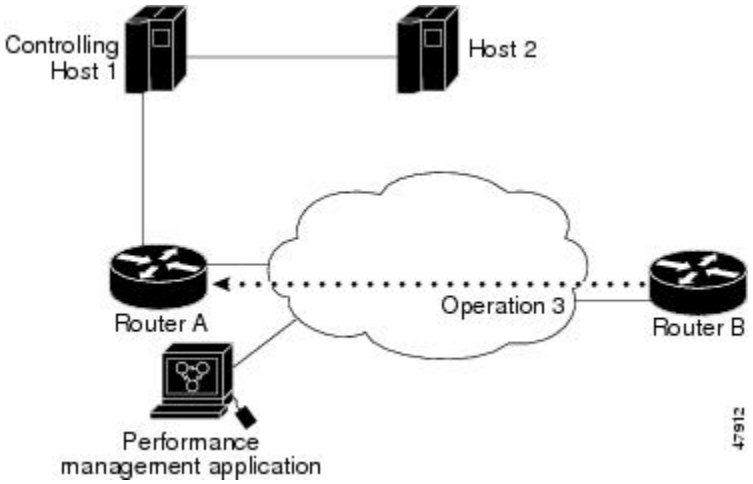
To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the "Configuring Proactive Threshold Monitoring" section.

Configuration Examples for IP SLAs ICMP Path Echo Operations

Example Configuring an ICMP Path Echo Operation

The following example shows how to configure an IP SLAs operation type of ICMP Path Echo that will start after 30 seconds and run for 5 minutes. The figure below depicts the ICMP Path Echo operation.

Figure 46: ICMP Path Echo Operation



This example sets a Path Echo operation (ip sla 3) from Device B to Device A using IP/ICMP. The operation attempts to execute three times in 25 seconds (first attempt at 0 seconds).

Device B Configuration

```
ip sla 3
 path-echo 172.29.139.134
 frequency 10
 tag SGN-RO
 timeout 1000
ip sla schedule 3 life 25
```

Additional References for IP SLAs ICMP Echo Operations

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP SLAs commands	Cisco IOS IP SLAs Command Reference
Information about Cisco IP SLAs	"Cisco IOS IP SLAs Overview" module of the *IP SLAs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 862	Echo Protocol

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

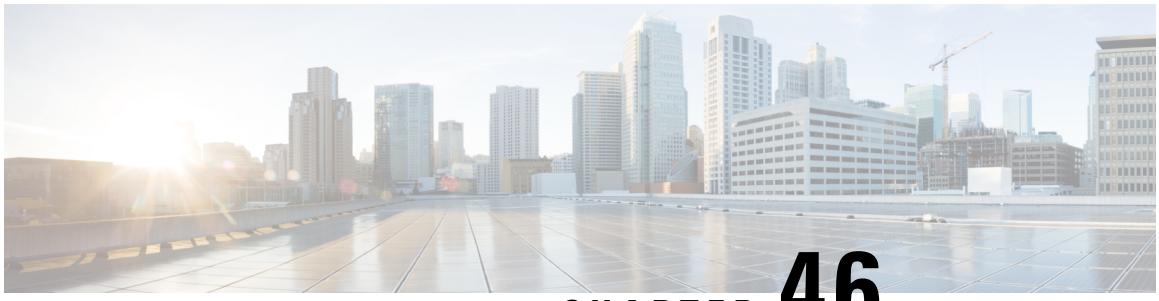
Feature Information for IP SLAs ICMP Path Echo Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 69: Feature Information for IP SLAs ICMP Path Echo Operations

Feature Name	Releases	Feature Information
IP SLAs ICMP Path Echo Operation	12.2(31)SB2 12.2(33)SRB1 12.2(33)SXH 12.3(14)T Cisco IOS XE Release 2.1 15.0(1)S Cisco IOS XE 3.1.0SG	The Cisco IOS IP SLAs Internet Control Message Protocol (ICMP) path echo operation allows you to measure end-to-end and hop-by-hop network response time between a Cisco device and other devices using IP.
IP SLA 4.0 - IP v6 phase2	15.2(3)T Cisco IOS XE Release 3.7S 15.1(2)SG Cisco IOS XE Release 3.4SG	Support was added for operability in IPv6 networks. The following commands are introduced or modified: **path-echo** (IP SLA), **show ip sla configuration**, **show ip sla summary**.



CHAPTER 46

Configuring IP SLAs ICMP Path Jitter Operations

This document describes how to configure an IP Service Level Agreements (SLAs) Internet Control Message Protocol (ICMP) Path Jitter operation to monitor hop-by-hop jitter (inter-packet delay variance). This document also demonstrates how the data gathered using the Path Jitter operations can be displayed and analyzed using Cisco commands.

Prerequisites for ICMP Path Jitter Operations

- Before configuring any IP SLAs application, you can use the **show ip sla application** command to verify that the operation type is supported on your software image.
- In contrast with other IP SLAs operations, the IP SLAs Responder does not have to be enabled on either the target device or intermediate devices for Path Jitter operations. However, the operational efficiency may improve if you enable the IP SLAs Responder.

Restrictions for ICMP Path Jitter Operations

- IP SLAs - ICMP Path Jitter is ICMP-based. ICMP-based operations can compensate for source processing delay but cannot compensate for target processing delay. For more robust monitoring and verifying, we recommend that you use the IP SLAs UDP Jitter operation.
- The jitter values obtained using IP SLAs - ICMP Path Jitter are approximates because ICMP does not provide the capability to embed processing times on devices in the packet. If the target device does not place ICMP packets as the highest priority, then the device will not respond properly. ICMP performance also can be affected by the configuration of priority queueing on the device and by ping response.
- A path jitter operation does not support hourly statistics and hop information.

- Unlike other IP SLAs operations, the ICMP Path Jitter operation is not supported in the RTTMON MIB. Path jitter operations can only be configured using Cisco commands and statistics can only be returned using the **show ip sla** commands.
- IP SLAs - Path Jitter does not support the IP SLAs History feature (statistics history buckets) because of the large data volume involved with jitter operations.
- The following commands, available in path jitter configuration mode, do not apply to path jitter operations:
  - **history buckets-kept**
  - **history distributions-of-statistics-kept**
  - **history enhanced**
  - **history filter**
  - **history hours-of-statistics-kept**
  - **history lives-kept**
  - **history statistics-distribution-interval**
  - **samples-of-history-kept**
  - **lsr-path**
  - **tos**
  - **threshold**
  - **verify-data**

Information About IP SLAs ICMP Path Jitter Operations

ICMP Path Jitter Operation

IP SLAs - ICMP Path Jitter provides hop-by-hop jitter, packet loss, and delay measurement statistics in an IP network. Path jitter operations function differently than the standard UDP Jitter operation, which provides total one-way data and total round-trip data.

An ICMP Path Jitter operation can be used a supplement to the standard UDP Jitter operation. For example, results from a UDP Jitter operation may indicate unexpected delays or high jitter values; an ICMP Path Jitter operation could then be used to troubleshoot the network path and determine if traffic is bottlenecking in a particular segment along the transmission path.

The operation first discovers the hop-by-hop IP route from the source to the destination using a traceroute utility, and then uses ICMP echoes to determine the response times, packet loss and approximate jitter values for each hop along the path. The jitter values obtained using IP SLAs - ICMP Path Jitter are approximates because ICMP only provides round trip times.

ICMP Path Jitter operations function by tracing the IP path from a source device to a specified destination device, then sending *N* number of Echo probes to each hop along the traced path, with a time interval of *T* milliseconds between each Echo probe. The operation as a whole is repeated at a frequency of once every *F* seconds. The attributes are user-configurable, as shown here:

Path Jitter Operation Parameter	Default	Configured Using:
Number of echo probes (*N*)	10 echos	**path-jitter** command, **num-packets** option

Path Jitter Operation Parameter	Default	Configured Using:
Time between Echo probes, in milliseconds (*T*)	20 ms	**path-jitter** command, **interval** option **Note** The operation's frequency is different than the operation's interval.
The frequency of how often the operation is repeated (*F*)	once every 60 seconds	**frequency** command

How to Configure the IP SLAs ICMP Path Jitter Operation

Configuring the IP SLAs Responder on a Destination Device

Note An IP SLAs Responder is not required on either the target device or intermediate devices for path jitter operations. However, operational efficiency may improve if you enable the IP SLAs Responder.

Before you begin

The networking device to be used as the responder must be a Cisco device and you must have connectivity to that device through the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla responder**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip sla responder** **Example:** **Example:** `Device(config)# ip sla responder`	(Optional) Temporarily enables IP SLAs Responder functionality on a Cisco device in response to control messages from source. • Control is enabled by default.
Step 4	**exit** **Example:** `Device(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring an ICMP Path Jitter Operation on the Source Device

Perform only one of the following procedures in this section:

Configuring a Basic ICMP Path Jitter Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **path-jitter** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] [**num-packets** *packet-number*] [**interval** *milliseconds*] [**targetOnly**]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.

	Command or Action	Purpose
Step 4	**path-jitter** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] [**num-packets** *packet-number*] [**interval** *milliseconds*] [**targetOnly**] **Example:** `Device(config-ip-sla)# path-jitter 172.31.1.129 source-ip 10.2.30.1 num-packets 12 interval 22`	Enters IP SLA Path Jitter configuration mode for configuring an ICMP Path Jitter operation.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-pathJitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-pathJitter)# end`	Exits to privileged EXEC mode.

Example

In the following example, the **targetOnly** keyword is used to bypass the hop-by-hop measurements. With this version of the command, echo probes will be sent to the destination only.

```
Device(config)# ip sla 1
Device(config-ip-sla)# path-jitter 172.17.246.20 num-packets 50 interval 30 targetOnly
```

Configuring an ICMP Path Jitter Operation with Additional Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **path-jitter** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] [**num-packets** *packet-number*] [**interval** *milliseconds*] [**targetOnly**]
5. **frequency** *seconds*
6. **owner** *owner-id*
7. **request-data-size** *bytes*
8. **tag** *text*
9. **timeout** *milliseconds*
10. **vrf** *vrf-name*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**path-jitter** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] [**num-packets** *packet-number*] [**interval** *milliseconds*] [**targetOnly**] **Example:** `Device(config-ip-sla)# path-jitter 172.31.1.129 source-ip 10.2.30.1 num-packets 12 interval 22`	Enters IP SLA Path Jitter configuration mode for defing an ICMP Path Jitter operation.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-pathJitter)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**owner** *owner-id* **Example:** `Device(config-ip-sla-pathJitter)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 7	**request-data-size** *bytes* **Example:** `Device(config-ip-sla-pathJitter)# request-data-size 64`	(Optional) Sets the protocol data size in the payload of an IP SLAs operation's request packet.
Step 8	**tag** *text* **Example:** `Device(config-ip-sla-pathJitter)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.

	Command or Action	Purpose
Step 9	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-pathJitter)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 10	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-pathJitter)# vrf vpn-A`	(Optional) Allows monitoring within Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) using IP SLAs operations.
Step 11	**end** **Example:** `Device(config-ip-sla-pathJitter)# end`	Exits to privileged EXEC mode.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`
		• Configures the scheduling parameters for an individual IP SLAs operation.
		• Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the "Configuring Proactive Threshold Monitoring" section.

Configuration Examples for IP SLAs ICMP Path Jitter Operations

Example Configuring a Path Jitter Operation

The following example shows the output when the ICMP Path Jitter operation is configured. Because the path jitter operation does not support hourly statistics and hop information, the output for the **show ip sla statistics** command for the path jitter operation displays only the statistics for the first hop.

The following example shows the output when the ICMP Path Jitter operation is configured.

```
Device# configure terminal
Device(config)# ip sla 15011
Device(config-sla-monitor)# path-jitter 10.222.1.100 source-ip 10.222.3.100 num-packets 20
Device(config-sla-monitor-pathJitter)# frequency 30
Device(config-sla-monitor-pathJitter)# exit
Device(config)# ip sla schedule 15011 life forever start-time now
Device(config)# exit
Device# show ip sla statistics 15011
Round Trip Time (RTT) for       Index 15011
        Latest RTT: 1 milliseconds
Latest operation start time: 15:37:35.443 EDT Mon Jun 16 2008
Latest operation return code: OK
---- Path Jitter Statistics ----
Hop IP 10.222.3.252:
Round Trip Time milliseconds:
        Latest RTT: 1 ms
        Number of RTT: 20
        RTT Min/Avg/Max: 1/1/3 ms
Jitter time milliseconds:
        Number of jitter: 2
        Jitter Min/Avg/Max: 2/2/2 ms
Packet Values:
        Packet Loss (Timeouts): 0
        Out of Sequence: 0
        Discarded Samples: 0
Operation time to live: Forever
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1889[11]	*RTP: A Transport Protocol for Real-Time Applications* ; see the section "Estimating the Interarrival Jitter"

[11] Support for the listed RFC is not claimed; listed as a reference only.

MIBs

MIBs	MIBs Link
MIB support for the Path Jitter operation is not provided.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

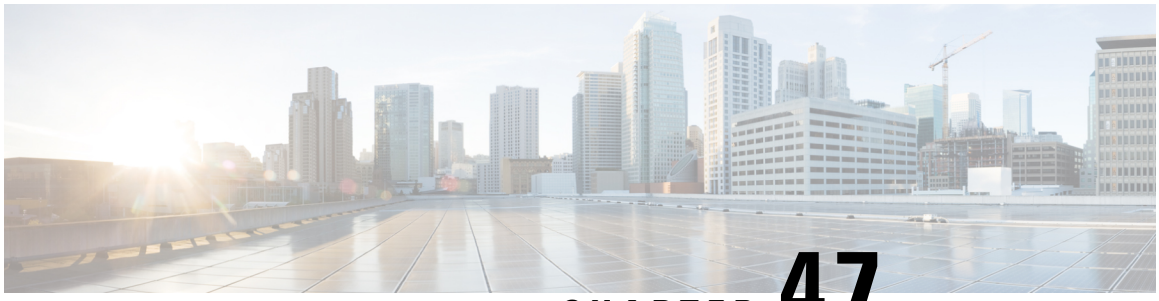
Feature Information for IP SLAs ICMP Path Jitter Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 70: Feature Information for IP SLAs ICMP Path Jitter Operations

Feature Name	Releases	Feature Information
IP SLAs Path Jitter Operation		The Cisco IOS IP SLAs Internet Control Message Protocol (ICMP) path jitter operation allows you to measure hop-by-hop jitter (inter-packet delay variance).
IPSLA 4.0 - IP v6 phase2		Support was added for operability in IPv6 networks. The following commands are introduced or modified: **path- jitter**, **show ip sla configuration**, **show ip sla summary**.



CHAPTER 47

Configuring IP SLAs FTP Operations

This module describes how to configure an IP Service Level Agreements (SLAs) File Transfer Protocol (FTP) operation to measure the response time between a Cisco device and an FTP server to retrieve a file. The IP SLAs FTP operation supports an FTP GET request only. This module also demonstrates how the results of the FTP operation can be displayed and analyzed to determine the capacity of your network. The FTP operation can be used also for troubleshooting FTP server performance.

Restrictions for IP SLAs FTP Operations

The IP SLAs FTP operation only supports FTP GET (download) requests.

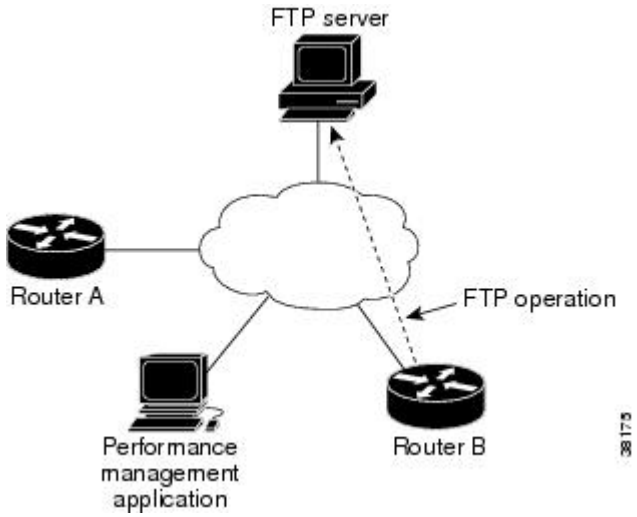
Information About IP SLAs FTP Operations

FTP Operation

The FTP operation measures the round-trip time (RTT) between a Cisco device and an FTP server to retrieve a file. FTP is an application protocol, part of the Transmission Control Protocol (TCP)/IP protocol stack, used for transferring files between network nodes.

In the figure below Device B is configured as the source IP SLAs device and an FTP operation is configured with the FTP server as the destination device.

Figure 47: FTP Operation



Connection response time is computed by measuring the time taken to download a file to Device B from the remote FTP server using FTP over TCP. This operation does not use the IP SLAs Responder.

Note To test the response time to connect to an FTP port (Port 21), use the IP SLAs TCP Connect operation.

Both active and passive FTP transfer modes are supported. The passive mode is enabled by default. Only the FTP GET (download) operation type is supported. The URL specified for the FTP GET operation must be in one of the following formats:

- ftp://username:password@host/filename
- ftp://host/filename

If the username and password are not specified, the defaults are anonymous and test, respectively.

FTP carries a significant amount of data traffic and can affect the performance of your network. The results of an IP SLAs FTP operation to retrieve a large file can be used to determine the capacity of the network but retrieve large files with caution because the FTP operation will consume more bandwidth. The FTP operation also measures your FTP server performance levels by determining the RTT taken to retrieve a file.

How to Configure IP SLAs FTP Operations

Configuring an FTP Operation on a Source Device

Note There is no need to configure an IP SLAs responder on the destination device.

Perform one of the following tasks:

Configuring a Basic FTP Operation on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ftp get** *url* [**source-ip** {*ip-address* | *hostname*}] [**mode** {**passive** | **active**}
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ftp get** *url* [**source-ip** {*ip-address* | *hostname*}] [**mode** {**passive** | **active**} **Example:** `Device(config-ip-sla)# ftp get ftp://username:password@hostip/test.cap`	Defines an FTP operation and enters IP SLA FTP configuration mode.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-ftp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-ftp)# exit`	Exits to privileged EXEC mode.

Configuring an FTP Operation with Optional Parameters on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ftp get** *url* [**source-ip** {*ip-address* | *hostname*}] [**mode** {**passive** | **active**}
5. **history buckets-kept** *size*
6. **history distributions-of-statistics-kept** *size*
7. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]
8. **history filter** {**none** | **all** | **overThreshold** | **failures**}
9. **frequency** *seconds*
10. **history hours-of-statistics-kept** *hours*
11. **history lives-kept** *lives*
12. **owner** *owner-id*
13. **history statistics-distribution-interval** *milliseconds*
14. **tag** *text*
15. **threshold** *milliseconds*
16. **timeout** *milliseconds*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ftp get** *url* [**source-ip** {*ip-address* | *hostname*}] [**mode** {**passive** | **active**} **Example:** `Device(config-ip-sla)# ftp get ftp://username:password@hostip/filename`	Defines an FTP operation and enters IP SLA FTP configuration mode.

	Command or Action	Purpose
Step 5	**history buckets-kept** *size* **Example:** `Device(config-ip-sla-ftp)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-ftp)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 7	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-ftp)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 8	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Device(config-ip-sla-ftp)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 9	**frequency** *seconds* **Example:** `Device(config-ip-sla-ftp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 10	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-ftp)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 11	**history lives-kept** *lives* **Example:** `Device(config-ip-sla-ftp)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 12	**owner** *owner-id* **Example:** `Device(config-ip-sla-ftp)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 13	**history statistics-distribution-interval** *milliseconds* **Example:**	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.

	Command or Action	Purpose
	`Device(config-ip-sla-ftp)# history statistics-distribution-interval 10`	
Step 14	**tag** *text* **Example:** `Device(config-ip-sla-ftp)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 15	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-ftp)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 16	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-ftp)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 17	**end** **Example:** `Device(config-ip-sla-ftp)# end`	Exits to privileged EXEC mode.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**

5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`
		• Configures the scheduling parameters for an individual IP SLAs operation.
		• Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:**	(Optional) Displays IP SLAs group schedule details.

	Command or Action	Purpose
	`Device# show ip sla group schedule`	
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Configuration Examples for IP SLAs FTP Operations

Example: Configuring an FTP Operation

The following example shows how to configure an FTP operation from Device B to the FTP server as shown in the "FTP Operation" figure in the "Information About IP SLAs FTP Operation" section. The operation is scheduled to start every day at 1:30 a.m. In this example, the file named test.cap is to be retrieved from the host, cisco.com, with a password of abc using FTP in active mode.

Device B Configuration

```
ip sla 10
 ftp get ftp://user1:abc@test.cisco.com/test.cap mode active
 frequency 20
 tos 128
 timeout 40000
 tag FLL-FTP
ip sla schedule 10 start-time 01:30:00 recurring
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco ACNS software configuration information	• *Cisco ACNS Software Caching Configuration Guide, Release 4.2* • Cisco ACNS Software listing page on Cisco.com
IP access list overview, configuration tasks, and commands	*Cisco IOS Security Command Reference*
IP addressing and services commands and configuration tasks	• *Cisco IOS IP Addressing Services Configuration Guide* • *Cisco IOS IP Addressing Services Command Reference*
WCCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

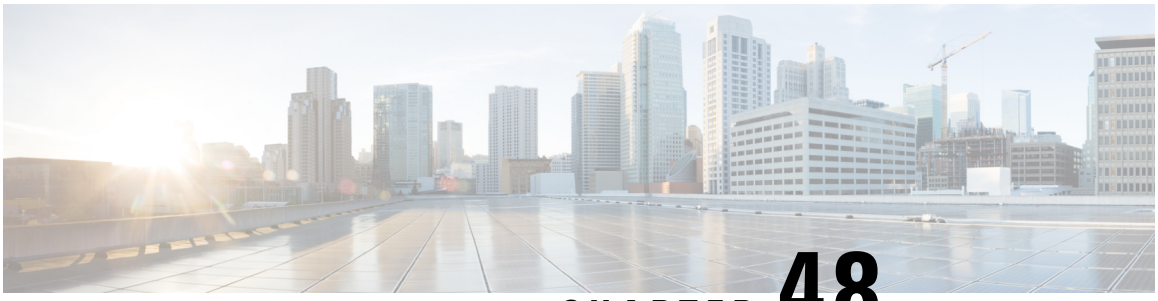
Feature Information for Configuring IP SLAs FTP Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 71: Feature Information for the IP SLAs FTP Operation

Feature Name	Releases	Feature Information
IP SLAs - FTP Operation	12.2(31)SB2 12.2(33)SRB1 12.2(33)SXH 12.3(14)T Cisco IOS XE Release 2.1 15.0(1)S Cisco IOS XE Release 3.1.0SG	The IP SLAs File Transfer Protocol (FTP) operation allows you to measure the network response time between a Cisco device and an FTP server to retrieve a file.
IPSLA 4.0 - IP v6 phase2	15.2(3)T 15.2(4)S Cisco IOS XE release XE 3.7S 15.1(2)SG Cisco IOS XE Release 3.4SG	Support was added for operability in IPv6 networks. The following commands are introduced or modified: **ftp get** (IP SLA), **show ip sla configuration**, **show ip sla summary**.
IP SLAs VRF Aware 2.0	12.4(2)T 15.1(1)S 15.1(1)SY Cisco IOS XE Release 3.8S	Support was added for IP SLAs VRF-aware capabilities for TCP connect, FTP, HTTP and DNS client operation types.



CHAPTER 48

Configuring IP SLAs DNS Operations

This module describes how to configure the IP Service Level Agreements (SLAs) Domain Name System (DNS) operation to measure the difference between the time taken to send a DNS request and receive a reply. This module also demonstrates how the results of the DNS operation can be displayed and analyzed to determine the DNS lookup time which is a critical element for determining the performance of a DNS or web server.

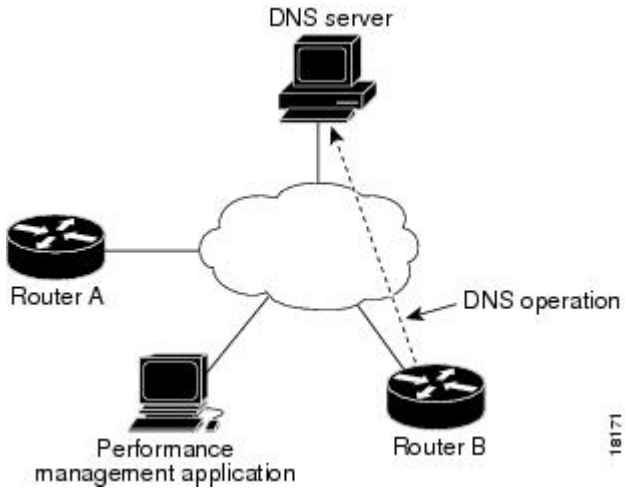
Information About IP SLAs DNS Operations

DNS Operation

The DNS operation measures the difference between the time taken to send a DNS request and receive a reply. DNS is used in the Internet for translating names of network nodes into addresses. The IP SLAs DNS operation queries for an IP address if you specify a host name, or queries for a host name if you specify an IP address.

In the figure below Device B is configured as the source IP SLAs device and a DNS operation is configured with the DNS server as the destination device.

Figure 48: DNS Operation



Connection response time is computed by measuring the difference between the time taken to send a request to the DNS server and the time a reply is received by Device B. The resulting DNS lookup time can help you analyze your DNS performance. Faster DNS lookup times translate to a faster web server access experience.

How to Configure IP SLAs DNS Operations

Configuring an IP SLAs DNS Operation on the Source Device

Note There is no need to configure an IP SLAs responder on the destination device.

Perform one of the following tasks:

Configuring a Basic DNS Operation on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] **Example:** `Device(config-ip-sla)# dns host1 name-server 172.20.2.132`	Defines a DNS operation and enters IP SLA DNS configuration mode.
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-dns)# frequency 60`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-dns)# end`	Exits to privileged EXEC mode.

Configuring a DNS Operation with Optional Parameters on the Source Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*]
5. **history buckets-kept** *size*
6. **history distributions-of-statistics-kept** *size*
7. **history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*]

8. **history filter** {**none** | **all** | **overThreshold** | **failures**}
9. **frequency** *seconds*
10. **history hours-of-statistics-kept** *hours*
11. **history lives-kept** *lives*
12. **owner** *owner-id*
13. **history statistics-distribution-interval** *milliseconds*
14. **tag** *text*
15. **threshold** *milliseconds*
16. **timeout** *milliseconds*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**dns** {*destination-ip-address* | *destination-hostname*} **name-server** *ip-address* [**source-ip** {*ip-address* | *hostname*} **source-port** *port-number*] **Example:** `Device(config-ip-sla)# dns host1 name-server 172.20.2.132`	Defines a DNS operation and enters IP SLA DNS configuration mode.
Step 5	**history buckets-kept** *size* **Example:** `Device(config-ip-sla-dns)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-dns)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.

	Command or Action	Purpose
Step 7	**history enhanced** [**interval** *seconds*] [**buckets** *number-of-buckets*] **Example:** `Device(config-ip-sla-dns)# history enhanced interval 900 buckets 100`	(Optional) Enables enhanced history gathering for an IP SLAs operation.
Step 8	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Device(config-ip-sla-dns)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 9	**frequency** *seconds* **Example:** `Device(config-ip-sla-dns)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 10	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-dns)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 11	**history lives-kept** *lives* **Example:** `Device(config-ip-sla-dns)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 12	**owner** *owner-id* **Example:** `Device(config-ip-sla-dns)# owner admin`	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.
Step 13	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-dns)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 14	**tag** *text* **Example:** `Device(config-ip-sla-dns)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.

	Command or Action	Purpose
Step 15	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-dns)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 16	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-dns)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 17	**end** **Example:** `Device(config-ip-sla-dns)# end`	Exits to privileged EXEC mode.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
   - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Enter one of the following commands:
	• **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
	• **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla group schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`
		• Configures the scheduling parameters for an individual IP SLAs operation.
		• Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the "Configuring Proactive Threshold Monitoring" section.

Configuration Examples for IP SLAs DNS Operations

Example Configuring a DNS Operation

The following example shows how to configure a DNS operation from Device B to the DNS server (IP address 172.20.2.132) as shown in the "DNS Operation" figure in the "DNS Operation" section. The operation is scheduled to start immediately. In this example, the target address is a hostname and the DNS operation will query the DNS server for the IP address associated with the hostname host1. No configuration is required at the DNS server.

Device B Configuration

```
ip sla 11
 dns host1 name-server 172.20.2.132
 frequency 50
 timeout 8000
 tag DNS-Test
ip sla schedule 11 start-time now
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference, All Releases
Cisco IOS IP SLAs: general information	"Cisco IOS IP SLAs Overview" module of the *Cisco IOS IP SLAs Configuration Guide*.

Related Topic	Document Title
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" module of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

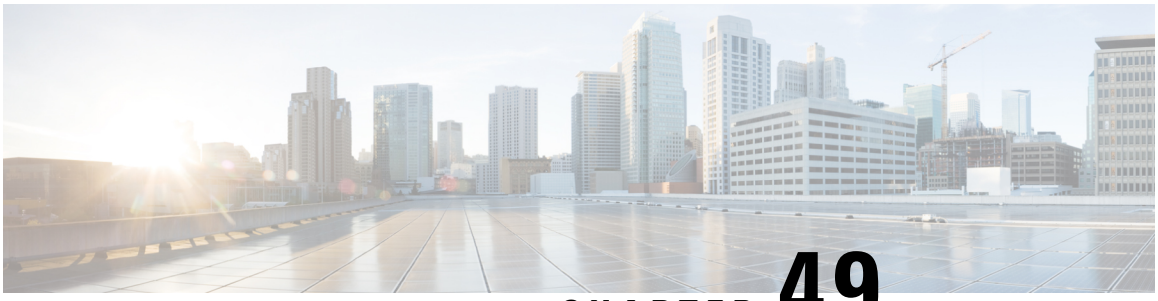
Feature Information for Configuring IP SLAs DNS Operation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 72: Feature Information for the IP SLAs - DNS Operation

Feature Name	Releases	Feature Information
IP SLAs - DNS Operation	12.2(31)SB2 12.2(33)SRB1 12.2(33)SXH 12.3(14)T Cisco IOS XE Release 2.1 15.0(1)S Cisco IOS XE 3.1.0SG	The IP SLAs Domain Name System (DNS) Operation feature allows you to measure the difference between the time taken to send a DNS request and receive a reply.
IPSLA 4.0 - IP v6 phase2	15.2(3)T Cisco IOS XE Release 3.7S 15.1(2)SG Cisco IOS XE Release 3.4SG	Support was added for operability in IPv6 networks. The following commands are introduced or modified: **dns (IP SLA)**, **show ip sla configuration**, **show ip sla summary**.
IP SLAs VRF Aware 2.0	12.4(2)T 15.1(1)S 15.1(1)SY Cisco IOS XE Release 3.8S	Support was added for IP SLAs VRF-aware capabilities for TCP connect, FTP, HTTP and DNS client operation types.



CHAPTER 49

Configuring IP SLAs DHCP Operations

This module describes how to configure an IP Service Level Agreements (SLAs) Dynamic Host Control Protocol (DHCP) probe to measure the response time between a Cisco device and a DHCP server to obtain an IP address.

Information About IP SLAs DHCP Operations

DHCP Operation

DHCP provides a mechanism for allocating IP addresses dynamically so that addresses can be reused when hosts no longer need them. The DHCP operation measures the round-trip time (RTT) taken to discover a DHCP server and obtain a leased IP address from it. IP SLAs releases the leased IP address after the operation.

You can use the RTT information to determine DHCP performance levels.

There are two modes for the DHCP operation. By default, the DHCP operation sends discovery packets on every available IP interface on the device. If a specific server is configured on the device, discovery packets are sent only to the specified DHCP server.

IP SLAs DHCP Relay Agent Options

A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP device, where IP packets are switched between networks somewhat transparently. Relay agents receive DHCP messages and then generate a new DHCP message to send out on another interface.

How to Configure IP SLAs DHCP Operations

Note There is no need to configure an IP SLAs responder on the destination device.

Configuring a DHCP Operation on the Source Device

Perform one of the following tasks:

Configuring a Basic DHCP Operation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **dhcp** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]
5. **frequency** *seconds*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**dhcp** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] **Example:** `Device(config-ip-sla)# dhcp 10.10.10.3`	Defines a DHCP operation and enters IP SLA DHCP configuration mode.

	Command or Action	Purpose
Step 5	**frequency** *seconds* **Example:** `Device(config-ip-sla-dhcp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 6	**end** **Example:** `Device(config-ip-sla-dhcp)# end`	Exits to privileged EXEC mode.

Configuring a DHCP Operation with Optional Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **dhcp** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}]
5. **history buckets-kept** *size*
6. **history distributions-of-statistics-kept** *size*
7. **history filter** {**none** | **all** | **overThreshold** | **failures**}
8. **frequency** *seconds*
9. **history hours-of-statistics-kept** *hours*
10. **history lives-kept** *lives*
11. **owner** *owner-id*
12. **history statistics-distribution-interval** *milliseconds*
13. **tag** *text*
14. **threshold** *milliseconds*
15. **timeout** *milliseconds*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 10`	Begins configuration for an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**dhcp** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*}] **Example:** `Device(config-ip-sla)# dhcp 10.10.10.3`	Defines a DHCP operation and enters IP SLA DHCP configuration mode.
Step 5	**history buckets-kept** *size* **Example:** `Device(config-ip-sla-dhcp)# history buckets-kept 25`	(Optional) Sets the number of history buckets that are kept during the lifetime of an IP SLAs operation.
Step 6	**history distributions-of-statistics-kept** *size* **Example:** `Device(config-ip-sla-dhcp)# history distributions-of-statistics-kept 5`	(Optional) Sets the number of statistics distributions kept per hop during an IP SLAs operation.
Step 7	**history filter** {**none** | **all** | **overThreshold** | **failures**} **Example:** `Device(config-ip-sla-dhcp)# history filter failures`	(Optional) Defines the type of information kept in the history table for an IP SLAs operation.
Step 8	**frequency** *seconds* **Example:** `Device(config-ip-sla-dhcp)# frequency 30`	(Optional) Sets the rate at which a specified IP SLAs operation repeats.
Step 9	**history hours-of-statistics-kept** *hours* **Example:** `Device(config-ip-sla-dhcp)# history hours-of-statistics-kept 4`	(Optional) Sets the number of hours for which statistics are maintained for an IP SLAs operation.
Step 10	**history lives-kept** *lives* **Example:** `Device(config-ip-sla-dhcp)# history lives-kept 5`	(Optional) Sets the number of lives maintained in the history table for an IP SLAs operation.
Step 11	**owner** *owner-id* **Example:**	(Optional) Configures the Simple Network Management Protocol (SNMP) owner of an IP SLAs operation.

	Command or Action	Purpose
	`Device(config-ip-sla-dhcp)# owner admin`	
Step 12	**history statistics-distribution-interval** *milliseconds* **Example:** `Device(config-ip-sla-dhcp)# history statistics-distribution-interval 10`	(Optional) Sets the time interval for each statistics distribution kept for an IP SLAs operation.
Step 13	**tag** *text* **Example:** `Device(config-ip-sla-dhcp)# tag TelnetPollServer1`	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 14	**threshold** *milliseconds* **Example:** `Device(config-ip-sla-dhcp)# threshold 10000`	(Optional) Sets the upper threshold value for calculating network monitoring statistics created by an IP SLAs operation.
Step 15	**timeout** *milliseconds* **Example:** `Device(config-ip-sla-dhcp)# timeout 10000`	(Optional) Sets the amount of time an IP SLAs operation waits for a response from its request packet.
Step 16	**end** **Example:** `Device(config-ip-sla-dhcp)# end`	Exits to privileged EXEC mode.

Scheduling IP SLAs Operations

Before you begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]

- **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]

4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever**	*seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range*	**schedule-together**} [**ageout** *seconds*] **frequency** *group-operation-frequency* [**life** {**forever**	*seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh*:*mm* [:*ss*]}] **Example:** `Device(config)# ip sla schedule 10 life forever start-time now` `Device(config)# ip sla group schedule 10 schedule-period frequency` `Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now` `Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.
Step 4	**end** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Troubleshooting Tips

- If the IP Service Level Agreements (SLAs) operation is not running and not generating statistics, add the **verify-data** command to the configuration (while configuring in IP SLA configuration mode) to enable data verification. When data verification is enabled, each operation response is checked for corruption. Use the **verify-data** command with caution during normal operations because it generates unnecessary overhead.
- Use the **debug ip sla trace** and **debug ip sla error** commands to help troubleshoot issues with an IP SLAs operation.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps (or for starting another operation) to an IP Service Level Agreements (SLAs) operation, see the “Configuring Proactive Threshold Monitoring” section.

Configuration Examples for IP SLAs DHCP Operations

Example Configuration for an IP SLAs DHCP Operation

In the following example, IP SLAs operation number 12 is configured as a DHCP operation enabled for DHCP server 172.16.20.3. Note that DHCP option 82 is used to specify the circuit ID.

Device B Configuration

```
ip dhcp-server 172.16.20.3
!
ip sla 12
 dhcp 10.10.10.3
 frequency 30
 timeout 5000
 tag DHCP_Test
!
ip sla schedule 12 start-time now
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference, All Releases
Cisco IOS IP SLAs: general information	"Cisco IOS IP SLAs Overview" module of the *Cisco IOS IP SLAs Configuration Guide*.
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" module of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

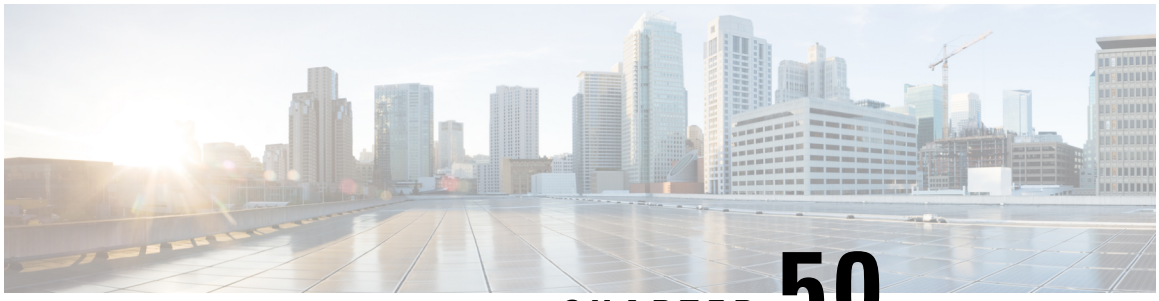
Feature Information for IP SLAs DHCP Operations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 73: Feature Information for IP SLAs DHCP Operations

Feature Name	Releases	Feature Information
IP SLAs DHCP Probe		The IP SLAs Dynamic Host Control Protocol (DHCP) Probe feature allows you to schedule and measure the network response time between a Cisco device and a DHCP server to obtain an IP address.



CHAPTER 50

Configuring an IP SLAs Multioperation Scheduler

This document describes how to schedule multiple operations at once using the IP Service Level Agreements (SLAs) Multioperations Scheduler feature.

Restrictions for an IP SLAs Multioperation Scheduler

Do not use the **no ip sla group schedule** and **ip sla group schedule** commands consecutively in a configuration file and copy it into the running configuration. This causes some of the Service Level Agreement (SLA) probes to go down.

Prerequisites for an IP SLAs Multioperation Scheduler

- Configure the IP SLAs operations to be included in a group before scheduling the group.
- Determine the IP SLAs operations you want to schedule as a single group.
- Identify the network traffic type and the location of your network management station.
- Identify the topology and the types of devices in your network.
- Decide on the frequency of testing for each operation.

Information About an IP SLAs Multioperation Scheduler

IP SLAs Multioperations Scheduler

Normal scheduling of IP SLAs operations allows you to schedule one operation at a time. If you have large networks with thousands of IP SLAs operations to monitor network performance, normal scheduling (scheduling each operation individually) will be inefficient and time-consuming.

Multiple operations scheduling allows you to schedule multiple IP SLAs operations using a single command through the command line interface (CLI) or the CISCO-RTTMON-MIB. This feature allows you to control the amount of IP SLAs monitoring traffic by scheduling the operations to run at evenly distributed times. You must specify the operation ID numbers to be scheduled and the time range over which all the IP SLAs operations should start. This feature automatically distributes the IP SLAs operations at equal intervals over a specified time frame. The spacing between the operations (start interval) is calculated and the operations are started. This distribution of IP SLAs operations helps minimize the CPU utilization and thereby enhances the scalability of the network.

The IP SLAs multiple operations scheduling functionality allows you to schedule multiple IP SLAs operations as a group, using the following configuration parameters:

- Group operation number--Group configuration or group schedule number of the IP SLAs operation to be scheduled.
- Operation ID numbers--A list of IP SLAs operation ID numbers in the scheduled operation group.
- Schedule period--Amount of time for which the IP SLAs operation group is scheduled.
- Ageout--Amount of time to keep the operation in memory when it is not actively collecting information. By default, the operation remains in memory indefinitely.
- Frequency--Amount of time after which each IP SLAs operation is restarted. When the frequency option is specified, it overwrites the operation frequency of all operations belonging to the group. Note that when the frequency option is not specified, the frequency for each operation is set to the value of the schedule period.
- Life--Amount of time the operation actively collects information. The operation can be configured to run indefinitely. By default, the lifetime of an operation is one hour.
- Start time--Time when the operation starts collecting information. You can specify an operation to start immediately or at an absolute start time using hours, minutes, seconds, day, and month.

The IP SLAs multiple operations scheduling functionality schedules the maximum number of operations possible without terminating. However, this functionality skips those IP SLAs operations that are already running or those that are not configured and hence do not exist. The total number of operations will be calculated based on the number of operations specified in the command, irrespective of the number of operations that are missing or already running. The IP SLAs multiple operations scheduling functionality displays a message showing the number of active and missing operations. However, these messages are displayed only if you schedule operations that are not configured or are already running.

A main benefit for scheduling multiple IP SLAs operations is that the load on the network is reduced by distributing the operations equally over a scheduled period. This distribution helps you to achieve more consistent monitoring coverage. To illustrate this scenario, consider configuring 60 operations to start during the same 1-second interval over a 60-second schedule period. If a network failure occurs 30 seconds after all

60 operations have started and the network is restored before the operations are due to start again (in another 30 seconds), then this failure would never be detected by any of the 60 operations. However, if the 60 operations are distributed equally at 1-second intervals over a 60-second schedule period, then some of the operations would detect the network failure. Conversely, if a network failure occurs when all 60 operations are active, then all 60 operations would fail, indicating that the failure is possibly more severe than it really is.

Operations of the same type and same frequency should be used for IP SLAs multiple operations scheduling. If you do not specify a frequency, the default frequency will be the same as that of the schedule period. The schedule period is the period of time in which all the specified operations should run.

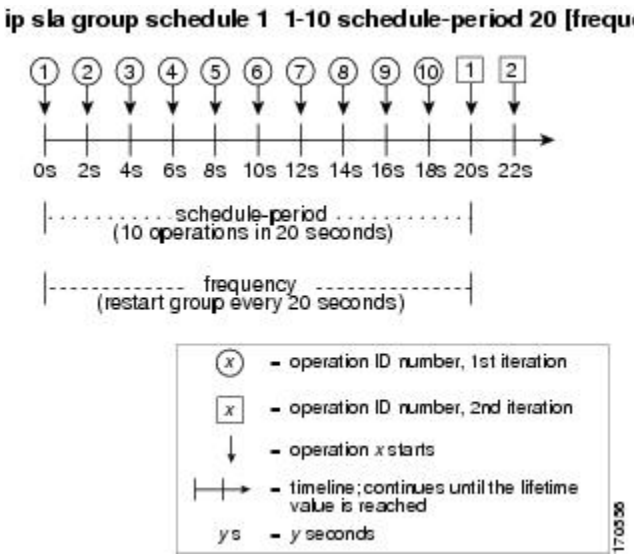
The following sections focus on the interaction of the schedule period and frequency values, additional values, such as start time and lifetime values, are not included in the illustrations.

Default Behavior of IP SLAs Multiple Operations Scheduling

The IP SLAs Multiple Operations Scheduling feature allows you to schedule multiple IP SLAs operations as a group.

The figure below illustrates the scheduling of operation group 1 that includes operation 1 to operation 10. Operation group 1 has a schedule period of 20 seconds, which means that all operations in the group will be started at equal intervals within a 20-second period. By default, the frequency is set to the same value as the configured schedule period. As shown in the figure below, configuring the frequencyis optional because 20 is the default.

Figure 49: Schedule Period Equals Frequency--Default Behavior



In this example, the first operation (operation 1) in operation group 1 will start at 0 seconds. All 10 operations in operation group 1 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation will start 2 seconds after the previous operation.

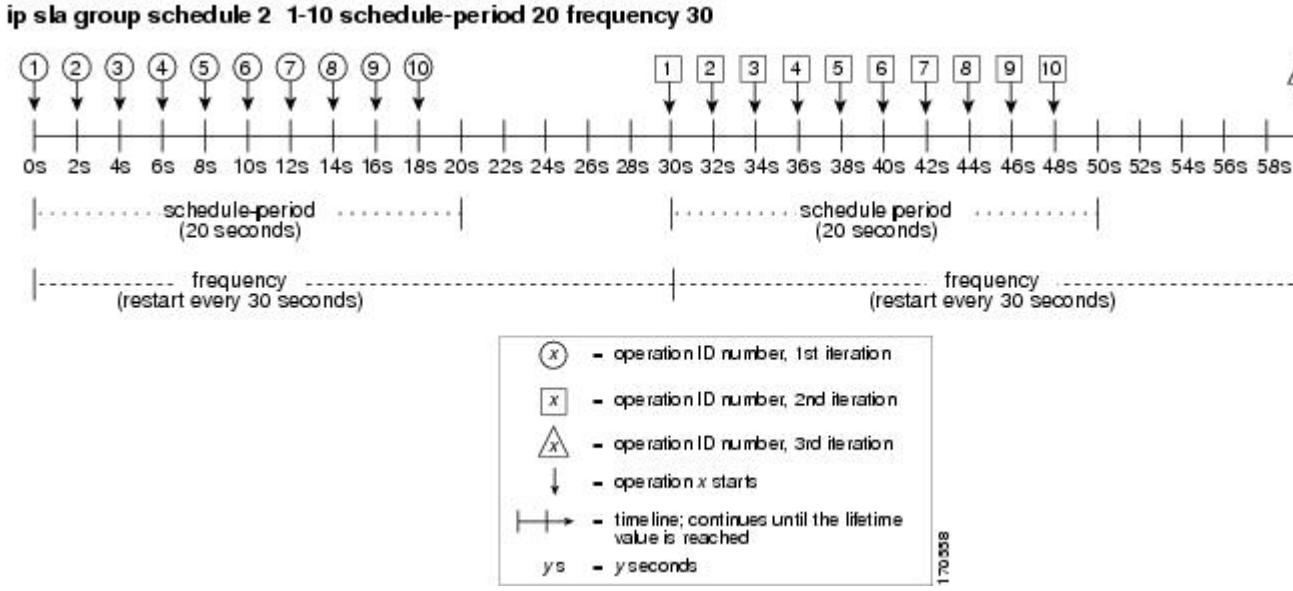
The frequency is the period of time that passes before the operation group is started again (repeated). If the frequency is not specified, the frequency is set to the value of the schedule period. In the example shown above, operation group 1 will start again every 20 seconds. This configuration provides optimal division (spacing) of operations over the specified schedule period.

IP SLAs Multiple Operations Scheduling with Scheduling Period Less Than Frequency

The frequency value is the amount of time that passes before the schedule group is restarted, if the schedule period is less than the frequency, there will be a period of time in which no operations are started.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 2. Operation group 2 has a schedule period of 20 seconds and a frequency of 30 seconds.

Figure 50: Schedule Period Is Less Than Frequency



In this example, the first operation (operation 1) in operation group 2 will start at 0 seconds. All 10 operations in operation group 2 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation will start 2 seconds after the previous operation.

In the first iteration of operation group 2, operation 1 starts at 0 seconds, and the last operation (operation 10) starts at 18 seconds. However, because the group frequency has been configured to 30 seconds each operation in the operation group is restarted every 30 seconds. So, after 18 seconds, there is a gap of 10 seconds as no operations are started in the time from 19 seconds to 29 seconds. Hence, at 30 seconds, the second iteration of operation group 2 starts. As all ten operations in the operation group 2 must start at an evenly distributed interval in the configured schedule period of 20 seconds, the last operation (operation 10) in the operation group 2 will always start 18 seconds after the first operation (operation 1).

As illustrated in the figure above, the following events occur:

- At 0 seconds, the first operation (operation 1) in operation group 2 is started.
- At 18 seconds, the last operation (operation 10) in operation group 2 is started. This means that the first iteration (schedule period) of operation group 1 ends here.
- From 19 to 29 seconds, no operations are started.
- At 30 seconds, the first operation (operation 1) in operation group 2 is started again. The second iteration of operation group 2 starts here.

- At 48 seconds (18 seconds after the second iteration started) the last operation (operation 10) in operation group 2 is started, and the second iteration of operation group 2 ends.
- At 60 seconds, the third iteration of operation group 2 starts.

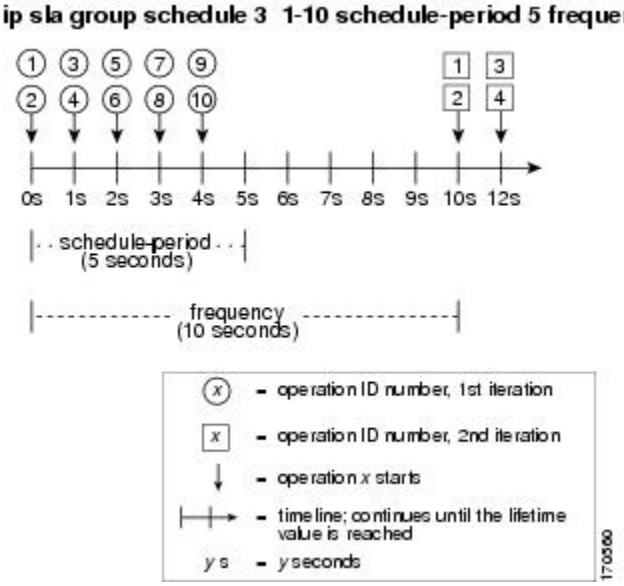
This process continues until the lifetime of operation group 2 ends. The lifetime value is configurable. The default lifetime for an operation group is forever.

Multiple Operations Scheduling When the Number of IP SLAs Operations Are Greater Than the Schedule Period

The minimum time interval between the start of IP SLAs operations in a group operation is 1 second. Therefore, if the number of operations to be multiple scheduled is greater than the schedule period, the IP SLAs multiple operations scheduling functionality will schedule more than one operation to start within the same 1-second interval. If the number of operations getting scheduled does not equally divide into 1-second intervals, then the operations are equally divided at the start of the schedule period with the remaining operations to start at the last 1-second interval.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 3. Operation group 3 has a schedule period of 5 seconds and a frequency of 10 seconds.

Figure 51: Number of IP SLAs Operations Is Greater Than the Schedule Period--Even Distribution



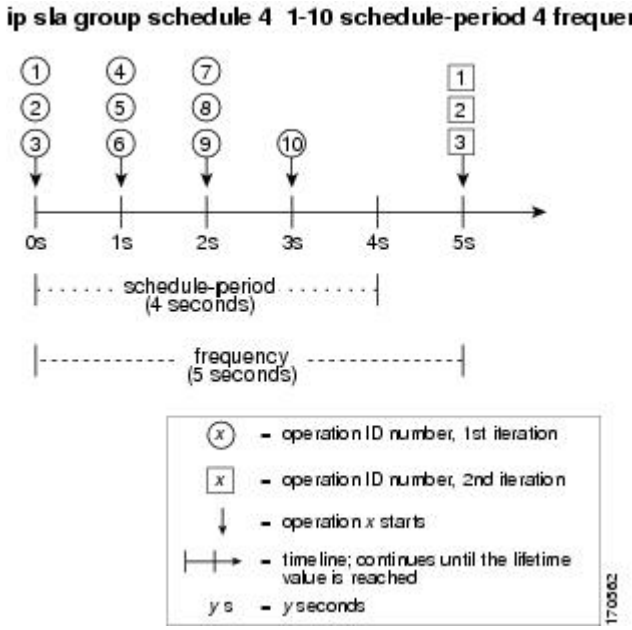
In this example, when dividing the schedule period by the number of operations (5 seconds divided by 10 operations, which equals one operation every 0.5 seconds) the start time of each IP SLAs operation is less than 1 second. Since the minimum time interval between the start of IP SLAs operations in a group operation is 1 second, the IP SLAs multiple operations scheduling functionality instead calculates how many operations it should start in each 1-second interval by dividing the number of operations by the schedule period (10 operations divided by 5 seconds). Therefore, as shown in the figure above, two operations will be started every 1 second.

As the frequency is set to 10 in this example, each iteration of operation group 3 will start 10 seconds after the start of the previous iteration. However, this distribution is not optimal as there is a gap of 5 seconds (frequency minus schedule period) between the cycles.

If the number of operations getting scheduled does not equally divide into 1-second intervals, then the operations are equally divided at the start of the schedule period with the remaining operations to start at the last 1-second interval.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 4. Operation group 4 has a schedule period of 4 seconds and a frequency of 5 seconds.

Figure 52: Number of IP SLAs Operations Is Greater Than the Schedule Period--Uneven Distribution



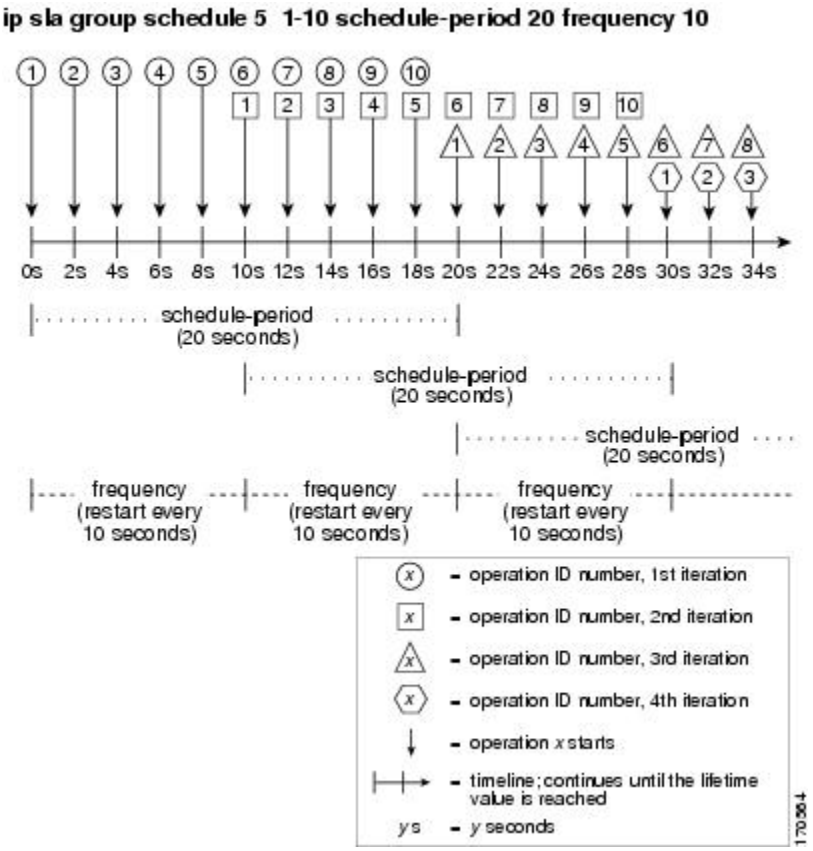
In this example, the IP SLAs multiple operations scheduling functionality calculates how many operations it should start in each 1-second interval by dividing the number of operations by the schedule period (10 operations divided by 4 seconds, which equals 2.5 operations every 1 second). Since the number of operations does not equally divide into 1-second intervals, this number will be rounded off to the next whole number (see the figure above) with the remaining operations to start at the last 1-second interval.

IP SLAs Multiple Operations Scheduling with Scheduling Period Greater Than Frequency

The value of frequency is the amount of time that passes before the schedule group is restarted. If the schedule period is greater than the frequency, there will be a period of time in which the operations in one iteration of an operation group overlap with the operations of the following iteration.

The figure below illustrates the scheduling of operation 1 to operation 10 within operation group 5. Operation group 5 has a schedule period of 20 seconds and a frequency of 10 seconds.

Figure 53: IP SLAs Group Scheduling with Schedule Period Greater Than Frequency



In this example, the first operation (operation 1) in operation group 5 will start at 0 seconds. All 10 operations in operation group 5 (operation 1 to operation 10) must be started in the schedule period of 20 seconds. The start time of each IP SLAs operation is evenly distributed over the schedule period by dividing the schedule period by the number of operations (20 seconds divided by 10 operations). Therefore, each operation will start 2 seconds after the previous operation.

In the first iteration of operation group 5, operation 1 starts at 0 seconds, and operation 10, the last operation in the operation group, starts at 18 seconds. Because the operation group is configured to restart every 10 seconds (**frequency 10**), the second iteration of operation group 5 starts again at 10 seconds, before the first iteration is completed. Therefore, an overlap of operations 6 to 10 of the first iteration occurs with operations 1 to 5 of the second iteration during the time period of 10 to 18 seconds (see the figure above). Similarly, there is an overlap of operations 6 to 10 of the second iteration with operations 1 to 5 of the third iteration during the time period of 20 to 28 seconds.

In this example, the start time of operation 1 and operation 6 need not be at exactly the same time, but will be within the same 2-second interval.

The configuration described in this section is not recommended as you can configure multiple operations to start within the same 1-second interval by configuring the number of operations greater than the schedule period. For information, see the "Multiple Operations Scheduling When the Number of IP SLAs Operations Are Greater Than the Schedule Period" section.

IP SLAs Random Scheduler

The IP SLAs Random Scheduler feature is an enhancement to the existing IP SLAs Multioperation Scheduling feature. The IP SLAs Multioperation Scheduling feature provides the capability to easily schedule multiple IP SLAs operations to begin at intervals equally distributed over a specified duration of time and to restart at a specified frequency. With the IP SLAs Random Scheduler feature, you can now schedule multiple IP SLAs operations to begin at random intervals uniformly distributed over a specified duration of time and to restart at uniformly distributed random frequencies within a specified frequency range. Random scheduling improves the statistical metrics for assessing network performance.

Note The IP SLAs Random Scheduler feature is not in compliance with RFC2330 because it does not account for inter-packet randomness.

The IP SLAs random scheduler option is disabled by default. To enable the random scheduler option, you must set a frequency range when configuring a group schedule in global configuration mode. The group of operations restarts at uniformly distributed random frequencies within the specified frequency range. The following guidelines apply for setting the frequency range:

- The starting value of the frequency range should be greater than the timeout values of all the operations in the group operation.
- The starting value of the frequency range should be greater than the schedule period (amount of time for which the group operation is scheduled). This guideline ensures that the same operation does not get scheduled more than once within the schedule period.

The following guidelines apply if the random scheduler option is enabled:

- The individual operations in a group operation will be uniformly distributed to begin at random intervals over the schedule period.
- The group of operations restarts at uniformly distributed random frequencies within the specified frequency range.
- The minimum time interval between the start of each operation in a group operation is 100 milliseconds (0.1 seconds). If the random scheduler option is disabled, the minimum time interval is 1 second.
- Only one operation can be scheduled to begin at any given time. If the random scheduler option is disabled, multiple operations can begin at the same time.
- The first operation will always begin at 0 milliseconds of the schedule period.
- The order in which each operation in a group operation begins is random.

How to Configure an IP SLAs Multioperation Scheduler

Scheduling Multiple IP SLAs Operations

Note

- All IP SLAs operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group should be the same.
- List of one or more operation ID numbers to be added to a multioperation group is limited to a maximum of 125 characters, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}]
4. **exit**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *schedule-period-range* [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever**	*seconds*}] [**start-time** {*hh:mm*[*:ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh:mm:ss*}] **Example:** `Device(config)# ip sla group schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	Specifies an IP SLAs operation group number and the range of operation numbers to be scheduled in global configuration mode.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to the privileged EXEC mode.
Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays the IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays the IP SLAs configuration details.

Enabling the IP SLAs Random Scheduler

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *seconds* [**ageout** *seconds*] [**frequency** [*seconds*| **range** *random-frequency-range*]] [**life**{**forever** | *seconds*}] [**start-time**{*hh:mm*[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla group schedule** *group-operation-number operation-id-numbers* **schedule-period** *seconds* [**ageout** *seconds*] [**frequency** [*seconds*| **range** *random-frequency-range*]] [**life**{**forever**	*seconds*}] [**start-time**{*hh:mm*[*:ss*] [*month day*	*day month*]	**pending**	**now**	**after** *hh:mm:ss*}]	Specifies the scheduling parameters of a group of IP SLAs operations. • To enable the IP SLAs random scheduler option, you must configure the **frequency range** *random-frequency-range* keywords and argument.

	Command or Action	Purpose
	Example: `Device(config)# ip sla group schedule 2 1-3 schedule-period 50 frequency range 80-100`	
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying IP SLAs Multiple Operations Scheduling

SUMMARY STEPS

1. **show ip sla statistics**
2. **show ip sla group schedule**
3. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip sla statistics** **Example:** `Device# show ip sla statistics`	(Optional) Displays the IP SLAs operation details.
Step 2	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays the IP SLAs group schedule details.
Step 3	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays the IP SLAs configuration details.

Examples

After you have scheduled the multiple IP SLAs operations, you can verify the latest operation details using the appropriate **show** commands.

The following example schedules IP SLAs operations 1 through 20 in the operation group 1 with a schedule period of 60 seconds and a life value of 1200 seconds. By default, the frequency is equivalent to the schedule period. In this example, the start interval is 3 seconds (schedule period divided by number of operations).

```
Device# ip sla group schedule 1 1-20 schedule-period 60 life 1200
```

The following example shows the details of the scheduled multiple IP SLAs operation using the **show ip sla group schedule** command.

```
Device# show ip sla group schedule
Group Entry Number: 1
Probes to be scheduled: 1-20
Total number of probes: 20
Schedule period: 60
Group operation frequency: Equals schedule period
Status of entry (SNMP RowStatus): Active
Next Scheduled Start Time: Start Time already passed
Life (seconds): 1200
Entry Ageout (seconds): never
```

The following example shows the details of the scheduled multiple IP SLAs operation using the **show ip sla configuration** command. The last line in the example indicates that the IP SLAs operations are multiple scheduled (TRUE).

```
Device# show ip sla configuration 1
Entry number: 1
Owner:
Tag:
Type of operation to perform: udpEcho
Target address: 10.2.31.121
Source address: 0.0.0.0
Target port: 9001
Source port: 0
Request size (ARR data portion): 16
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Data pattern:
Vrf Name:
Control Packets: enabled
Operation frequency (seconds): 60
Next Scheduled Start Time: Start Time already passed
Life (seconds): 1200
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
Enhanced History:
Number of history Lives kept: 0
Number of history Buckets kept: 15
History Filter Type: None
Group Scheduled : TRUE
```

The following example shows the latest operation start time of the scheduled multiple IP SLAs operation, when the operations are scheduled at equal intervals, using the **show ip sla statistics** command:

```
Device# show ip sla statistics | include Latest operation start time
Latest operation start time: *03:06:21.760 UTC Tue Oct 21 2003
Latest operation start time: *03:06:24.754 UTC Tue Oct 21 2003
Latest operation start time: *03:06:27.751 UTC Tue Oct 21 2003
```

```
Latest operation start time: *03:06:30.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:33.754 UTC Tue Oct 21 2003
Latest operation start time: *03:06:36.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:39.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:42.753 UTC Tue Oct 21 2003
Latest operation start time: *03:06:45.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:48.752 UTC Tue Oct 21 2003
Latest operation start time: *03:06:51.753 UTC Tue Oct 21 2003
Latest operation start time: *03:06:54.755 UTC Tue Oct 21 2003
Latest operation start time: *03:06:57.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:00.753 UTC Tue Oct 21 2003
Latest operation start time: *03:07:03.754 UTC Tue Oct 21 2003
Latest operation start time: *03:07:06.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:09.752 UTC Tue Oct 21 2003
Latest operation start time: *03:07:12.753 UTC Tue Oct 21 2003
Latest operation start time: *03:07:15.755 UTC Tue Oct 21 2003
Latest operation start time: *03:07:18.752 UTC Tue Oct 21 2003
```

Configuration Examples for an IP SLAs Multioperation Scheduler

Example Scheduling Multiple IP SLAs Operations

The following example shows how to scheduls IP SLAs operations 1 to 10 in the operation group 1 with a schedule period of 20 seconds. By default, the frequency is equivalent to the schedule period.

```
Device# ip sla group schedule 1 1-10 schedule-period 20
```

The following example shows the details of the scheduled multiple IP SLAs operation using the **show ip sla group schedule** command. The last line in the example indicates that the IP SLAs operations are multiple scheduled (TRUE).

```
Device# show ip sla group schedule
Multi-Scheduling Configuration:
Group Entry Number: 1
Probes to be scheduled: 1-10
Schedule period :20
Group operation frequency: 20
Multi-scheduled: TRUE
```

Example Enabling the IP SLAs Random Scheduler

The following example shows how to schedule IP SLAs operations 1 to 3 as a group (identified as group 2). In this example, the operations are scheduled to begin at uniformly distributed random intervals over a schedule period of 50 seconds. The first operation is scheduled to start immediately. The interval is chosen from the specified range upon every invocation of the probe. The random scheduler option is enabled and the uniformly distributed random frequencies at which the group of operations will restart is chosen within the range of 80-100 seconds.

```
ip sla group schedule 2 1-3 schedule-period 50 frequency range 80-100 start-time now
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference, All Releases
Cisco IOS IP SLAs: general information	"Cisco IOS IP SLAs Overview" module of the *Cisco IOS IP SLAs Configuration Guide*.
Multioperation scheduling for IP SLAs	"Configuring Multioperation Scheduling of IP SLAs Operations" module of the *Cisco IOS P SLAs Configuration Guide*
Proactive threshold monitoring for IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

MIBs

MIBs	MIBs Link
CISCO-RTTMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

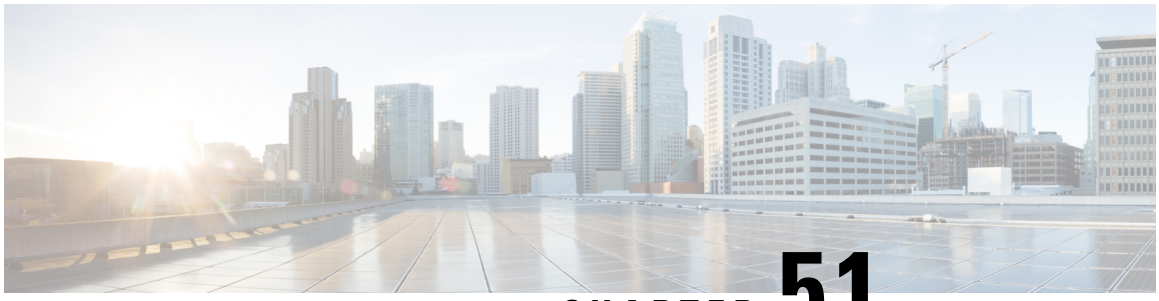
Feature Information for a IP SLAs Multioperation Scheduler

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 74: Feature Information for IP SLAs Multioperation Scheduling

Feature Name	Releases	Feature Information
IP SLAs Multioperation Scheduler		The IP SLAs Multioperation Scheduler feature provides a highly scalable infrastructure for IP SLAs by allowing you to schedule multiple IP SLAs operations using a single command.
IP SLAs Random Scheduler		The IP SLAs Random Scheduler feature provides the capability to schedule multiple IP SLAs operations to begin at random intervals uniformly distributed over a specified duration of time and to restart at uniformly distributed random frequencies within a specified frequency range.



CHAPTER 51

Configuring Proactive Threshold Monitoring for IP SLAs Operations

This document describes the proactive monitoring capabilities of IP Service Level Agreements (SLAs) using thresholds and reaction triggering.

Information About Proactive Threshold Monitoring

IP SLAs Reaction Configuration

IP SLAs reactions are configured to trigger when a monitored value exceeds or falls below a specified level or when a monitored event, such as a timeout or connection loss, occurs. If IP SLAs measures too high or too low of any configured reaction, IP SLAs can generate a notification to a network management application or trigger another IP SLA operation to gather more data.

When an IP SLA operation is triggered, the (triggered) target operation starts and continues to run independently and without knowledge of the condition of the triggering operation. The target operation continues to run until its life expires, as specified by the target operation's configured lifetime value. The target operation must finish its life before it can be triggered again.

In Cisco IOS Release 15.2(3) and later releases, the (triggered) target operation runs until the condition-cleared event. After which the target operation gracefully stops and the state of the target operation changes from Active to Pending so it can be triggered again.

Supported Reactions by IP SLAs Operation

The tables below list which reactions are supported for each IP SLA operation.

Table 75: Supported Reaction Configuration, by IP SLA Operation

Reaction	ICMP Echo	Path Echo	UDP Jitter	UDP Echo	TCP Connect	DHCP	DLSW	ICMP Jitter	DNS	Frame Relay
Failure	Y	--	Y	Y	Y	Y	--	Y	Y	--
RTT	Y	Y	--	Y	Y	Y	Y	--	Y	Y
RTTAvg	--	--	Y	--	--	--	--	Y	--	--
timeout	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
connectionLoss	--	--	Y	Y	Y	--	--	--	--	
verifyError	--	--	Y	Y	--	--	--	Y	--	Y
jitterSDAvg	--	--	Y	--	--	--		Y	--	--
jitterAvg	--	--	Y	--	--	--	--	Y	--	--
packetLateArrival	--	--	Y	--	--	--	--	Y	--	--
packetOutOfSequence	--	--	Y	--	--	--	--	Y	--	--
MaxOfPostiveSD	--	--	Y	--	--	--		Y	--	--
MaxOfNegativeSD	--	--	Y	--	--	--	--	Y	--	--
MaxOfPostiveDS	--	--	Y	--	--	--	--	Y	--	--
MaxOfNegativeDS	--	--	Y	--	--	--	--	Y	--	--
MOS	--	--	Y	--	--	--		--	--	--
ICPIF	--	--	Y	--	--	--	--	--	--	--
PacketLossDS	--	--	Y	--	--	--	--	--	--	--
PacketLossSD	--	--	Y	--	--	--	--	--	--	--
PacketMIA	--	--	Y	--	--	--		--	--	--
iaJitterDS	--	--	--	--	--	--	--	--	--	--
frameLossDS	--	--	--	--	--	--	--	--	--	--
mosLQDSS	--	--	--	--	--	--	--	--	--	--
mosCQDS	--	--	--	--	--	--	--	--	--	--
rfactorDS	--	--	--	--	--	--	--	--	--	--
iaJitterSD	--	--	--	--	--	--	--	--	--	--
successivePacketLoss	--	--	--	--	--	--	--	Y	--	--
MaxOfLatencyDS	--	--	--	--	--	--	--	Y	--	--

Reaction	ICMP Echo	Path Echo	UDP Jitter	UDP Echo	TCP Connect	DHCP	DLSW	ICMP Jitter	DNS	Frame Relay
MaxOfLatencySD	--	--	--	--	--	--	--	Y	--	--
LatencyDS	--	--	--	--	--	--	--	Y	--	--
LatencySD	--	--	--	--	--	--	--	Y	--	--
packetLoss	--	--	--	--	--	--	--	Y	--	--

Table 76: Supported Reaction Configuration, by IP SLA Operation

Reaction	HTTP	SLM	RTP	FTP	Lsp Trace	Post delay	Path Jitter	LSP Ping	Gatekeeper Registration
Failure	--	--	--	--	--	--	--	--	--
RTT	Y	Y	Y	Y	Y	Y	Y	Y	Y
RTTAvg	--	--	--	--	--	--	--	--	--
timeout	Y	Y	Y	Y	--	Y	Y	Y	Y
connectionLoss	Y		Y	Y	Y	--	--	Y	--
verifyError	--	--	--	--	--	--	--	--	--
jitterSDAvg	--	--	--	--	--	--	Y	--	--
jitterAvg	--	--	--	--	--	--	Y	--	--
packetLateArrival	--	--	--	--	--	--	Y	--	--
packetOutOfSequence	--	--	--	--	--	--	Y	--	--
MaxOfPostiveSD	--	--	--	--	--	--	Y	--	--
MaxOfNegativeSD	--	--	--	--	--	--	Y	--	--
MaxOfPostiveDS	--	--	--	--	--	--	Y	--	--
MaxOfNegativeDS	--	--	--	--	--	--	Y	--	--
MOS	--	--	--	--	--	--	--	--	--
ICPIF	--	--	--	--	--	--	--	--	--
PacketLossDS	--	--	Y	--	--	--	--	--	--
PacketLossSD	--	--	Y	--	--	--	--	--	--
PacketMIA	--	--	Y	--	--	--	--	--	--
iaJitterDS	--	--	Y	--	--	--	--	--	--
frameLossDS	--	--	Y	--	--	--	--	--	--

Reaction	HTTP	SLM	RTP	FTP	Lsp Trace	Post delay	Path Jitter	LSP Ping	Gatekeeper Registration
mosLQDSS	--	--	Y	--	--	--	--	--	--
mosCQDS	--	--	Y	--	--	--	--	--	--
rfactorDS	--	--	Y						
iaJitterSD	--	--	Y	--	--	--	--	--	--
successivePacketLoss	--	--	--	--	--	--	--	--	--
MaxOfLatencyDS	--	--	--	--	--	--	--	--	--
MaxOfLatencySD	--	--	--	--	--	--	--	--	--
LatencyDS	--	--	--	--	--	--	--	--	--
LatencySD	--	--	--	--	--	--	--	--	--
packetLoss	--	--	--	--	--	--	--	--	--

IP SLAs Threshold Monitoring and Notifications

IP SLAs supports proactive threshold monitoring and notifications for performance parameters such as average jitter, unidirectional latency, bidirectional round-trip time (RTT), and connectivity for most IP SLAs operations. The proactive monitoring capability also provides options for configuring reaction thresholds for important VoIP related parameters including unidirectional jitter, unidirectional packet loss, and unidirectional VoIP voice quality scoring.

Notifications for IP SLAs are configured as a triggered reaction. Packet loss, jitter, and Mean Operation Score (MOS) statistics are specific to IP SLAs jitter operations. Notifications can be generated for violations in either direction (source-to-destination and destination-to-source) or for out-of-range RTT values for packet loss and jitter. Events, such as traps, are triggered when the RTT value rises above or falls below a specified threshold.

IP SLAs can generate system logging (syslog) messages when a reaction condition occurs. System logging messages can be sent as Simple Network Management Protocol (SNMP) traps (notifications) using the CISCO-RTTMON-MIB. SNMP traps for IP SLAs are supported by the CISCO-RTTMON-MIB and CISCO-SYSLOG-MIB.

Severity levels in the CISCO-SYSLOG-MIB are defined as follows: SyslogSeverity INTEGER {emergency(1), alert(2), critical(3), error(4), warning(5), notice(6), info(7), debug(8)}

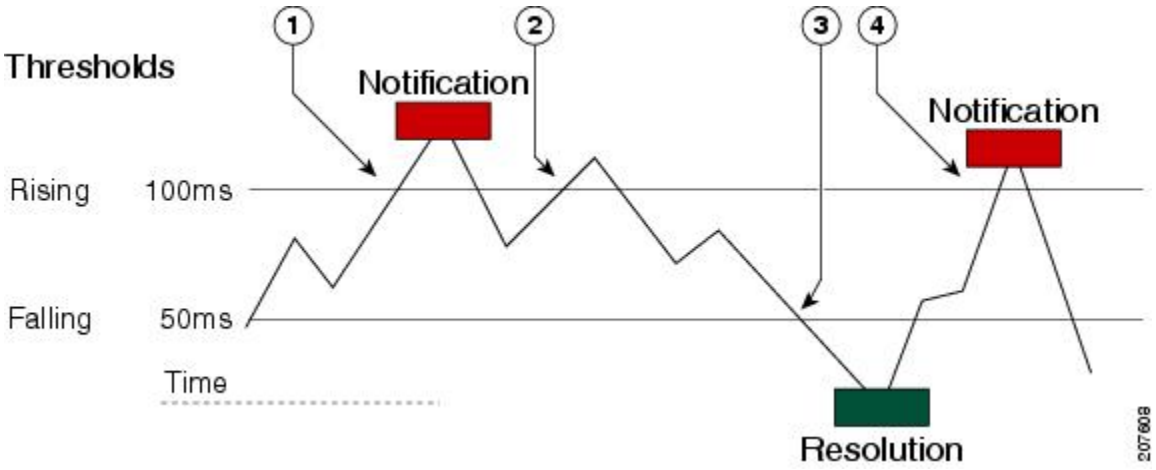
The values for severity levels are defined differently for the system logging process in software. Severity levels for the system logging process in Cisco software are defined as follows: {emergency (0), alert (1), critical (2), error (3), warning (4), notice (5), informational (6), debugging (7)}.

IP SLAs Threshold violations are logged as level 6 (informational) within the Cisco system logging process but are sent as level 7 (info) traps from the CISCO-SYSLOG-MIB.

Notifications are not issued for every occurrence of a threshold violation. The figure below illustrates the sequence for a triggered reaction that occurs when the monitored element exceeds the upper threshold. An event is sent and a notification is issued when the rising threshold is exceeded for the first time. Subsequent

threshold-exceeded notifications are issued only after the monitored value falls below the falling threshold before exceeding the rising threshold ag ain .

Figure 54: IP SLAs Triggered Reaction Condition and Notifications for Threshold Exceeded



1	An event is sent and a threshold-exceeded notification is issued when the rising threshold is exceeded for the first time.
2	Consecutive over-rising threshold violations occur without issuing additional notifications.
3	The monitored value goes below the falling threshold.
4	Another threshold-exceeded notification is issued when the rising threshold is exceeded only after the monitored value first fell below the falling threshold.

Note A lower-threshold notification is also issued the first time that the monitored element falls below the falling threshold (3). As described, subsequent notifications for lower-threshold violations will be issued only after the rising threshold is exceeded before the monitored value falls below the falling threshold again.

RTT Reactions for Jitter Operations

RTT reactions for jitter operations are triggered only at the end of the operation and use the latest value for the return-trip time (LatestRTT), which matches the value of the average return-trip time (RTTAvg).

SNMP traps for RTT for jitter operations are based on the value of the average return-trip time (RTTAvg) for the whole operation and do not include RTT values for each individual packet sent during the operation. For example, if the average is below the threshold, up to half of the packets can actually be above threshold but this detail is not included in the notification because the value is for the whole operation only.

Only syslog messages are supported for RTTAvg threshold violations. Syslog nmessages are sent from the CISCO-RTTMON-MIB.

How to Configure Proactive Threshold Monitoring

Configuring Proactive Threshold Monitoring

Perform this task to configure thresholds and reactive triggering for generating traps or starting another operation.

Before you begin

- IP SLAs operations to be started when violation conditions are met must be configured.

Note

- RTT reactions for jitter operations are triggered only at the end of the operation and use the latest value for the return-trip time (LatestRTT).
- SNMP traps for RTT for jitter operations are based on the average value for the return-trip time (RTTAvg) for the whole operation only and do not include return-trip time values for individual packets sent during the operation. Only syslog messages are supported for RTTAvg threshold violations.
- Only syslog messages are supported for RTT violations during Jitter operations.
- Only SNMP traps are supported for RTT violations during non-Jitter operations.
- Only syslog messages are supported for non-RTT violations other than timeout, connectionLoss, or verifyError.
- Both SNMP traps and syslog messages are supported for timeout, connectionLoss, or verifyError violations only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla reaction-configuration** *operation-number* **react** *monitored-element* [**action-type** *option*] [**threshold-type** {**average** [*number-of-measurements*] | **consecutive** [*occurrences*] | **immediate** | **never** | **xofy** [*x-value y-value*]}] [**threshold-value** *upper-threshold lower-threshold*]
4. **ip sla reaction-trigger** *operation-number target-operation*
5. **ip sla logging traps**
6. Do one of the following:
   - **snmp-server enable traps rtr**
   - **snmp-server enable traps syslog**
7. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
8. **exit**
9. **show ip sla reaction- configuration** [*operation-number*]
10. **show ip sla reaction- trigger** [*operation-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla reaction-configuration** *operation-number* **react** *monitored-element* [**action-type** *option*] [**threshold-type** {**average** [*number-of-measurements*] | **consecutive** [*occurrences*] | **immediate** | **never** | **xofy** [*x-value y-value*]}] [**threshold-value** *upper-threshold lower-threshold*] **Example:** `Device(config)# ip sla reaction-configuration 10 react jitterAvg threshold-type immediate threshold-value 5000 3000 action-type trapAndTrigger`	Configures the action (SNMP trap or IP SLAs trigger) that is to occur based on violations of specified thresholds.
Step 4	**ip sla reaction-trigger** *operation-number target-operation* **Example:** `Device(config)# ip sla reaction-trigger 10 2`	(Optional) Starts another IP SLAs operation when the violation conditions are met. • Required only if the **ip sla reaction-configuration** command is configured with either the **trapAndTrigger**or **triggerOnly**keyword.
Step 5	**ip sla logging traps** **Example:** `Device(config)# ip sla logging traps`	(Optional) Enables IP SLAs syslog messages from CISCO-RTTMON-MIB.

Step 6	Do one of the following:
	• **snmp-server enable traps rtr**
	• **snmp-server enable traps syslog**
	Example:
	`Device(config)# snmp-server enable traps rtr`
	Example:
	`Device(config)# snmp-server enable traps syslog`
		• (Optional) The first example shows how to enable the system to generate CISCO-RTTMON-MIB traps.
		• (Optional) The second example shows how to enable the system to generate CISCO-SYSLOG-MIB traps.

Step 7	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** |	(Optional) Sends traps to a remote host.

	Command or Action	Purpose
	priv]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Device(config)# snmp-server host 10.1.1.1 public syslog`	• Required if the **snmp-server enable traps** command is configured.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 9	**show ip sla reaction- configuration** [*operation-number*] **Example:** `Device# show ip sla reaction-configuration 10`	(Optional) Displays the configuration of proactive threshold monitoring.
Step 10	**show ip sla reaction- trigger** [*operation-number*] **Example:** `Device# show ip sla reaction-trigger 2`	(Optional) Displays the configuration status and operational state of target operations to be triggered.

Configuration Examples for Proactive Threshold Monitoring

Example Configuring an IP SLAs Reaction Configuration

In the following example, IP SLAs operation 10 is configured to send an SNMP logging trap when the MOS value either exceeds 4.9 (best quality) or falls below 2.5 (poor quality):

```
Device(config)# ip sla reaction-configuration 10 react mos threshold-type immediate
threshold-value 490 250 action-type trapOnly
```

The following example shows the default configuration for the **ip sla reaction-configuration** command:

```
Device# show ip sla reaction-configuration 1
Entry number: 1
Reaction Configuration not configured
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip sla reaction-configuration 1
Device(config)# do show ip sla reaction-configuration 1
Entry number: 1
Reaction: rtt
Threshold Type: Never
Rising (milliseconds): 5000
Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Action Type: None
```

Example Verifying an IP SLAs Reaction Configuration

The following example shows that multiple monitored elements are configured for the IP SLAs operation (1), as indicated by the values of Reaction: in the output:

```
Device# show ip sla reaction-configuration

Entry Number: 1
Reaction: RTT
Threshold type: Never
Rising (milliseconds): 5000
Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Action Type: None
Reaction: jitterDSAvg
Threshold type: average
Rising (milliseconds): 5
Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: triggerOnly
Reaction: jitterDSAvg
Threshold type:  immediate
Rising (milliseconds): 5
Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: trapOnly
Reaction: PacketLossSD
Threshold type:  immediate
Rising (milliseconds): 5
Threshold Falling (milliseconds): 3
Threshold Count: 5
Threshold Count2: 5
Action Type: trapOnly
```

Example Triggering SNMP Notifications

The following example shows how to configure proactive threshold monitoring so that CISCO-SYSLOG-MIB traps are sent to the remote host at 10.1.1.1 if the threshold values for RTT or VoIP MOS are violated:

```
! Configure the operation on source.
Device(config)# ip sla 1

Device(config-ip-sla)# udp-jitter 10.1.1.1 3000 codec g711alaw
Device(config-ip-sla-jitter)# exit

Device(config)# ip sla schedule 1 start now life forever

! Configure thresholds and reactions.
Device(config)# ip sla reaction-configuration 1 react rtt threshold-type immediate
threshold-value 3000 2000 action-type trapOnly

Device(config)# ip sla reaction-configuration 1 react MOS threshold-type consecutive 4
threshold-value 390 220 action-type trapOnly

Device(config)# ip sla logging traps
```

```
! The following command sends traps to the specified remote host.
Device(config)# snmp-server host 10.1.1.1 version 2c public syslog

! The following command is needed for the system to generate CISCO-SYSLOG-MIB traps.
Device(config)# snmp-server enable traps syslog
```

The following sample system logging messages shows that IP SLAs threshold violation notifications are generated as level 6 (informational) in the Cisco system logging process:

```
3d18h:%RTT-6-SAATHRESHOLD:RTR(11):Threshold exceeded for MOS
```

This following sample SNMP notification from the CISCO-SYSLOG-MIB for the same violation is a level 7 (info) notification:

```
3d18h:SNMP:V2 Trap, reqid 2, errstat 0, erridx 0
 sysUpTime.0 = 32613038
 snmpTrapOID.0 = ciscoSyslogMIB.2.0.1
 clogHistoryEntry.2.71 = RTT
 clogHistoryEntry.3.71 = 7
 clogHistoryEntry.4.71 = SAATHRESHOLD
 clogHistoryEntry.5.71 = RTR(11):Threshold exceeded for MOS
 clogHistoryEntry.6.71 = 32613037
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

MIBs

MIBs	MIBs Link
• CISCO-RTTMON-MIB • CISCO-SYSLOG-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

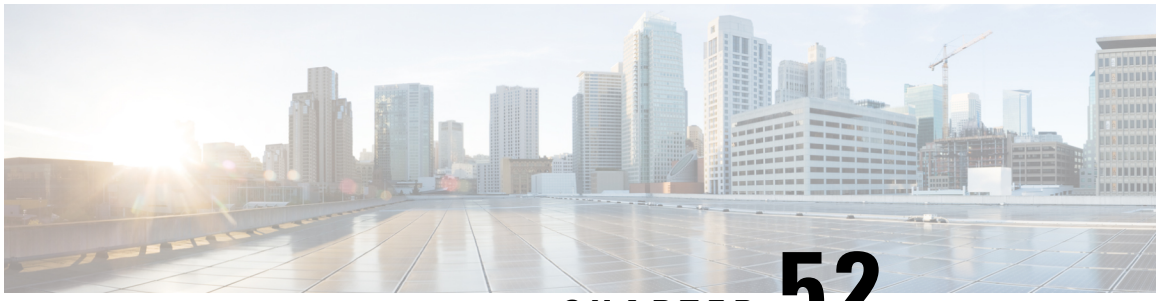
Feature Information for IP SLAs Proactive Threshold Monitoring

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 77: Feature Information for IP SLAs Proactive Threshold Monitoring

Feature Name	Releases	Feature Information
IP SLAs - Reaction Threshold		Cisco IOS IP SLAs proactive threshold monitoring capability allows you to configure an IP SLAs operation to react to certain measured network conditions.
IP SLAs - VoIP Traps		The IP SLA - VoIP Traps feature includes new capabilities for configuring reaction thresholds for important VoIP related parameters such as unidirectional jitter, unidirectional packet loss, and unidirectional VoIP voice quality scoring (MOS scores).
IP SLAs Additional Threshold Traps		This enhancement for IP SLAs reaction threshold monitoring includes per direction average jitter, per direction packet loss, maximum positive and negative jitter, and Mean Opinion Score (MOS) traps. The feature also enables one-way latency jitter, packet loss and latency traps within IP SLAs and includes traps for packet loss due to missing in action and late arrivals.



CHAPTER 52

IP SLAs TWAMP Responder

The Two-Way Active Measurement Protocol (TWAMP) defines a flexible method for measuring round-trip IP performance between any two devices.

TWAMP enables complete IP performance measurement. TWAMP also provides a flexible choice of solutions because it supports all devices deployed in the network.

This chapter describes how to configure the Two-Way Active Measurement Protocol (TWAMP) responder on a Cisco device to measure IP performance between the Cisco device and a non-Cisco TWAMP control device on your network.

Note IPv6 is supported for IP SLA TWAMP Responder on the RSP3 module.

Prerequisites for IP SLAs TWAMP Responder

For the IP SLAs TWAMP responder to function, a TWAMP control-client and the session-sender must be configured in your network.

Restrictions for IP SLAs TWAMP Responder

- For IP SLAs TWAMP Responder v1.0, the TWAMP server and the session-reflector must be configured on the same Cisco device.
- Time stamping is not supported for TWAMP test packets that ingress/egress via management interface.
- Time stamping is not supported on interfaces that are not routed or BDI interfaces.

- Time stamping is not supported on MPLS/VPLS interfaces.
- TWAMP client and session sender is not supported.
- Upto nine session-senders can be configured for one TWAMP responder.
- TWAMP Light mode is not supported.

IP SLAs TWAMP Architecture

Two-Way Active Measurement Protocol (TWAMP)

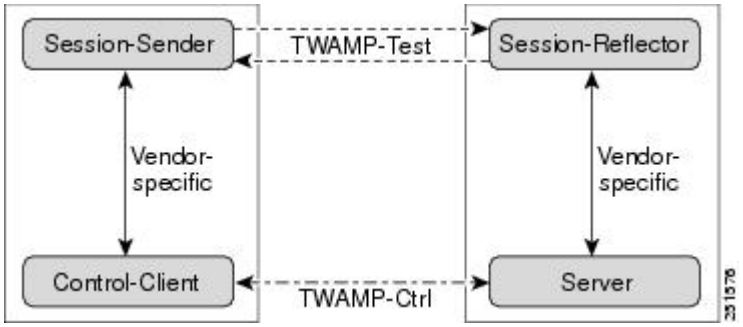
The IETF Two-Way Active Measurement Protocol (TWAMP) defines a standard for measuring round-trip network performance between any two devices that support the TWAMP protocols. The TWAMP-Control protocol is used to set up performance measurement sessions. The TWAMP-Test protocol is used to send and receive performance measurement probes.

The TWAMP architecture is composed of the following four logical entities that are responsible for starting a monitoring session and exchanging packets:

- The control client: It sets up, starts, and stops TWAMP test sessions.
- The session sender: It instantiates TWAMP test packets that are sent to the session reflector.
- The session reflector: It reflects a measurement packet upon receiving a TWAMP test packet. The session reflector does not collect packet statistics in TWAMP.
- The TWAMP server: It is an end system that manages one or more TWAMP sessions and is also capable of configuring each session ports in the end points. The server listens on the TCP port. The session-reflector and server make up the TWAMP responder in an IP SLAs operation.

Although TWAMP defines the different entities for flexibility, it also allows for logical merging of the roles on a single device for ease of implementation. The figure below shows the interactions of four entities of the TWAMP architecture.

Figure 55: TWAMP Architecture

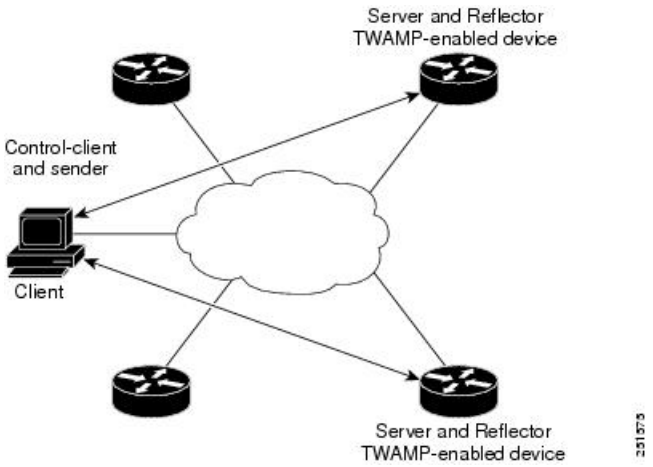


IP SLAs TWAMP Responder

A TWAMP responder interoperates with the control client and session sender on another device that supports TWAMP. In the current implementation, the session reflector and TWAMP server that make up the responder must be co-located on the same device.

In the figure below, one device is the control client and session sender (TWAMP control device), and the other two devices are Cisco devices that are configured as IP SLAs TWAMP responders. Each IP SLAs TWAMP responder is both a TWAMP server and a session-reflector.

Figure 56: IP SLAs TWAMP Responders in a Basic TWAMP Deployment



Configure an IP SLAs TWAMP Responder

Note Effective Cisco IOS-XE Everest 16.6.1, time stamping for sender (T1, T4) and receiver (T3, T2) is performed by the hardware, instead of the software. This time stamping is done bythe hardware to improve the accuracy of jitter and latency measurements.

Note Software time stamping is implemented for TWAMP IP SLA packets on the RSP3 module.

Configuring the TWAMP Server

Note In the current implementation of IP SLAs TWAMP Responder, the TWAMP server and the session reflector must be configured on the same device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla server twamp**
4. **port** *port-number*
5. **timer inactivity** *seconds*
6. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla server twamp**

Example:

```
Device(config)#  ip sla server twamp
```

Configures the device as a TWAMP server and enters TWAMP server configuration mode.

Step 4 **port** *port-number*

Example:

```
Device(config-twamp-srvr)# port 9000
```

(Optional) Configures the port to be used by the TWAMP server to listen for connection and control requests.

Step 5 **timer inactivity** *seconds*

Example:

```
Device(config-twamp-srvr)# timer inactivity 300
```

(Optional) Configures the inactivity timer for a TWAMP control session.

Step 6 **end**

Example:

```
Device(config-twamp-srvr)# end
```

Returns to privileged EXEC mode.

Configuring the Session Reflector

Note In the current implementation of IP SLAs TWAMP Responder, the TWAMP server and the session reflector must be configured on the same device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla responder twamp**
4. **timeout** *seconds*
5. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ip sla responder twamp**

Example:

```
Device(config)# ip sla responder twamp
```

Configures the device as a TWAMP responder and enters TWAMP reflector configuration mode.

Step 4 **timeout** *seconds*

Example:

```
Device(config-twamp-ref)#  timeout 300
```

(Optional) Configures an inactivity timer for a TWAMP test session.

Step 5 **end**

Example:

```
Device(config-twamp-ref)# end
```

Exits to privileged EXEC mode.

Configuration Examples for IP SLAs TWAMP Responder

IP SLAs TWAMP Responder v1.0 Example

The following example and partial output shows how to configure the TWAMP server and the session-reflector for IP SLAs TWAMP Responder v1.0 on the same Cisco device. In this configuration, port 862 is the (default) port to be used by the TWAMP server to listen for connection and control requests. The default port for the server listener is the RFC-specified port and can be reconfigured, if required.

Note In order for the IP SLAs TWAMP responder to function, a control-client and the session-sender must be configured in your network.

```
Device> enable
Device# configure terminal
Device(config)#  ip sla server twamp
Device(config-twamp-srvr)#  exit
Device(config)#  ip sla responder twamp
Device(config-twamp-ref)# end
Device> show running-config
.
.
.
ip sla responder
ip sla responder twamp
ip sla server twamp
port 862
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IP SLAs commands	Cisco IOS IP SLAs Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5357	*Two-Way Active Measurement Protocol (TWAMP)*
RFC 4656	*One-way Active Measurement Protocol (OWAMP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP SLAs TWAMP Responder

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

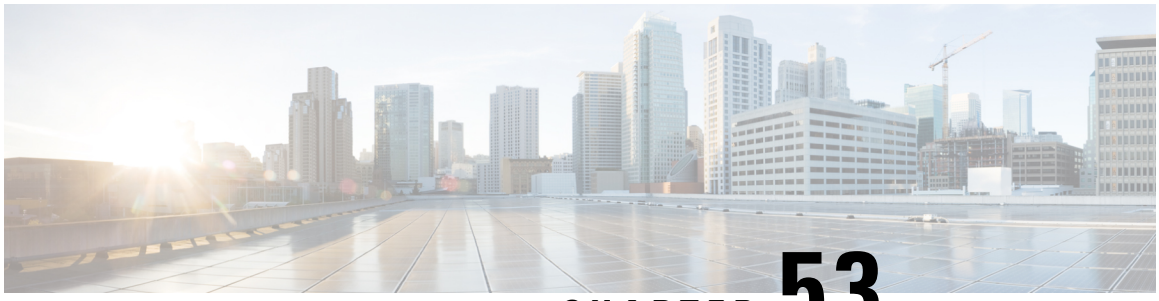
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 78: Feature Information for IP SLAs TWAMP Responder

Feature Name	Releases	Feature Information
IP SLAs TWAMP Responder v1.0		This feature enables you to configure the TWAMP server and the session-reflector on a Cisco device for measuring the round-trip performance between an IP SLAs TWAMP responder and a non-Cisco TWAMP control device in your network. The following commands were introduced or modified: **ip sla responder twamp**, **ip sla server twamp**, **port (twamp)**, **show ip sla standards**, **show ip sla twamp connection**, **show ip sla twamp session**, **show ip sla twamp standards**, **timer inactivity**, **timeout (twamp)**.. In Cisco IOS XE Release 12.2SE, support was added for Cisco ASR 1000 series, Cisco ISR 4000 series, and Cisco CSR 1000v.

PART V

ARP



CHAPTER 53

Address Resolution Protocol

The Address Resolution Protocol (ARP) feature performs a required function in IP routing. ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. ARP maintains a cache (table) in which MAC addresses are mapped to IP addresses. ARP is part of all Cisco systems that run IP.

This feature module explains ARP for IP routing and the optional ARP features you can configure, such as static ARP entries, timeout for dynamic ARP entries, clearing the cache, and proxy ARP.

Information About the Address Resolution Protocol

Layer 2 and Layer 3 Addressing

IP addressing occurs at Layer 2 (data link) and Layer 3 (network) of the Open System Interconnection (OSI) reference model. OSI is an architectural network model developed by ISO and ITU-T that consists of seven layers, each of which specifies particular network functions such as addressing, flow control, error control, encapsulation, and reliable message transfer.

Layer 2 addresses are used for local transmissions between devices that are directly connected. Layer 3 addresses are used for indirectly connected devices in an internetwork environment. Each network uses addressing to identify and group devices so that transmissions can be sent and received. Ethernet (802.2, 802.3, Ethernet II, and Subnetwork Access Protocol [SNAP]), Token Ring, and Fiber Distributed Data Interface (FDDI) use media access control (MAC) addresses that are “burned in” to the network interface card (NIC). The most commonly used network types are Ethernet II and SNAP.

Note For the supported interface types, see the data sheet for your hardware platform.

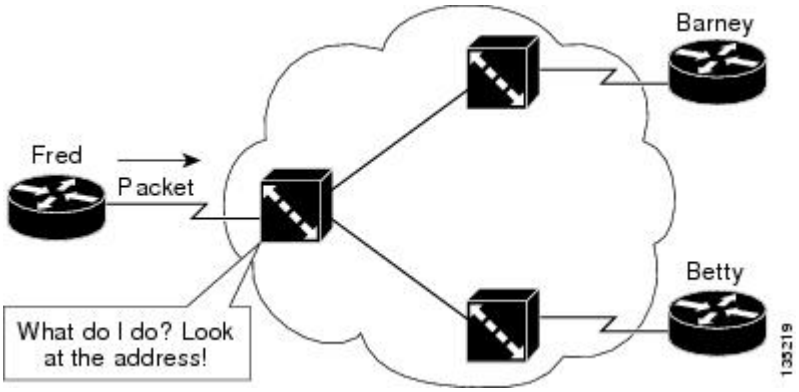
In order for devices to be able to communicate with each when they are not part of the same network, the 48-bit MAC address must be mapped to an IP address. Some of the Layer 3 protocols used to perform the mapping are:

- Address Resolution Protocol (ARP)
- Reverse ARP (RARP)
- Serial Line ARP (SLARP)
- Inverse ARP

For the purposes of IP mapping, Ethernet, Token Ring, and FDDI frames contain the destination and source addresses. Frame Relay and Asynchronous Transfer Mode (ATM) networks, which are packet-switched, data packets take different routes to reach the same destination. At the receiving end, the packet is reassembled in the correct order.

In a Frame Relay network, there is one physical link that has many logical circuits called virtual circuits (VCs). The address field in the frame contains a data-link connection identifier (DLCI), which identifies each VC. For example, in the figure below, the Frame Relay switch to which device Fred is connected receives frames; the switch forwards the frames to either Barney or Betty based on the DLCI that identifies each VC. So Fred has one physical connection but multiple logical connections.

Figure 57: Frame Relay Network



ATM networks use point-to-point serial links with the High-Level Data Link Control (HDLC) protocol. HDLC includes a meaningless address field included in five bytes of the frame header frame with the recipient implied since there can be only one.

Overview of the Address Resolution Protocol

The Address Resolution Protocol (ARP) was developed to enable communications on an internetwork and is defined by RFC 826. Layer 3 devices need ARP to map IP network addresses to MAC hardware addresses so that IP packets can be sent across networks. Before a device sends a datagram to another device, it looks in its ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network. Each device compares the IP address to its own. Only the device with the matching IP address replies to the sending device with a packet containing the MAC address for the device (except in the case of "proxy ARP"). The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. The figure below illustrates the ARP broadcast and response process.

Figure 58: ARP Process

When the destination device lies on a remote network, one beyond another Layer 3 device, the process is the same except that the sending device sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The Layer 3 device on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet.

Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet use Subnetwork Access Protocol (SNAP).

The ARP request message has the following fields:

- HLN—Hardware address length. Specifies how long the hardware addresses are in the message. For IEEE 802 MAC addresses (Ethernet) the value is 6.
- PLN—Protocol address length. Specifies how long the protocol (Layer 3) addresses are in the message. For IPv4, the value is 4.
- OP—Opcode. Specifies the nature of the message by code:
  - 1—ARP request.
  - 2—ARP reply.
  - 3 through 9—RARP and Inverse ARP requests and replies.
- SHA—Sender hardware address. Specifies the Layer 2 hardware address of the device sending the message.
- SPA—Sender protocol address. Specifies the IP address of the sending device.
- THA—Target hardware address. Specifies the Layer 2 hardware address of the receiving device.
- TPA—Target protocol address. Specifies the IP address of the receiving device.

ARP Caching

Because the mapping of IP addresses to media access control (MAC) addresses occurs at each hop (Layer 3 device) on the network for every datagram sent over an internetwork, performance of the network could be compromised. To minimize broadcasts and limit wasteful use of network resources, Address Resolution Protocol (ARP) caching was implemented.

ARP caching is the method of storing network addresses and the associated data-link addresses in memory for a period of time as the addresses are learned. This minimizes the use of valuable network resources to broadcast for the same address each time a datagram is sent. The cache entries must be maintained because the information could become outdated, so it is critical that the cache entries are set to expire periodically. Every device on a network updates its tables as addresses are broadcast.

There are static ARP cache entries and dynamic ARP cache entries. Static entries are manually configured and kept in the cache table on a permanent basis. Static entries are best for devices that have to communicate with other devices usually in the same network on a regular basis. Dynamic entries are added by Cisco software, kept for a period of time, and then removed.

Static and Dynamic Entries in the ARP Cache

Static routing requires an administrator to manually enter IP addresses, subnet masks, gateways, and corresponding media access control (MAC) addresses for each interface of each device into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

Dynamic routing uses protocols that enable the devices in a network to exchange routing table information with each other. The table is built and changed automatically. No administrative tasks are needed unless a time limit is added, so dynamic routing is more efficient than static routing. The default time limit is 4 hours. If the network has a great many routes that are added and deleted from the cache, the time limit should be adjusted.

The routing protocols that dynamic routing uses to learn routes, such as distance-vector and link-state, is beyond the scope of this document.

Note The Cisco IOS XE does not install the ARPs and forward entries instantaneously. So, there will be some delay in the installation process based on the system performance.

Devices That Do Not Use ARP

When a network is divided into two segments, a bridge joins the segments and filters traffic to each segment based on Media Access Control (MAC) addresses. The bridge builds its own address table, which uses MAC addresses only, as opposed to a router, which has an Address Resolution Protocol (ARP) cache that contains both IP addresses and the corresponding MAC addresses.

Passive hubs are central-connection devices that physically connect other devices in a network. They send messages out all ports to the devices and operate at Layer 1, but they do not maintain an address table.

Layer 2 switches determine which port is connected to a device to which the message is addressed and send the message only to that port, unlike a hub, which sends the message out all its ports. However, Layer 3 switches are routers that build an ARP cache (table).

Inverse ARP

Inverse ARP, which is enabled by default in ATM networks, builds an ATM map entry and is necessary to send unicast packets to a server (or relay agent) on the other end of a connection. Inverse ARP is supported only for the **aal5snap** encapsulation type.

For multipoint interfaces, an IP address can be acquired using other encapsulation types because broadcast packets are used. However, unicast packets to the other end will fail because there is no ATM map entry and thus DHCP renewals and releases also fail.

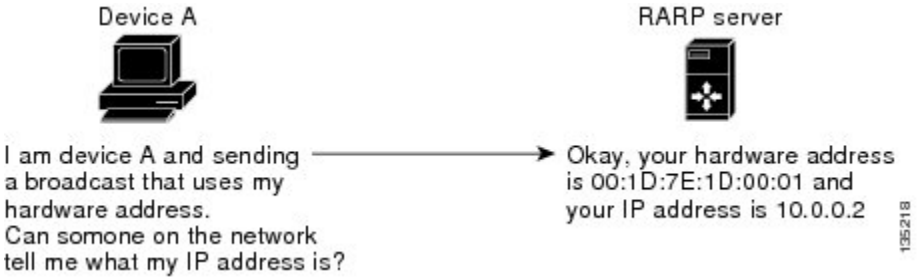
For more information about Inverse ARP and ATM networks, see the “Configuring ATM” feature module in the *Asynchronous Transfer Mode Configuration Guide*.

Reverse ARP

Reverse ARP (RARP) as defined by RFC 903 works the same way as the Address Resolution Protocol (ARP), except that the RARP request packet requests an IP address instead of a media access control (MAC) address. RARP often is used by diskless workstations because this type of device has no way to store IP addresses to use when they boot. The only address that is known is the MAC address because it is burned in to the hardware.

RARP requires a RARP server on the same network segment as the device interface. The figure below illustrates how RARP works.

Figure 59: RARP Process



Because of the limitations with RARP, most businesses use Dynamic Host Configuration Protocol (DHCP) to assign IP addresses dynamically. DHCP is cost-effective and requires less maintenance than RARP. The most important limitations with RARP are as follows:

- Because RARP uses hardware addresses, if the internetwork is large with many physical networks, a RARP server must be on every segment with an additional server for redundancy. Maintaining two servers for every segment is costly.
- Each server must be configured with a table of static mappings between the hardware addresses and the IP addresses. Maintenance of the IP addresses is difficult.
- RARP only provides IP addresses of the hosts but not subnet masks or default gateways.

Cisco software attempts to use RARP if it does not know the IP address of an interface at startup to respond to RARP requests that it is able to answer. The AutoInstall feature of the software automates the configuration of Cisco devices.

AutoInstall supports RARP and enables a network manager to connect a new device to a network, turn it on, and automatically load a pre-existing configuration file. The process begins when no valid configuration file is found in NVRAM. For more information about AutoInstall, see the *Configuration Fundamentals Configuration Guide*.

Proxy ARP

Proxy Address Resolution Protocol, as defined in RFC 1027, was implemented to enable devices that are separated into physical network segments connected by a router in the same IP network or subnetwork to resolve IP-to-MAC addresses. When devices are not in the same data link layer network but are in the same IP network, they try to transmit data to each other as if they were on the local network. However, the router that separates the devices will not send a broadcast message because routers do not pass hardware-layer broadcasts. Therefore, the addresses cannot be resolved.

Proxy ARP is enabled by default so the “proxy router” that resides between the local networks responds with its MAC address as if it were the router to which the broadcast is addressed. When the sending device receives

the MAC address of the proxy router, it sends the datagram to the proxy router, which in turns sends the datagram to the designated device.

Proxy ARP is invoked by the following conditions:

- The target IP address is not on the same physical network (LAN) on which the request is received.
- The networking device has one or more routes to the target IP address.
- All of the routes to the target IP address go through interfaces other than the one on which the request is received.

When proxy ARP is disabled, a device responds to ARP requests received on its interface only if the target IP address is the same as its IP address or if the target IP address in the ARP request has a statically configured ARP alias.

Serial Line Address Resolution Protocol

Serial Line ARP (SLARP) is used for serial interfaces that use High-Level Data Link Control (HDLC) encapsulation. A SLARP server, intermediate (staging) device, and another device providing a SLARP service might be required in addition to a TFTP server. If an interface is not directly connected to a server, the staging device is required to forward the address-resolution requests to the server. Otherwise, a directly connected device with SLARP service is required. Cisco software attempts to use SLARP if it does not know the IP address of an interface at startup to respond to SLARP requests that software is able to answer.

Cisco software automates the configuration of Cisco devices with the AutoInstall feature. AutoInstall supports SLARP and enables a network manager to connect a new device to a network, turn it on, and automatically load a pre-existing configuration file. The process begins when no valid configuration file is found in NVRAM. For more information about AutoInstall, see the *Configuration Fundamentals Configuration Guide*.

Note AutoInstall supports serial interfaces that use Frame Relay encapsulation.

Authorized ARP

Authorized ARP addresses a requirement of explicitly knowing when a user has logged off, either voluntarily or due to a failure of a network device. It is implemented for Public wireless LANs (WLANs) and DHCP. For more information about authorized ARP, refer to the “Configuring DHCP Services for Accounting and Security” chapter of the *DHCP Configuration Guide* , Cisco IOS Release 12.4.

Security (ARP/NDP cache entries) Enhancements

The Security (ARP/NDP cache entries) Enhancements feature implements ARP global limit and ARP interface limit. You can set a limit on the dynamic ARP entries per interface. Using the Security (ARP/NDP cache entries) Enhancements feature you can set a limit at either global level or interface level. Interface level configuration overrides the value of global limit when set. When the interface limit is not set, the global limit value is applied if the global limit is configured. When you disable interface-limit on an interface, you must execute the **no arp entries interface-limit** command to enable the interface-limit.

How to Configure the Address Resolution Protocol

By default, the Address Resolution Protocol (ARP) feature is enabled and is set to use Ethernet encapsulation. Perform the following tasks to change or verify ARP functionality:

Enabling the Interface Encapsulation

Perform this task to support a type of encapsulation for a specific network, such as Ethernet, Frame Relay, FDDI, or Token Ring. When Frame Relay encapsulation is specified, the interface is configured for a Frame Relay subnetwork with one physical link that has many logical circuits called virtual circuits (VCs). The address field in the frame contains a data-link connection identifier (DLCI) that identifies each VC. When SNAP encapsulation is specified, the interface is configured for FDDI or Token Ring networks.

Note The encapsulation type specified in this task should match the encapsulation type specified in the "Defining Static ARP Entries" task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **arp** {**arpa** | **frame-relay** | **snap**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Enters interface configuration mode.

	Command or Action	Purpose
Step 4	**arp** {**arpa** | **frame-relay** | **snap**} **Example:** `Device(config-if)# arp arpa`	Specifies the encapsulation type for an interface by type of network, such as Ethernet, FDDI, Frame Relay, and Token Ring. The keywords are as follows: • **arpa**—Enables encapsulation for an Ethernet 802.3 network. • **frame-relay**—Enables encapsulation for a Frame Relay network. • **snap**—Enables encapsulation for FDDI and Token Ring networks.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Defining Static ARP Entries

Perform this task to define static mapping between an IP address (32-bit address) and a Media Access Control (MAC) address (48-bit address) for hosts that do not support dynamic Address Resolution Protocol (ARP). Because most hosts support dynamic address resolution, defining static ARP cache entries is usually not required. Performing this task installs a permanent entry in the ARP cache that never times out. The entries remain in the ARP table until they are removed using the **no arp** command or the **clear arp interface** command for each interface.

Note The encapsulation type specified in this task should match the encapsulation type specified in the "Enabling the Interface Encapsulation" task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **arp** {*ip-address* | **vrf** *vrf-name*} *hardware-address encap-type* [*interface-type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**arp** {*ip-address* | **vrf** *vrf-name*} *hardware-address encap-type* [*interface-type*]
	Example:
	`Device(config)# arp 10.0.0.0 aabb.cc03.8200 arpa`
		Globally associates an IP address with a MAC address in the ARP cache.
		• *ip-address*—IP address in four-part dotted decimal format corresponding to the local data-link address.
		• **vrf** *vrf-name*—Virtual routing and forwarding instance for a Virtual Private Network (VPN). The *vrf-name* argument is the name of the VRF table.
		• *hardware-address*—Local data-link address (a 48-bit address).
		• *encap-type*—Encapsulation type for the static entry. The keywords are as follows:
		  • **arpa**—For Ethernet interfaces.
		  • **sap**—For Hewlett Packard interfaces.
		  • **smds**—For Switched Multimegabit Data Service (SMDS) interfaces.
		  • **snap**—For FDDI and Token Ring interfaces.
		  • **srp-a**—Switch route processor side A (SRP-A) interfaces.
		  • **srp-b**—Switch route processor side B (SRP-B) interfaces.
		Note Some keywords might not apply to your hardware platform.
		• *interface-type*—(Optional) Interface type (for more information, use the question mark (?) online help).

Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Setting an Expiration Time for Dynamic Entries in the ARP Cache

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **arp timeout** *seconds*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Enters interface configuration mode.
Step 4	**arp timeout** *seconds* **Example:** `Device(config-if)# arp timeout 30`	Sets the duration of time, in seconds, an Address Resolution Protocol (ARP) cache entry stays in the cache. The default is 14400 seconds (4 hours). The general recommended value for ARP timeout is the configured default value, which is 4 hours. If the network has frequent changes to cache entries, change the default to a shorter time period. As you reduce the ARP timeout, your network traffic increases. A low ARP timeout value might lead to network outage, and a value less than an hour (or 3600 seconds) will generate significantly increased traffic across the network. **Caution** We recommend that you set an ARP timeout value greater than 60 seconds.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Globally Disabling Proxy ARP

Proxy Address Resolution Protocol (ARP) is enabled by default; perform this task to globally disable proxy ARP on all interfaces.

The Cisco software uses proxy ARP (as defined in RFC 1027) to help hosts with no knowledge of routing determine the media access control (MAC) addresses of hosts on other networks or subnets. For example, if

hosts A and B are on different physical networks, host B does not receive the ARP broadcast request from host A and cannot respond to it. However, if the physical network of host A is connected by a gateway to the physical network of host B, the gateway sees the ARP request from host A.

Assuming that subnet numbers were assigned to correspond to physical networks, the gateway can also tell that the request is for a host that is on a different physical network. The gateway can then respond for host B, saying that the network address for host B is that of the gateway itself. Host A sees this reply, caches it, and sends future IP packets for host B to the gateway.

The gateway forwards such packets to host B by using the configured IP routing protocols. The gateway is also referred to as a transparent subnet gateway or ARP subnet gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip arp proxy disable**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip arp proxy disable** **Example:** `Device(config)# ip arp proxy disable`	Disables proxy ARP on all interfaces. • The **ip arp proxy disable** command overrides any proxy ARP interface configuration. • To reenable proxy ARP, use the **no ip arp proxy disable** command. • You can also use the **default ip proxy arp** command to return to the default proxy ARP behavior, which is enabled.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Disabling Proxy ARP on an Interface

Proxy Address Resolution Protocol (ARP) is enabled by default; perform this task to disable proxy ARP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip proxy-arp**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Enters interface configuration mode.
Step 4	**no ip proxy-arp** **Example:** `Device(config-if)# no ip proxy-arp`	Disables proxy ARP on the interface. • To reenable proxy ARP, use the **ip proxy-arp** command. • You can also use the **default ip proxy-arp** command to return to the default proxy ARP behavior on the interface, which is enabled.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Clearing the ARP Cache

Perform the following tasks to clear the Address Resolution Protocol (ARP) cache of entries associated with an interface and to clear all dynamic entries from the ARP cache, the fast-switching cache, and the IP route cache.

SUMMARY STEPS

1. **enable**
2. **clear arp interface** *type number*
3. **clear arp-cache**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear arp interface** *type number* **Example:** `Device# clear arp interface Gigabitethernet0/0/0`	Clears the entire ARP cache on the interface.
Step 3	**clear arp-cache** **Example:** `Device# clear arp-cache`	Clears all dynamic entries from the ARP cache, the fast-switching cache, and the IP route cache.
Step 4	**exit** **Example:** `Device# exit`	Returns to user EXEC mode.

Configuring Security (ARP/NDP cache entries) Enhancements

To configure ARP entry limit globally:

```
enable
configure terminal
 arp entries interface-limit 1 log 1
 end
```

To configure ARP entry limit on an interface:

```
enable
configure terminal
 interface Ethernet 0/0
    ip address 1.1.1.40 255.255.255.0
```

```
    arp entries interface-limit 1 log 1
    end
```

To disable ARP entry limit:

```
enable
configure terminal
 interface Ethernet 0/1
    ip address 2.1.1.1 255.255.255.0
    arp entries interface-limit disable
    end
```

Verifying the ARP Configuration

To verify the ARP configuration, perform the following steps.

SUMMARY STEPS

1. **show interfaces**
2. **show arp**
3. **show ip arp**
4. **show processes cpu | include (ARP|PID)**

DETAILED STEPS

Step 1 **show interfaces**

To display the type of ARP being used on a particular interface and also display the ARP timeout value, use the **show interfaces** EXEC command.

Example:

```
Router# show interfaces
Ethernet 0 is up, line protocol is up
 Hardware is MCI Ethernet, address is 0000.0c00.750c (bia 0000.0c00.750c)
 Internet address is 10.108.28.8, subnet mask is 255.255.255.0
 MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
 Encapsulation ARPA, loopback not set, keepalive set (10 sec)
 ARP type: ARPA, ARP Timeout 4:00:00
 Last input 0:00:00, output 0:00:00, output hang never
 Last clearing of "show interface" counters 0:00:00
 Output queue 0/40, 0 drops; input queue 0/75, 0 drops
 Five minute input rate 0 bits/sec, 0 packets/sec
 Five minute output rate 2000 bits/sec, 4 packets/sec
 1127576 packets input, 447251251 bytes, 0 no buffer
 Received 354125 broadcasts, 0 runts, 0 giants, 57186* throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  5332142 packets output, 496316039 bytes, 0 underruns
  0 output errors, 432 collisions, 0 interface resets, 0 restarts
```

Step 2 **show arp**

Use the **show arp** EXEC command to examine the contents of the ARP cache.

Example:

```
Router# show arp
```

```
Protocol    Address            Age (min)   Hardware Addr      Type    Interface
Internet    10.108.42.112      120         0000.a710.4baf     ARPA    Ethernet3
AppleTalk   4028.5             29          0000.0c01.0e56     SNAP    Ethernet2
Internet    110.108.42.114     105         0000.a710.859b     ARPA    Ethernet3
AppleTalk   4028.9             -           0000.0c02.a03c     SNAP    Ethernet2
Internet    10.108.42.121      42          0000.a710.68cd     ARPA    Ethernet3
Internet    10.108.36.9        -           0000.3080.6fd4     SNAP    TokenRing0
AppleTalk   4036.9             -           0000.3080.6fd4     SNAP    TokenRing0
Internet    10.108.33.9        -           0000.0c01.7bbd     SNAP    Fddi0
```

Step 3 show ip arp

Use the **show ip arp** EXEC command to show IP entries. To remove all nonstatic entries from the ARP cache, use the **clear arp-cache**privileged EXEC command.

Example:

```
Router# show ip arp
Protocol    Address           Age(min)      Hardware Addr     Type     Interface
Internet    171.69.233.22     9             0000.0c59.f892    ARPA     Ethernet0/0
Internet    171.69.233.21     8             0000.0c07.ac00    ARPA     Ethernet0/0
Internet    171.69.233.19     -             0000.0c63.1300    ARPA     Ethernet0/0
Internet    171.69.233.30     9             0000.0c36.6965    ARPA     Ethernet0/0
Internet    172.19.168.11     -             0000.0c63.1300    ARPA     Ethernet0/0
Internet    172.19.168.254    9             0000.0c36.6965    ARPA     Ethernet0/0
```

Step 4 show processes cpu | include (ARP|PID)

Use the **show processes cpu** | **include (ARP|PID)** command to display ARP and RARP processes.

Example:

```
Router# show processes cpu | include (ARP|PID)
PID      Runtime(ms) Invoked uSecs   5Sec  1Min  5Min    TTY Process
1          1736         58   29931     0%    0%    0%    Check heaps
2            68        585     116  1.00% 1.00%    0%    IP Input
3             0        744       0     0%    0%    0%    TCP Timer
4             0          2       0     0%    0%    0%    TCP Protocols
5             0          1       0     0%    0%    0%    BOOTP Server
6            16        130     123     0%    0%    0%    ARP Input
7             0          1       0     0%    0%    0%    Probe Input
8             0          7       0     0%    0%    0%    MOP Protocols
9             0          2       0     0%    0%    0%    Timers
10          692         64   10812     0%    0%    0%   Net Background
11            0          5       0     0%    0%    0%   Logger
12            0         38       0     0%    0%    0%   BGP Open
13            0          1       0     0%    0%    0%   Net Input
14          540       3466     155     0%    0%    0%   TTY Background
15            0          1       0     0%    0%    0%   BGP I/O
16         5100       1367    3730     0%    0%    0%   IGRP Router
17           88       4232      20 0.20% 1.00%    0%   BGP Router
18          152      14650      10     0%    0%    0%   BGP Scanner
19          224         99    2262     0%    0% 1.00%   Exec
```

Configuration Examples for the Address Resolution Protocol

Example: Static ARP Entry Configuration

The following example shows how to configure a static Address Resolution Protocol (ARP) entry in the cache by using the **alias** keyword, allowing the software to respond to ARP requests as if it were the interface of the specified address:

```
arp 10.0.0.0 aabb.cc03.8200 alias
interface gigabitethernet0/0/0
```

Example: Encapsulation Type Configuration

The following example shows how to configure the encapsulation on the interface. The **arpa** keyword indicates that interface is connected to an Ethernet 802.3 network:

```
interface gigabitethernet0/0/0
 ip address 10.108.10.1 255.255.255.0
 arp arpa
```

Example: Proxy ARP Configuration

The following example shows how to configure proxy ARP because it was disabled for the interface:

```
interface gigabitethernet0/0/0
 ip proxy-arp
```

Examples: Clearing the ARP Cache

The following example shows how to clear all entries in the ARP cache associated with an interface:

```
Device# clear arp interface gigabitethernet0/0/0
```

The following example shows how to clear all dynamic entries in the ARP cache:

```
Device# clear arp-cache
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
ARP commands	Cisco IOS IP Addressing Services Command Reference

Related Topic	Document Title
AppleTalk addressing scheme	Core Competence AppleTalk (white paper) at www.corecom.com/html/appletalk.html
Authorized ARP	"Configuring DHCP Services for Accounting and Security" feature module in the *IP Addressing: DHCP Configuration Guide* (part of the *IP Addressing Configuration Guide Library*)
Inverse ARP and ATM networks	"Configuring ATM" feature module in the *Asynchronous Transfer Mode Configuration Guide*
AutoInstall	*Configuration Fundamentals Configuration Guide*

RFCs

RFCs	Title
RFC 826	*Address Resolution Protocol*
RFC 903	*Reverse Address Resolution Protocol*
RFC 1027	*Proxy Address Resolution Protocol*
RFC 1042	*Standard for the Transmission of IP Datagrams over IEEE 802 Networks*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the Address Resolution Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

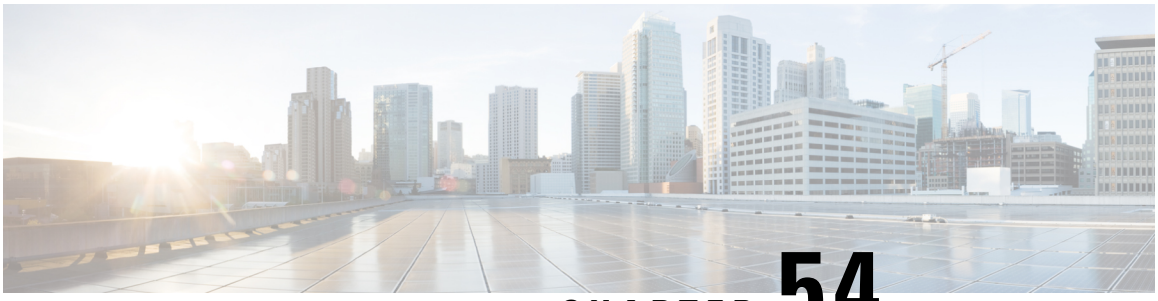
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 79: Feature Information for the Address Resolution Protocol

Feature Name	Software Releases	Feature Information
Address Resolution Protocol		The Address Resolution Protocol (ARP) feature performs a required function in IP routing. ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. ARP maintains a cache (table) in which MAC addresses are mapped to IP addresses. ARP is part of all Cisco systems that run IP.
Security (ARP/NDP cache entries) Enhancements	Cisco IOS XE Everest 16.4.1	The Security (ARP/NDP cache entries) Enhancements feature implements ARP global limit and ARP interface limit. You can set a limit on the dynamic ARP entries per interface. The following command was introduced: **arp entries interface-limit**

PART VI

DHCP



CHAPTER 54

Configuring the Cisco IOS XE DHCP Server

Cisco routers running Cisco IOS XE software include Dynamic Host Configuration Protocol (DHCP) server and relay agent software. The Cisco IOS XE DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the domain name system (DNS) server and the default router.

This module describes the concepts and the tasks needed to configure the DHCP server.

Prerequisites for Configuring the DHCP Server

- Before you configure a Cisco Dynamic Host Control Protocol (DHCP) server, you must understand the concepts documented in the Overview of the DHCP Server section.
- The Cisco DHCP server and the relay agent services are enabled by default. Use the **no service dhcp** command to disable the Cisco DHCP server and the relay agent and the **service dhcp** command to reenable the functionality.
- Port 67 (the DHCP server port) is closed in the Cisco DHCP/BOOTP default configuration. There are two logical parts to the **service dhcp** command: service enabled and service running. The DHCP service is enabled by default, but port 67 does not open until the DHCP service is running. If the DHCP service is running, the **show ip sockets details** or the **show sockets detail** command displays port 67 as open.
- The Cisco DHCP relay agent is enabled on an interface only when you configure the **ip helper-address** command. This command enables a DHCP broadcast to be forwarded to the configured DHCP server.

Information About the Cisco IOS XE DHCP Server

Overview of the DHCP Server

The Cisco DHCP server accepts address assignment requests and renewals from the client and assigns the addresses from predefined groups of addresses within DHCP address pools. These address pools can also be configured to supply additional information to the requesting client such as the IP address of the Domain Name System (DNS) server, the default device, and other configuration parameters. The Cisco DHCP server can accept broadcasts from locally attached LAN segments or from DHCP requests that have been forwarded by other DHCP relay agents within the network.

Database Agents

A DHCP database agent is any host (for example, an FTP, TFTP, or rcp server) or storage media on the DHCP server (for example, flash disk) that stores the DHCP bindings database. You can configure multiple DHCP database agents, and you can configure the interval between database updates and transfers for each agent.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic binding information (such as lease expiration date and time, interface index, and VPN routing and forwarding [VRF] name) is stored on a database agent. The bindings are saved as text records for easy maintenance.

Address Conflicts

An address conflict occurs when two hosts use the same IP address. During address assignment, DHCP checks for conflicts using ping and gratuitous Address Resolution Protocol (ARP). If a conflict is detected, the address is removed from the pool. The address will not be assigned until the administrator resolves the conflict.

DHCP Address Pool Conventions

You can configure a DHCP address pool with a name that is a symbolic string (such as “engineering”) or an integer (such as 0). Configuring a DHCP address pool also puts the router into DHCP pool configuration mode--identified by the (dhcp-config)# prompt--from which you can configure pool parameters (for example, the IP subnet number and default router list).

DHCP Address Pool Selection

DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. The process by which the DHCP server identifies which DHCP address pool to use to service a client request is described in this section.

The DHCP server identifies which DHCP address pool to use to service a client request as follows:

- If the client is not directly connected (the giaddr field of the DHCPDISCOVER broadcast message is non-zero), the DHCP server matches the DHCPDISCOVER with a DHCP pool that has the subnet that contains the IP address in the giaddr field. Giaddr field is the gateway IP address field of a DHCP packet. A DHCP relay agent sets the gateway address and adds the relay agent information option (option82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing option 82.

- If the client is directly connected (the giaddr field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pool(s) that contain the subnet(s) configured on the receiving interface. If the interface has secondary IP addresses, the subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.

Cisco IOS XE DHCP server software supports advanced capabilities for IP address allocation. See the "DHCP Server Address Allocation Using Option 82" section for more information.

Address Bindings

An address binding is a mapping between the IP address and MAC address of a client. The IP address of a client can be assigned manually by an administrator or assigned automatically from a pool by a DHCP server.

Manual bindings are IP addresses that have been manually mapped to the MAC addresses of hosts that are found in the DHCP database. Manual bindings are stored in NVRAM on the DHCP server. Manual bindings are just special address pools. There is no limit on the number of manual bindings, but you can only configure one manual binding per host pool.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Because the bindings are stored in volatile memory on the DHCP server, binding information is lost in the event of a power failure or upon router reload for any other reason. To prevent the loss of automatic binding information in such an event, a copy of the automatic binding information can be stored on a remote host called a DHCP database agent. The bindings are periodically written to the database agent. If the router reloads, the bindings are read back from the database agent to the DHCP database on the DHCP server.

Note We strongly recommend using database agents. However, the Cisco IOS XE DHCP server can function without database agents.

All DHCP clients send a client identifier (DHCP option 61) in the DHCP packet. To configure manual bindings, you must enter the **client-identifier** DHCP pool configuration command with the appropriate hexadecimal values identifying the DHCP client.

Ping Packet Settings

By default, the DHCP server pings a pool address twice before assigning a particular address to a requesting client. If the ping is unanswered, the DHCP server assumes (with a high probability) that the address is not in use and assigns the address to the requesting client.

By default, the DHCP server waits 2 seconds before timing out a ping packet.

DHCP Attribute Inheritance

The DHCP server database is organized as a tree. The root of the tree is the address pool for natural networks, branches are subnetwork address pools, and leaves are manual bindings to clients. Subnetworks inherit network parameters and clients inherit subnetwork parameters. Therefore, common parameters (for example, the domain name) should be configured at the highest (network or subnetwork) level of the tree.

Inherited parameters can be overridden. For example, if a parameter is defined in both the natural network and a subnetwork, the definition of the subnetwork is used.

Address leases are not inherited. If a lease is not specified for an IP address, by default, the DHCP server assigns a one-day lease for the address.

DHCP Server Address Allocation Using Option 82

The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) sent by the relay agent.

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items are also called options. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

Automatic DHCP address allocation is based on an IP address. This IP address can either be the gateway address (giaddr field of the DHCP packet) or the IP address of an incoming interface. In some networks, it is necessary to use additional information to further determine which IP addresses to allocate. By using option 82, the Cisco IOS DHCP relay agent can include additional information about itself when forwarding client-originated DHCP packets to a DHCP server. The Cisco IOS DHCP server can also use option 82 to provide additional information to properly allocate IP addresses to DHCP clients. The information sent via option 82 is used to identify the port where the DHCP request arrives. Automatic DHCP address allocation does not parse out the individual suboptions contained in option 82. Rather, the address allocation is done by matching a configured pattern byte by byte.

This feature introduces a new DHCP class capability, which is a method to group DHCP clients based on some shared characteristics other than the subnet in which the clients reside.

For example, DHCP clients are connected to two ports of a single switch. Each port can be configured to be a part of two VLANs: VLAN1 and VLAN2. DHCP clients belong to either VLAN1 or VLAN2 and the switch can differentiate the VLAN that a particular DHCP Discover message belongs to (possibly through Layer 2 encapsulation). Each VLAN has its own subnet and all DHCP messages from the same VLAN (same switch) have the giaddr field set to the same value indicating the subnet of the VLAN.

Problems can occur while allocating IP addresses to DHCP clients that are connected to different ports of the same VLAN. These IP addresses must be part of the same subnet but the range of IP addresses must be different. In the preceding example, when a DHCP client that is connected to a port of VLAN1 must be allocated an IP address from a range of IP addresses within the VLAN's subnet, whereas a DHCP client connecting to port 2 of VLAN1 must be allocated an IP address from another range of IP addresses. The two range of IP addresses are part of the same subnet (and have the same subnet mask). Generally, during DHCP address allocation, the DHCP server refers only to the giaddr field and is unable to differentiate between the two ranges.

To solve this problem, a relay agent residing at the switch inserts the relay information option (option 82), which carries information specific to the port, and the DHCP server inspects both the giaddr field and the inserted option 82 during the address selection process.

When you enable option 82 on a device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the device receives the DHCP request, it adds the option 82 information in the packet. The option 82 information contains the device MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
3. The device adds the IP address of the relay agent to the DHCP packet.
4. The device forwards the DHCP request that includes the option 82 field to the DHCP server.

5. The DHCP server receives the packet. If the server is option 82 capable, it uses the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the option 82 field in the DHCP reply.
6. The DHCP server unicasts the reply to the device if the request is relayed to the server by the device. The device verifies that it originally inserted the option 82 data by inspecting remote ID and possibly circuit ID fields. The device removes the option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

The Cisco software refers to a pool of IP addresses (giaddr or incoming interface IP address) and matches the request to a class or classes configured in the pool in the order the classes are specified in the DHCP pool configuration.

When a DHCP address pool is configured with one or more DHCP classes, the pool becomes a restricted access pool, which means that no addresses are allocated from the pool unless one or more classes in the pool matches. This design allows DHCP classes to be used either for access control (no default class is configured on the pool) or to provide further address range partitions within the subnet of the pool.

Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.

The following capabilities are supported for DHCP class-based address allocation:

- Specifying the full relay agent information option value as a raw hexadecimal string by using the **relay-information hex** command in new relay agent information configuration mode.
- Support for bit-masking the raw relay information hexadecimal value.
- Support for a wildcard at the end of a hexadecimal string specified by the **relay-information hex** command.

If the relay agent inserts option 82 but does not set the giaddr field in the DHCP packet, the DHCP server interface must be configured as a trusted interface by using the **ip dhcp relay information trusted** command. This configuration prevents the server from dropping the DHCP message.

DHCP Address Allocation Using Option 82 Feature Design

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items themselves are also called options. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

This feature is designed to allow the Cisco IOS XE DHCP server to use option 82 information to help determine which IP addresses to allocate to clients. The information sent via option 82 will be used to identify which port the DHCP request came in on. This feature does not parse out the individual suboptions contained within option 82. Rather, the address allocation is done by matching a configured pattern byte by byte.

The feature introduces a new DHCP class capability, which is a method to group DHCP clients based on some shared characteristics other than the subnet in which the clients reside.

Usage Scenario for DHCP Address Allocation Using Option 82

In an example application, DHCP clients are connected to two ports of a single switch. Each port can be configured to be part of two VLANs: VLAN1 and VLAN2. DHCP clients belong to either VLAN1 or VLAN2 and it is assumed that the switch can differentiate the VLAN that a particular DHCP Discover message belongs to (possibly through Layer 2 encapsulation). Each VLAN has its own subnet and all DHCP messages from

the same VLAN (same switch) will have the giaddr field set to the same value indicating the subnet of the VLAN.

The problem is that for a DHCP client connecting to port 1 of VLAN1, it must be allocated an IP address from one range within the VLAN's subnet, whereas a DHCP client connecting to port 2 of VLAN1 must be allocated an IP address from another range. Both these two IP address ranges are part of the same subnet (and have the same subnet mask). In the normal DHCP address allocation, the DHCP server will look only at the giaddr field and thus will not be able to differentiate between the two ranges.

To solve this problem, a relay agent residing at the switch inserts the relay information option (option 82), which carries information specific to the port, and the DHCP server must inspect both the giaddr field and the inserted option 82 during the address selection process.

DHCP Class Capability

The Cisco IOS XE software will look up a pool based on IP address (giaddr or incoming interface IP address) and then match the request to a class or classes configured in the pool in the order the classes are specified in the DHCP pool configuration.

When a DHCP address pool has been configured with one or more DHCP classes, the pool becomes a restricted access pool, which means that no addresses will be allocated from the pool unless one or more of the classes in the pool is matched. This design allows DHCP classes to be used for either access control (no default class is configured on the pool) or to provide further address range partitions with the subnet of the pool.

Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.

The following capabilities are currently supported for DHCP class-based address allocation:

- Specifying the full relay agent information option value as a raw hexadecimal string by using the **relay-information hex** command in the new relay agent information configuration mode.
- Support for bitmasking the raw relay information hexadecimal value.
- Support for a wildcard at the end of the hexadecimal string specified by the **relay-information hex** command.

RegEx and Longest Match Support

DHCP server software supports advanced capabilities for IP address allocation. Earlier, DHCP server supported only exact match on hexadecimal codes. Effective with Cisco IOS XE Fuji 16.9.1, DHCP server is enhanced to support Regular expression (RegEx) based match or longest match. DHCP server provides options to set of DHCP clients with Vendor Class ID (VCI). Each set of clients are serviced from specific DHCP pool with one or more Vendor Classes. RegEx based Vendor Class Identifier match is included to support this feature.

For one class option, either Exact Match or Regex Match or Longest Match is supported. The configured Regex or hexadecimal string is matched against VCI string received in DHCP packets. In case of successful match, server assigns an IP address from the address range specified in pool class configuration. In case of multiple class match, the first occurrence of the match is considered. In case of no match, no address is allocated.

How to Configure the Cisco IOS XE DHCP Server

Configuring a DHCP Database Agent or Disabling Conflict Logging

A DHCP database agent is any host (for example, an FTP, a TFTP, or a remote copy protocol [RCP] server) or storage media on a DHCP server (for example, disk0) that stores the DHCP bindings database. You can configure multiple DHCP database agents, and the interval between database updates and transfers for each agent.

Automatic bindings are IP addresses that are automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic binding information (such as lease expiration date and time, interface index, and VPN routing and forwarding [VRF] name) is stored in a database agent. The bindings are saved as text records for easy maintenance.

An address conflict occurs when two hosts use the same IP address. During address assignment, DHCP checks for conflicts by using ping and gratuitous Address Resolution Protocol (ARP). If a conflict is detected, the address is removed from the pool. The address is not assigned until the administrator resolves the conflict.

Note We strongly recommend using database agents. However, the Cisco DHCP server can run without database agents. If you choose not to configure a DHCP database agent, disable the recording of DHCP address conflicts on the DHCP server by using the **no ip dhcp conflict logging** command in global configuration mode. If there is a conflict logging but no database agent is configured, bindings during a switchover are lost when a device reboots. Possible false conflicts can occur causing the address to be removed from the address pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip dhcp database** *url* [**timeout** *seconds* | **write-delay** *seconds*]
   - **no ip dhcp conflict logging**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 3	Do one of the following:
	• **ip dhcp database** *url* [**timeout** *seconds* | **write-delay** *seconds*]
	• **no ip dhcp conflict logging**
	Example:
	`Device(config)# ip dhcp database ftp://user:password@172.16.1.1/router-dhcp timeout 80`
	Example:
	`Device(config)# no ip dhcp conflict logging`
		Configures a DHCP server to save automatic bindings on a remote host, known as a database agent. When setting this up, ensure the file name is added as part of the ftp/tftp URL. Alternatively, you can choose to disable DHCP address conflict logging.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Excluding IP Addresses

The IP address configured on a device interface is automatically excluded from the DHCP address pool. The DHCP server assumes that all other IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients.

You must exclude addresses from the pool if the DHCP server does not allocate those IP addresses to DHCP clients. Consider a scenario where two DHCP servers are set up for the same network segment (subnet) for redundancy. If DHCP servers do not coordinate their services with each other using a protocol such as DHCP failover, each DHCP server must be configured to allocate addresses from a nonoverlapping set of addresses in the shared subnet. See the Configuring Manual Bindings section for a configuration example.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp excluded-address** *low-address* [*high-address*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp excluded-address** *low-address* [*high-address*] **Example:** `Device(config)# ip dhcp excluded-address 172.16.1.100 172.16.1.103`	Specifies IP addresses that the DHCP server should not assign to DHCP clients.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring DHCP Address Pools

Configuring a DHCP Address Pool

On a per-address pool basis, specify DHCP options for the client as necessary.

You can configure a DHCP address pool with a name that is a string (such as “engineering”) or an integer (such as 0). Configuring a DHCP address pool also puts the device into DHCP pool configuration mode—identified by the (dhcp-config)# prompt—from which you can configure pool parameters (for example, the IP subnet number and default device list).

DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. The process by which the DHCP server identifies the DHCP address pool to use for a client request is described in the Configuring Manual Bindings section.

The DHCP server identifies and uses DHCP address pools for a client request, in the following manner:

- If the client is not directly connected to the DHCP server (the giaddr field of the DHCPDISCOVER broadcast message is nonzero), the server matches the DHCPDISCOVER with the DHCP pool that has the subnet that contains the IP address in the giaddr field.
- If the client is directly connected to the DHCP server (the giaddr field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pools that contain the subnets configured on the receiving interface. If the interface has secondary IP addresses, subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.
- If you want to unconfigure the last DHCP pool when the DHCP traffic is active, ensure to have one of the following configurations:
  1. **ip dhcp pool** <*dummy_pool*> any dummy pool without any configuration

     OR
  2. **ip helper-address** x.x.x.x on dummy interface(loopback)

Cisco DHCP server software supports advanced capabilities for IP address allocation. See the Configuring DHCP Address Allocation Using Option 82 section for more information.

Before you begin

Before you configure the DHCP address pool, you must:

- Identify DHCP options for devices where necessary, including the following:
  - Default boot image name
  - Default devices
  - Domain Name System (DNS) servers
  - Network Basic Input/Output System (NetBIOS) name server
  - Primary subnet
  - Secondary subnets and subnet-specific default device lists (see Configuring a DHCP Address Pool with Secondary Subnets for information on secondary subnets).
- Decide on a NetBIOS node type (b, p, m, or h).
- Decide on a DNS domain name.

Note You cannot configure manual bindings within the same pool that is configured with the **network** DHCP pool configuration command. To configure manual bindings, see the Configuring Manual Bindings section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **utilization mark high** *percentage-number* [**log**]
5. **utilization mark low** *percentage-number* [**log**]
6. **network** *network-number* [*mask* | */prefix-length*] [**secondary**]
7. **domain-name** *domain*
8. **dns-server** *address* [*address2 ... address8*]
9. **bootfile** *filename*
10. **next-server** *address* [*address2 ... address8*]
11. **netbios-name-server** *address* [*address2 ... address8*]
12. **netbios-node-type** *type*
13. **default-router** *address* [*address2 ... address8*]
14. **option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*}
15. **import** {**all** | **interface** *interface_name*
16. **lease** {*days* [*hours* [*minutes*]] | **infinite**}
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device(config)# ip dhcp pool 1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**utilization mark high** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark high 80 log`	(Optional) Configures the high utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.
Step 5	**utilization mark low** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark low 70 log`	(Optional) Configures the low utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	**network** *network-number* [*mask* | */prefix-length*] [**secondary**] **Example:** `Device(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.
Step 7	**domain-name** *domain* **Example:** `Device(dhcp-config)# domain-name cisco.com`	Specifies the domain name for the client.
Step 8	**dns-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command.

	Command or Action	Purpose
		• Servers should be listed in order of preference.
Step 9	**bootfile** *filename* **Example:** `Device(dhcp-config)# bootfile xllboot`	(Optional) Specifies the name of the default boot image for a DHCP client. • The boot file is used to store the boot image for the client. The boot image is generally the operating system that the client uses to load.
Step 10	**next-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# next-server 172.17.1.103 172.17.2.103`	(Optional) Configures the next server in the boot process of a DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • If multiple servers are specified, DHCP assigns them to clients in a round-robin order. The first client gets address 1, the next client gets address 2, and so on. • If this command is not configured, DHCP uses the server specified by the **ip helper address** command as the boot server.
Step 11	**netbios-name-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103`	(Optional) Specifies the NetBIOS WINS server that is available to a Microsoft DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • Servers should be listed in order of preference.
Step 12	**netbios-node-type** *type* **Example:** `Device(dhcp-config)# netbios-node-type h-node`	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.
Step 13	**default-router** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# default-router 172.16.1.100 172.16.1.101`	(Optional) Specifies the IP address of the default device for a DHCP client. • The IP address should be on the same subnet as the client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. These default devices are listed in order of preference; that is, *address* is the most preferred device, *address2* is the next most preferred device, and so on. • When a DHCP client requests an IP address, the device—acting as a DHCP server—accesses the default device list to select another device that the DHCP client will use as the first hop for forwarding

	Command or Action	Purpose
		messages. After a DHCP client has booted, the client begins sending packets to its default device.
Step 14	**option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*} **Example:** `Device(dhcp-config)# option 19 hex 01`	(Optional) Configures DHCP server options. Configuration supports Longest match and RegEx match for option 60. The **option code** sub command can be used to configure any DHCP options.

Step 15	**import** {**all** | **interface** *interface_name*
	Example:
	`Device(dhcp-config)# import all`
	`Device(dhcp-config) # import interface Ethernet0/0`
		The **import all** command learns options from all the interfaces.
		The **import interface** learns options only from the specified interface.

Step 16	**lease** {*days* [*hours* [*minutes*]] | **infinite**} **Example:** `Device(dhcp-config)# lease 30`	(Optional) Specifies the duration of the lease. • The default is a one-day lease. • The **infinite** keyword specifies that the duration of the lease is unlimited.
Step 17	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.

Configuring a DHCP Address Pool with Secondary Subnets

For any DHCP pool, you can configure a primary subnet and any number of secondary subnets. Each subnet is a range of IP addresses that the device uses to allocate an IP address to a DHCP client. The DHCP server multiple subnet functionality enables a Cisco DHCP server address pool to manage additional IP addresses by adding the addresses to a secondary subnet of an existing DHCP address pool (instead of using a separate address pool).

Configuring a secondary DHCP subnetwork places the device in DHCP pool secondary subnet configuration mode—identified by the (config-dhcp-subnet-secondary)# prompt—where you can configure a default address list that is specific to the secondary subnet. You can also specify the utilization rate of the secondary subnet, which allows pools of IP addresses to dynamically increase or reduce in size depending on the address utilization level. This setting overrides the global utilization rate.

If the DHCP server selects an address pool that contains multiple subnets, the DHCP server allocates an IP address from the subnets as follows:

- When the DHCP server receives an address assignment request, it looks for an available IP address in the primary subnet.
- When the primary subnet is exhausted, the DHCP server automatically looks for an available IP address in any of the secondary subnets maintained by the DHCP server (even though the giaddr does not

necessarily match the secondary subnet). The server inspects the subnets for address availability in the order of subnets that were added to the pool.

- If the giaddr matches a secondary subnet in the pool, the DHCP server allocates an IP address from that particular secondary subnet (even if IP addresses are available in the primary subnet and irrespective of the order of secondary subnets that were added).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **utilization mark high** *percentage-number* [**log**]
5. **utilization mark low** *percentage-number* [**log**]
6. **network** *network-number* [*mask* | */prefix-length*]
7. **domain-name** *domain*
8. **dns-server** *address* [*address2 ... address8*]
9. **bootfile** *filename*
10. **next-server** *address* [*address2 ... address8*]
11. **netbios-name-server** *address* [*address2 ... address8*]
12. **netbios-node-type** *type*
13. **default-router** *address* [*address2 ... address8*]
14. **option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*}
15. **lease** {*days* [*hours*] [*minutes*] | **infinite**}
16. **network** *network-number* [*mask* | */prefix-length*] [**secondary**]
17. **override default-router** *address* [*address2 ... address8*]
18. **override utilization high** *percentage-number*
19. **override utilization low** *percentage-number*
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device(config)# ip dhcp pool 1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.

	Command or Action	Purpose
Step 4	**utilization mark high** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark high 80 log`	(Optional) Configures the high utilization mark of the current address pool size. • The **log** keyword enables logging of a system message. A system message is generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.
Step 5	**utilization mark low** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark low 70 log`	(Optional) Configures the low utilization mark of the current address pool size. • The **log** keyword enables logging of a system message. A system message is generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet network number and mask of the primary DHCP address pool.
Step 7	**domain-name** *domain* **Example:** `Device(dhcp-config)# domain-name cisco.com`	Specifies the domain name for the client.
Step 8	**dns-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command. • Servers should be listed in the order of preference.
Step 9	**bootfile** *filename* **Example:** `Device(dhcp-config)# bootfile xllboot`	(Optional) Specifies the name of the default boot image for a DHCP client. • The boot file is used to store the boot image for the client. The boot image is generally the operating system image that the client loads.
Step 10	**next-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# next-server 172.17.1.103 172.17.2.103`	(Optional) Configures the next server in the boot process of a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. • If multiple servers are specified, DHCP assigns the servers to clients in a round-robin order. The first client gets address 1, the next client gets address 2, and so on.

	Command or Action	Purpose
		• If this command is not configured, DHCP uses the server specified by the **ip helper address** command as the boot server.
Step 11	**netbios-name-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103`	(Optional) Specifies the NetBIOS Windows Internet Naming Service (WINS) server that is available to a Microsoft DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • Servers should be listed in order of preference.
Step 12	**netbios-node-type** *type* **Example:** `Device(dhcp-config)# netbios-node-type h-node`	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.
Step 13	**default-router** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# default-router 172.16.1.100 172.16.1.101`	(Optional) Specifies the IP address of the default device for a DHCP client. • The IP address should be on the same subnet as the client. • One IP address is required; however, you can specify a up to eight IP addresses in one command line. These default devices are listed in order of preference; that is, *address* is the most preferred device, *address2* is the next most preferred device, and so on. • When a DHCP client requests for an IP address, the device—acting as a DHCP server—accesses the default device list to select another device that the DHCP client uses as the first hop for forwarding messages. After a DHCP client has booted, the client begins sending packets to its default device.
Step 14	**option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*} **Example:** `Device(dhcp-config)# option 19 hex 01`	(Optional) Configures DHCP server options.
Step 15	**lease** {*days* [*hours*] [*minutes*] | **infinite**} **Example:** `Device(dhcp-config)# lease 30`	(Optional) Specifies the duration of the lease. • The default is a one-day lease. • The **infinite** keyword specifies that the duration of the lease is unlimited.

	Command or Action	Purpose
Step 16	**network** *network-number* [*mask* | */prefix-length*] [**secondary**] **Example:** `Device(dhcp-config)# network 10.10.0.0 255.255.0.0 secondary`	(Optional) Specifies the network number and mask of a secondary DHCP server address pool. • Any number of secondary subnets can be added to a DHCP server address pool. • During execution of this command, the configuration mode changes to DHCP pool secondary subnet configuration mode, which is identified by (config-dhcp-subnet-secondary)# prompt. In this mode, the administrator can configure a default device list that is specific to the subnet. • See Troubleshooting Tips section if you are using secondary IP addresses under a loopback interface with DHCP secondary subnets.
Step 17	**override default-router** *address* [*address2 ... address8*] **Example:** `Device(config-dhcp-subnet-secondary)# override default-router 10.10.0.100 10.10.0.101`	(Optional) Specifies the default device list that is used when an IP address is assigned to a DHCP client from a particular secondary subnet. • If the subnet-specific override value is configured, this override value is used when assigning an IP address from the subnet; the network-wide default device list is used only to set the gateway device for the primary subnet. • If this subnet-specific override value is not configured, the network-wide default device list is used when assigning an IP address from the subnet. • See Example: Configuring a DHCP Address Pool with Multiple Disjoint Subnets section for a sample configuration.
Step 18	**override utilization high** *percentage-number* **Example:** `Device(config-dhcp-subnet-secondary)# override utilization high 60`	(Optional) Sets the high utilization mark of the subnet size. • This command overrides the global default setting specified by the **utilization mark high** command.
Step 19	**override utilization low** *percentage-number* **Example:** `Device(config-dhcp-subnet-secondary)# override utilization low 40`	(Optional) Sets the low utilization mark of the subnet size. • This command overrides the global default setting specified by the **utilization mark low** command.
Step 20	**end** **Example:** `Device(config-dhcp-subnet-secondary)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

If you are using secondary IP addresses under a single loopback interface and using secondary subnets under a DHCP pool, use one DHCP pool to configure networks for all the secondary subnets instead of using one pool per secondary subnet. The **network** *network-number* [*mask* | */prefix-length*] [**secondary**] command must be configured under a single DHCP address pool rather than multiple DHCP address pools.

The following is the correct configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 network 172.16.2.0 255.255.255.0 secondary
 network 172.16.3.0 255.255.255.0 secondary
 network 172.16.4.0 255.255.255.0 secondary
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

The following is the incorrect configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_2
 network 172.16.2.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_3
 network 172.16.3.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_4
 network 172.16.4.0 255.255.255.0
 lease 1 20 30
 accounting default
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

Verifying the DHCP Address Pool Configuration

The following configuration commands are optional. You can enter the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **show ip dhcp pool** [*name*]
3. **show ip dhcp binding** [*address*]

4. **show ip dhcp conflict** [*address*]
5. **show ip dhcp database** [*url*]
6. **show ip dhcp server statistics** [*type-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dhcp pool** [*name*] **Example:** `Device# show ip dhcp pool`	(Optional) Displays information about DHCP address pools.
Step 3	**show ip dhcp binding** [*address*] **Example:** `Device# show ip dhcp binding`	(Optional) Displays a list of all bindings created on a specific DHCP server. • Use the **show ip dhcp binding** command to display the IP addresses that have already been assigned. Verify that the address pool is not exhausted. If necessary, recreate the pool to create a larger pool of addresses. • Use the **show ip dhcp binding** command to display the lease expiration date and time of the IP address of the host.
Step 4	**show ip dhcp conflict** [*address*] **Example:** `Device# show ip dhcp conflict`	(Optional) Displays a list of all IP address conflicts.
Step 5	**show ip dhcp database** [*url*] **Example:** `Device# show ip dhcp database`	(Optional) Displays recent activity on the DHCP database.
Step 6	**show ip dhcp server statistics** [*type-number*] **Example:** `Device# show ip dhcp server statistics`	(Optional) Displays count information about server statistics and messages sent and received.

Configuring Manual Bindings

An address binding is a mapping between the IP address and MAC address of a client. The IP address of a client can be assigned manually by an administrator or assigned automatically from a pool by a DHCP server.

Manual bindings are IP addresses that are manually mapped to MAC addresses of hosts that are found in the DHCP database. Manual bindings are stored in the NVRAM of the DHCP server. Manual bindings are just special address pools. There is no limit to the number of manual bindings, but you can configure only one manual binding per host pool.

Automatic bindings are IP addresses that have been automatically mapped to MAC addresses of hosts that are found in the DHCP database. Because the bindings are stored in the volatile memory of the DHCP server, binding information is lost in the event of power failures or on device reloads. To prevent the loss of automatic binding information, a copy of the automatic binding information is stored on a remote host called the DHCP database agent. The bindings are periodically written to the database agent. When the device reloads, the bindings are read from the database agent to the DHCP database in the DHCP server.

Note We strongly recommend that you use database agents. However, Cisco DHCP server can function even without database agents.

Some DHCP clients send a client identifier (DHCP option 61) in the DHCP packet. To configure manual bindings for such clients, you must enter the **client-identifier** command with the hexadecimal values that identify the DHCP client. To configure manual bindings for clients that do not send a client identifier option, you must enter the **hardware-address** DHCP pool configuration command with the hexadecimal hardware address of the client.

Depending on your release, the DHCP server sends infinite lease time to the clients for which manual bindings are configured.

Depending on your release, the DHCP server sends lease time that is configured using the **lease** command to clients for which manual bindings are configured.

Note You cannot configure manual bindings within the same pool that is configured with the **network** command in DHCP pool configuration mode. See the Configuring DHCP Address Pools section for information about DHCP address pools and the **network** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **host** *address* [*mask* | */prefix-length*]
5. **client-identifier** *unique-identifier*
6. **hardware-address** *hardware-address* [*protocol-type* | *hardware-number*]
7. **client-name** *name*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Device(config)# ip dhcp pool pool1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**host** *address* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# host 172.16.0.1`	Specifies the IP address and subnet mask of the client. • There is no limit to the number of manual bindings you can configure. However, you can configure only one manual binding per host pool.

Step 5	**client-identifier** *unique-identifier*
	Example:
	`Device(dhcp-config)# client-identifier 01b7.0813.8811.66`
		Specifies the unique identifier for DHCP clients.
		• This command is used for DHCP requests.
		• DHCP clients require client identifiers. You can specify the unique identifier for the client in either of the following ways:
		  • A 7-byte dotted hexadecimal notation. For example, 01b7.0813.8811.66, where 01 represents the Ethernet media type and the remaining bytes represent the MAC address of the DHCP client.
		  • A 27-byte dotted hexadecimal notation. For example, 7665e646f722d30302342e3937623032e333734312d466130231. The equivalent ASCII string for this hexadecimal value is vendor-0024.97b0.3741-fa0/1, where vendor represents the vendor, 0024.97b0.3741 represents the MAC address of the source interface, and fa0/1 represents the source interface of the DHCP client.
		• See the Troubleshooting section for information about how to determine the client identifier of the DHCP client.

	Command or Action	Purpose
		Note The identifier specified here is considered for a DHCP client that sends a client identifier in the packet.
Step 6	**hardware-address** *hardware-address* [*protocol-type* | *hardware-number*] **Example:** `Device(dhcp-config)# hardware-address b708.1388.f166 ethernet`	Specifies a hardware address for the client. • This command is used for BOOTP requests. **Note** The hardware address specified here is considered for a DHCP client that does not send a client identifier in the packet.
Step 7	**client-name** *name* **Example:** `Device(dhcp-config)# client-name client1`	(Optional) Specifies the name of the client using any standard ASCII character. • The client name should not include the domain name. For example, the name client1 should not be specified as client1.cisco.com.
Step 8	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

Use the following command to debug any errors that you may encounter when you configure DHCP to automatically generate a unique ID:

- **debug ip dhcp server packets**

Configuring DHCP Static Mapping

The DHCP Static Mapping feature enables the assignment of static IP addresses (without creating numerous host pools with manual bindings) by using a customer-created text file that the DHCP server reads. The benefit of this feature is that it eliminates the need for a long configuration file and reduces the space required in NVRAM to maintain address pools.

A DHCP database contains the mappings between a client IP address and the hardware address, which is referred to as a binding. There are two types of bindings: manual bindings that map a single hardware address to a single IP address, and automatic bindings that dynamically map a hardware address to an IP address from a pool of IP addresses. Manual (also known as static) bindings can be configured individually directly on the device or by using the DHCP Static Mapping feature. These static bindings can be read from a separate static mapping text file. The static mapping text files are read when a device reloads or the DHCP service restarts. These files are read-only.

The read static bindings are treated just like the manual bindings, in that they are:

- Retained across DHCPRELEASEs from the clients.

- Not timed out.
- Deleted only upon deletion of the pool.
- Provided appropriate exclusions for the contained addresses, which are created at the time of the read.

Just like automatic bindings, manual (or static) bindings from the static mapping text file are also displayed by using the **show ip dhcp binding** command.

Perform this task to create the static mapping text file. You will input your addresses in the text file, which is stored in the DHCP database for the DHCP server to read. There is no limit to the number of addresses that can be stored in the file. The file format has the following elements:

- Database version number
- End-of-file designator
- Hardware type
- Hardware address
- IP address
- Lease expiration
- Time the file was created

See the following table for more details about the format of the text file.

The following is a sample static mapping text file:

```
*time* Jan 21 2005 03:52 PM
*version* 2
!IP address    Type    Hardware address     Lease expiration
10.0.0.4 /24   1       0090.bff6.081e       Infinite
10.0.0.5 /28   id      00b7.0813.88f1.66    Infinite
10.0.0.2 /21   1       0090.bff6.081d       Infinite
*end*
```

Table 80: Static Mapping Text File Field Descriptions

Field	Description
time	Specifies the time the file was created. This field allows DHCP to differentiate between the new and old database versions when multiple agents are configured. The valid format of the time is mm dd yyyy hh:mm AM/PM.
version 2	Specifies the database version number.
IP address	Specifies the static IP address. If the subnet mask is not specified, a mask is automatically assigned depending on the IP address. The IP address and the mask is separated by a space.
Type	Specifies the hardware type. For example, type “1” indicates Ethernet. The type “id” indicates that the field is a DHCP client identifier. Legal values can be found online at http://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml in the “Number Hardware Type” list.

Field	Description
Hardware address	Specifies the hardware address. When the type is numeric, the type refers to the hardware media. Legal values can be found online at http://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml in the "Number Hardware Type" list. When the type is "id," the type refers to a match on the client identifier. For more information about the client identifier, see RFC 2132, *DHCP Options and BOOTP Vendor Extensions*, section 9.14, located at https://www.ietf.org/rfc/rfc2132.txt, or the **client-identifier** command. If you are unsure about the client identifier to match with the hardware type, use the **debug dhcp detail** command to display the client identifier being sent to the DHCP server from the client.
Lease expiration	Specifies the expiration of the lease. "Infinite" specifies that the duration of the lease is unlimited.
end	End of file. DHCP uses the *end* designator to detect file truncation.

Configuring the DHCP Server to Read a Static Mapping Text File

Before you begin

The administrator must create the static mapping text file in the correct format and configure the address pools before performing this task.

Before editing the file, you must disable the DHCP server using the **no service dhcp** command.

Note The static bindings must not be deleted when a DHCPRELEASE is received or must not be timed out by the DHCP timer. The static bindings should be created by using the **ip dhcp pool** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **origin file** *url*
5. **end**
6. **show ip dhcp binding** [*address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device(config)# ip dhcp pool pool1`	Assigns a name to a DHCP pool and enters DHCP configuration mode. **Note** If you have already configured the IP DHCP pool name using the **ip dhcp pool** command and the static file URL using the **origin file** command, you must perform a fresh read using the **no service dhcp** command and the **service dhcp** command.
Step 4	**origin file** *url* **Example:** `Device(dhcp-config)# origin file tftp://10.1.0.1/static-bindings`	Specifies the URL that the DHCP server can access to locate the text file.
Step 5	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.
Step 6	**show ip dhcp binding** [*address*] **Example:** `Device# show ip dhcp binding`	(Optional) Displays a list of all bindings created on a specific DHCP server.

Examples

The following sample output from the **show ip dhcp binding** command displays address bindings that are configured:

```
Device# show ip dhcp binding

00:05:14:%SYS-5-CONFIG_I: Configured from console by console
Bindings from all pools not associated with VRF:
IP address  Client-ID/            Ls expir   Type    Hw address           User name
10.9.9.4/8  0063.7363.2d30.3036.  Infinite   Static  302e.3762.2e39.3634. 632d.4574.8892.
10.9.9.1/24 0063.6973.636f.2d30.  Infinite   Static  3036.302e.3437.3165. 2e64.6462.342d.
```

The following sample output displays each entry in the static mapping text file:

```
*time* Jan 21 2005 22:52 PM
!IP address     Type        Hardware address                          Lease expiration
```

```
10.19.9.1 /24    id          0063.6973.636f.2d30.3036.302e.3437
10.9.9.4         id          0063.7363.2d30.3036.302e.3762.2e39.3634.632d  Infinite
*end*
```

The following sample debug output shows the reading of the static mapping text file from the TFTP server:

```
Device# debug ip dhcp server

Loading abc/static_pool from 10.19.192.33 (via Ethernet0):
[OK - 333 bytes]
*May 26 23:14:21.259: DHCPD: contacting agent tftp://10.19.192.33/abc/static_pool (attempt
 0)
*May 26 23:14:21.467: DHCPD: agent tftp://10.19.192.33/abc/static_pool is responding.
*May 26 23:14:21.467: DHCPD: IFS is ready.
*May 26 23:14:21.467: DHCPD: reading bindings from tftp://10.19.192.33/abc/static_pool.
*May 26 23:14:21.707: DHCPD: read 333 / 1024 bytes.
*May 26 23:14:21.707: DHCPD: parsing text line
*time* Apr 22 2002 11:31 AM
*May 26 23:14:21.707: DHCPD: parsing text line “”
*May 26 23:14:21.707: DHCPD: parsing text line
!IP address Type Hardware address Lease expiration
*May 26 23:14:21.707: DHCPD: parsing text line
“10.9.9.1 /24 id 0063.6973.636f.2d30.3036.302e.3437”
*May 26 23:14:21.707: DHCPD: creating binding for 10.9.9.1
*May 26 23:14:21.707: DHCPD: Adding binding to radix tree (10.9.9.1)
*May 26 23:14:21.707: DHCPD: Adding binding to hash tree
*May 26 23:14:21.707: DHCPD: parsing text line
“10.9.9.4 id 0063.7363.2d30.3036.302e.3762.2e39.3634.632d”
*May 26 23:14:21.711: DHCPD: creating binding for 10.9.9.4
*May 26 23:14:21.711: DHCPD: Adding binding to radix tree (10.9.9.4)
*May 26 23:14:21.711: DHCPD: Adding binding to hash tree
*May 26 23:14:21.711: DHCPD: parsing text line “Infinite”
*May 26 23:14:21.711: DHCPD: parsing text line “”
*May 26 23:14:21.711: DHCPD: parsing text line
!IP address Interface-index Lease expiration VRF
*May 26 23:14:21.711: DHCPD: parsing text line “*end*”
*May 26 23:14:21.711: DHCPD: read static bindings from tftp://10.19.192.33/abcemp/static_pool.
```

Customizing DHCP Server Operation

By default, the DHCP server pings a pool address twice before assigning a particular address to a requesting client. If the ping is unanswered, the DHCP server assumes (with a high probability) that the address is not in use and assigns the address to the requesting client.

By default, the DHCP server waits for 2 seconds before timing out a ping packet.

You can configure the DHCP server to ignore and not reply to any BOOTP requests that the server receives. This functionality is beneficial when there is a mix of BOOTP and DHCP clients in a network segment and there is a BOOTP server and a Cisco DHCP server servicing the network segment. The BOOTP server is configured with static bindings for the BOOTP clients and the BOOTP clients must obtain their addresses from the BOOTP server. However, DHCP servers can also respond to BOOTP requests and the DHCP server may offer an address that causes the BOOTP clients to boot with the address from the DHCP server, instead of the address from the BOOTP server. Configuring the DHCP server to ignore BOOTP requests ensures that the BOOTP clients will receive address information from the BOOTP server and will not accept an address from a DHCP server.

Cisco software can forward these ignored BOOTP request packets to another DHCP server if the **ip helper-address** command is configured on the incoming interface.

Note It is not recommended to use DHCP ping checks on Cisco Catalyst switches implemented in switch stack or VSS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp ping packets** *number*
4. **ip dhcp ping timeout** *milliseconds*
5. **ip dhcp bootp ignore**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp ping packets** *number* **Example:** `Device(config)# ip dhcp ping packets 5`	(Optional) Specifies the number of ping packets the DHCP server sends to a pool address before assigning the address to a requesting client. • The default is two packets. Setting the *number* argument to a value of 0 disables the DHCP server ping operation.
Step 4	**ip dhcp ping timeout** *milliseconds* **Example:** `Device(config)# ip dhcp ping timeout 850`	(Optional) Specifies the duration the DHCP server waits for a ping reply from an address pool.
Step 5	**ip dhcp bootp ignore** **Example:** `Device(config)# ip dhcp bootp ignore`	(Optional) Allows the DHCP server to selectively ignore and not reply to received BOOTP requests. • The **ip dhcp bootp ignore** command applies to all DHCP pools configured on the device. BOOTP requests cannot be selectively ignored on a per-DHCP pool basis.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server

The Cisco DHCP server can dynamically configure options such as the Domain Name System (DNS) and Windows Internet Name Service (WINS) addresses to respond to DHCP requests from local clients behind the customer premises equipment (CPE). Earlier, network administrators configured the Cisco DHCP server on each device manually. Now, the Cisco DHCP server is enhanced to allow configuration information to be updated automatically. Network administrators can configure one or more centralized DHCP servers to update specific DHCP options within the DHCP pools. The remote servers can request or "import" these option parameters from centralized servers.

This section contains the following tasks:

Configuring the Central DHCP Server to Update DHCP Options

Perform the following task to configure the Central DHCP Server to update DHCP options:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **network** *network-number* [*mask* | */prefix-length*]
5. **dns-server** *address* [*address2 ... address8*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:**	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.

	Command or Action	Purpose
	`Device(config)# ip dhcp pool 1`	
Step 4	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet number and mask of the DHCP address pool.
Step 5	**dns-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	(Optional) Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. • Servers should be listed in the order of preference.
Step 6	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.

Configuring the Remote Device to Import DHCP Options

Perform the following task to configure the remote device to import DHCP options:

Note When two servers provide DHCP addresses to a single device configured with **ip address dhcp** on two different interfaces, the imported information is merged and, for those options that take a single value, the last known option value will be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **network** *network-number* [*mask* | */prefix-length*]
5. **import** {**all** | **interface** *interface_name*
6. **exit**
7. **interface** *type number*
8. **ip address dhcp**
9. **end**
10. **show ip dhcp import**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Device(config)# ip dhcp pool pool1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 172.30.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.

Step 5	**import** {**all** | **interface** *interface_name*
	Example:
	`Device(dhcp-config)# import all`
	`Device(dhcp-config) # import interface Ethernet0/0`
		Imports DHCP option parameters into the DHCP server database.

Step 6	**exit** **Example:** `Device(dhcp-config)# exit`	Exits DHCP pool configuration mode and enters global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Configures an interface and enters interface configuration mode.
Step 8	**ip address dhcp** **Example:** `Device(config-if)# ip address dhcp`	Specifies that the interface acquires an IP address through DHCP.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 10	**show ip dhcp import** **Example:**	Displays the options that are imported from the central DHCP server.

	Command or Action	Purpose
	`Device# show ip dhcp import`	

Configuring DHCP Address Allocation Using Option 82

Restrictions for DHCP Address Allocation Using Option 82

If the relay agent inserts option 82 but does not set the giaddr field in the DHCP packet, the DHCP server interface must be configured as a trusted interface by using the **ip dhcp relay information trusted** global configuration command. This configuration prevents the server from dropping the DHCP message.

Enabling Option 82 for DHCP Address Allocation

By default, the Cisco DHCP server uses information provided by option 82 to allocate IP addresses. If the DHCP address allocation is disabled, perform the task described in this section to reenable this capability.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp use class**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp use class** **Example:** `Device(config)# ip dhcp use class`	Controls DHCP classes that are used for address allocation. • This functionality is enabled by default. • Use the **no** form of this command to disable this functionality without deleting the DHCP class configuration.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

If DHCP classes are configured in the pool, but the DHCP server does not use the classes, verify if the **no ip dhcp use class** command was configured.

Defining the DHCP Class and Relay Agent Information Patterns

Before you begin

You must know the hexadecimal value of each byte location in option 82 to configure the **relay-information hex** command. The option 82 format may vary from product to product. Contact the relay agent vendor for this information.

Perform this task to define the DHCP class and relay agent information patterns:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp class** *class-name*
4. **relay agent information**
5. **relay-information hex** *pattern* [*] [**bitmask** *mask*]
6. Repeat Steps 3 through 5 for each DHCP class you need to configure.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp class** *class-name* **Example:** `Device(config)# ip dhcp class CLASS1`	Defines a DHCP class and enters DHCP class configuration mode.
Step 4	**relay agent information** **Example:** `Device(dhcp-class)# relay agent information`	Enters relay agent information option configuration mode. • If you omit this step, the DHCP class matches any relay agent information option, whether the relay agent information option value is available or not.

	Command or Action	Purpose
Step 5	**relay-information hex** *pattern* [*] [**bitmask** *mask*] **Example:** `Device(dhcp-class-relayinfo)# relay-information hex 01030a0b0c02050000000123`	(Optional) Specifies a hexadecimal value for full relay information option. • The *pattern* argument creates a pattern that is used to match the DHCP class. • If you omit this step, no pattern is configured and it is considered a match to any relay agent information option value, but the relay information option must be available in the DHCP packet. • You can configure multiple **relay-information hex** commands in a DHCP class.
Step 6	Repeat Steps 3 through 5 for each DHCP class you need to configure.	
Step 7	**end** **Example:** `Device(dhcp-class-relayinfo)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

If DHCP classes are configured in the pool, but the DHCP server does not use the classes, verify if the **no ip dhcp use class** command was configured.

Defining the DHCP Address Pool

Perform this task to define the DHCP address pool:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **network** *network-number* [*mask* | */prefix-length*]
5. **class** *class-name*
6. **address range** *start-ip end-ip*
7. Repeat Steps 5 and 6 for each DHCP class you need to associate with the DHCP pool.
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device# ip dhcp pool ABC`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode. • Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.
Step 4	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 10.0.20.0`	Configures the subnet and mask for a DHCP address pool on a Cisco IOS DHCP server.
Step 5	**class** *class-name* **Example:** `Device(dhcp-config)# class CLASS1`	Associates a class with a pool and enters DHCP pool class configuration mode. • This command also creates a DHCP class if the DHCP class is not yet defined.
Step 6	**address range** *start-ip end-ip* **Example:** `Device(dhcp-pool-class)# address range 10.0.20.1 10.0.20.100`	(Optional) Sets an address range for the DHCP class in a DHCP server address pool. • If this command is not configured for a class, the default value is the entire subnet of the pool. Each class in the DHCP pool is examined for a match in the order configured.
Step 7	Repeat Steps 5 and 6 for each DHCP class you need to associate with the DHCP pool.	
Step 8	**end** **Example:** `Device(dhcp-pool-class)# end`	Returns to privileged EXEC mode.

Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP

Perform this task to configure a static route to use a DHCP default gateway as the next-hop router.

This task enables static routes to be assigned using a DHCP default gateway as the next-hop router. This behavior was not possible before the introduction of this feature because the gateway IP address is not known

until after the DHCP address assignment. A static route could not be configured with the command-line interface (CLI) that used that DHCP-supplied address.

The static routes are installed in the routing table when the default gateway is assigned by the DHCP server. The routes remain in the routing table until the DHCP lease expires at which time the routes are removed.

When a DHCP client releases an address, the corresponding static route (the route configured with the **ip route**command) is automatically removed from the routing table. If the DHCP router option (option 3 of the DHCP packet) changes during the client renewal, the DHCP default gateway changes to the new IP address supplied in the renewal.

This feature is particularly useful for VPN deployments such as Dynamic Multipoint VPNs (DMVPNs). This feature is useful when a non-physical interface like a multipoint generic routing encapsulation (mGRE) tunnel is configured on the router and certain traffic needs to be excluded from going to the tunnel interface.

Before you begin

Verify all DHCP client and server configuration steps. Ensure that the DHCP client and server are properly defined to supply a DHCP router option 3.

Note

- If the DHCP client is not able to obtain an IP address or default router IP address, the static route is not installed in the routing table.
- If the lease has expired and the DHCP client cannot renew the address, the DHCP IP address assigned to the client is released and any associated static routes are removed from the routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **dhcp** [*distance*]
4. **end**
5. **show ip route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **dhcp** [*distance*] **Example:**	Assigns a static route for the default next-hop router when the DHCP server is accessed for an IP address.

	Command or Action	Purpose

	`Device(config)# ip route 209.165.200.225 255.255.255.255 GigabitEthernet 0/0/0 dhcp`
	Example:
	`Device(config)# ip route 209.165.200.226 255.255.255.255 GigabitEthernet 0/0/1 dhcp 20`
		• If more than one interface on a router is configured to obtain an IP address from a DHCP server, use the **ip route** *prefix mask interface-type interface-number* **dhcp** command for each interface. If the interface is not specified, the route is added to the routing table as soon as any of the interfaces obtain an IP address and default router.

Step 4	**end** **Example:** `Device(config)# end`	Returns to global configuration mode.
Step 5	**show ip route** **Example:** `Device# show ip route`	(Optional) Displays the current state of the routing table. • Use this command to display assigned static routes once the DHCP client obtains an address and a default router address from the DHCP server.

Clearing DHCP Server Variables

Perform this task to clear DHCP server variables:

SUMMARY STEPS

1. **enable**
2. **clear ip dhcp binding** {*address* | *****}
3. **clear ip dhcp conflict** {*address* | *****}
4. **clear ip dhcp server statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip dhcp binding** {*address*	*****} **Example:** `Device# clear ip dhcp binding *`	Deletes an automatic address binding from the DHCP database. • Specifying the *address* argument clears the automatic binding for a specific (client) IP address, whereas specifying an asterisk (*****) clears all automatic bindings.
Step 3	**clear ip dhcp conflict** {*address*	*****} **Example:**	Clears an address conflict from the DHCP database.

	Command or Action	Purpose
	`Device# clear ip dhcp conflict 172.16.1.103`	• Specifying the *address* argument clears the conflict for a specific IP address, whereas specifying an asterisk (*****) clears conflicts for all addresses.
Step 4	**clear ip dhcp server statistics** **Example:** `Device# clear ip dhcp server statistics`	Resets all DHCP server counters to 0.

Configuration Examples for the Cisco IOS XE DHCP Server

Example: Configuring the DHCP Database Agent

The following example shows how to store bindings on host 172.16.4.253. The file transfer protocol is FTP. The server waits for 2 minutes (120 seconds) before performing database changes.

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
```

Example: Excluding IP Addresses

In the following example, server A and server B service the subnet 10.0.20.0/24. If the subnet is split equally between the two servers, server A is configured to allocate IP addresses 10.0.20.1 to 10.0.20.125 and server B is configured to allocate IP addresses 10.0.20.126 to 10.0.20.254.

Server A

```
ip dhcp excluded-address 10.0.20.126 10.0.20.255
!
ip dhcp pool A
 network 10.0.20.0 255.255.255.0
```

Server B

```
ip dhcp excluded-address 10.0.20.0 10.0.20.125
!
ip dhcp pool B
 network 10.0.20.0 255.255.255.0
```

Example: Configuring DHCP Address Pools

In the following example, three DHCP address pools are created: one in network 172.16.0.0, one in subnetwork 172.16.1.0, and one in subnetwork 172.16.2.0. Attributes from network 172.16.0.0—such as the domain name, Domain Name System (DNS) server, (Network Basic Input/Output System) NetBIOS name server, and NetBIOS node type—are inherited in subnetworks 172.16.1.0 and 172.16.2.0. In each pool, clients are granted 30-day leases and all addresses in each subnetwork, except the excluded addresses, are available to the DHCP

server for assigning to clients. The table below lists the IP addresses for the devices in three DHCP address pools.

Table 81: DHCP Address Pool Configuration

Pool 0 (Network 172.16.0.0)	**Pool 1 (Subnetwork 172.16.1.0)**	**Pool 2 (Subnetwork 172.16.2.0)**			
Device	IP Address	Device	IP Address	Device	IP Address
Default devices	—	Default devices	172.16.1.100 172.16.1.101	Default devices	172.16.2.100 172.16.2.101
DNS server	172.16.1.102 172.16.2.102	—	—	—	—
NetBIOS name server	172.16.1.103 172.16.2.103	—	—	—	—
NetBIOS node type	h-node	—	—	—	—

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
ip dhcp excluded-address 172.16.1.100 172.16.1.103
ip dhcp excluded-address 172.16.2.100 172.16.2.103
!
ip dhcp pool 0
 network 172.16.0.0 /16
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
!
ip dhcp pool 1
 network 172.16.1.0 /24
 default-router 172.16.1.100 172.16.1.101
 lease 30
!
ip dhcp pool 2
 network 172.16.2.0 /24
 default-router 172.16.2.100 172.16.2.101
 lease 30
```

The following example shows how to configure DHCP pool to support RegEx feature:

```
!
ip dhcp pool test
network 192.168.10.0 255.255.255.0
default-router 192.168.10.1
 dns-server 40.0.0.100
 class cisco_devices
  address range 192.168.10.2 192.168.10.100
!
class smart_phones
  address range 192.168.10.101 192.168.10.220
!
!
ip dhcp class cisco_devices
 option 60 cisco_string ----------------------<this is option 60 VCI string, exact match>
```

```
!
ip dhcp class smart_phones
 option 60 smartphone* ----------------------<option 60 VCI string, regex match>
!
```

The following example shows how to configure DHCP server class:

```
Router#
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip dhcp class HATHWAY_STB
Router(config-dhcp-class)#?
DHCP class configuration commands:
  exit        Exit from DHCP class configuration mode
  no         Negate a command or set its defaults
 option   Raw DHCP options
  relay    Enter relay agent information option configuration submode
  remark  Specify a remark for this class

Router(config-dhcp-class)#option ?
  <0-254>  DHCP option code

Router(config-dhcp-class)#option 60 ?
  hex  Specify hex value of the option
  WORD  Specify a regular expression string

Router(config-dhcp-class)#option 60 stb*  ?
<cr>
```

The following example shows how to Import options learnt on specific interface to LAN side DHCP pool:

```
!
ip dhcp pool LAN_Pool
import interface Ethernet0/0
!

Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# ip dhcp pool pc_pool
Router(dhcp-config)# import ?
  all              all DHCP options
  interface   Select an interface to import options
Router(dhcp-config)# import interface Ethernet0/1
```

Example: Configuring a DHCP Address Pool with Multiple Disjoint Subnets

Multiple disjoint subnets in a DHCP pool can be used in any of the following network topologies:

- IP address pooling—The DHCP client and server reside on the same subnet.
- DHCP relay—The DHCP client and DHCP server communicate through a DHCP relay agent where the relay interface is configured with secondary IP addresses.
- Hierarchical DHCP—The DHCP server is configured as the DHCP subnet allocation server. The DHCP client and DHCP subnet allocation server communicate through an on-demand address pool (ODAP) router.

In the following example, one DHCP address pool named pool3 is created; the primary subnet is 172.16.0.0/16, one secondary subnet is 172.16.1.0/24, and the other secondary subnet is 172.16.2.0/24.

- When IP addresses in the primary subnet are exhausted, the DHCP server inspects the secondary subnets in the order in which the subnets were added to the pool.
- When the DHCP server allocates an IP address from the secondary subnet 172.16.1.0/24, the server uses the subnet-specific default device list that consists of IP addresses 172.16.1.100 and 172.16.1.101. However, when the DHCP server allocates an IP address from the subnet 172.16.2.0/24, the server uses the pool-wide list that consists of the four IP addresses from 172.16.0.100 to 172.16.0.103.
- Other attributes from the primary subnet 172.16.0.0/16—such as the domain name, DNS server, NetBIOS name server, and NetBIOS node type—are inherited in both the secondary subnets.
- DHCP clients are granted 30-day leases on IP addresses in the pool. All addresses in each subnet, except the excluded addresses, are available to the DHCP server for assigning to clients.

The table below lists the IP addresses for the devices in the DHCP address pool that consists of three disjoint subnets.

Table 82: DHCP Address Pool Configuration with Multiple Disjoint Subnets

Primary Subnet (172.16.0.0/16)	**First Secondary Subnet (172.16.1.0/24)**	**Second Secondary Subnet (172.16.2.0/24)**			
Device	IP Address	Device	IP Address	Device	IP Address
Default devices	172.16.0.100 172.16.0.101 172.16.0.102 172.16.0.103	Default devices	172.16.1.100 172.16.1.101	Default devices	172.16.0.100 172.16.0.101 172.16.0.102 172.16.0.103
DNS server	172.16.1.102 172.16.2.102	—	—	—	—
NetBIOS name server	172.16.1.103 172.16.2.103	—	—	—	—
NetBIOS node type	h-node	—	—	—	—

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
ip dhcp excluded-address 172.16.0.100 172.16.1.103
ip dhcp excluded-address 172.16.1.100 172.16.1.101
!
ip dhcp pool pool3
 network 172.16.0.0 /16
 default-router 172.16.0.100 172.16.2.101 172.16.0.102 172.16.0.103
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
 lease 30
!
 network 172.16.1.0 /24 secondary
  override default-router 172.16.1.100 172.16.1.101
  end
```

```
!
 network 172.16.2.0 /24 secondary
```

Configuring Manual Bindings Example

The following example shows how to create a manual binding for a client named Mars.cisco.com. The MAC address of the client is 02c7.f800.0422 and the IP address of the client is 172.16.2.254.

```
ip dhcp pool Mars
 host 172.16.2.254
 hardware-address 02c7.f800.0422 ieee802
 client-name Mars
```

Because attributes are inherited, the previous configuration is equivalent to the following:

```
ip dhcp pool Mars
 host 172.16.2.254 mask 255.255.255.0
 hardware-address 02c7.f800.0422 ieee802
 client-name Mars
 default-router 172.16.2.100 172.16.2.101
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
```

Example: Configuring Static Mapping

The following example shows how to restart the DHCP server, configure the pool, and specify the URL where the static mapping text file is stored:

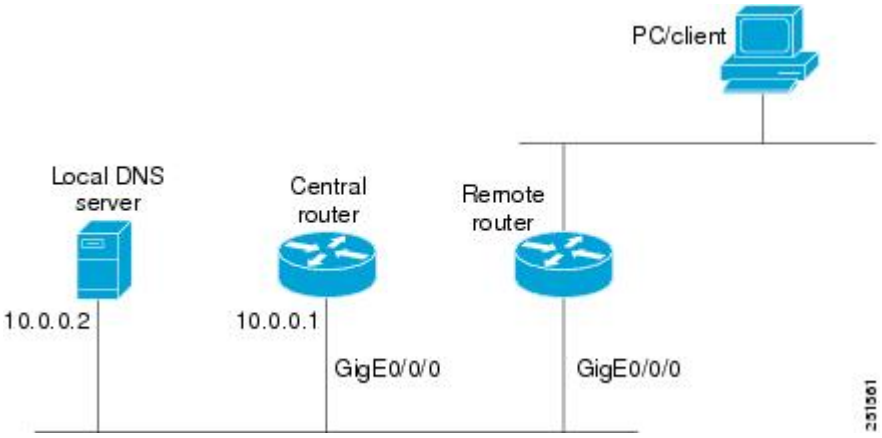
```
no service dhcp
service dhcp
ip dhcp pool abcpool
origin file tftp://10.1.0.1/staticfilename
```

Note The static mapping text file can be copied to flash memory on the device and served by the TFTP process of the device. In this case, the IP address in the original file line must be an address owned by the device and one additional line of configuration is required on the device:**tftp-server flash** *static-filename*.

Importing DHCP Options Example

The following example shows a remote and central server configured to support the importing of DHCP options. The central server is configured to automatically update DHCP options, such as DNS and WINs addresses, within the DHCP pools. In response to a DHCP request from a local client behind CPE equipment, the remote server can request or “import” these option parameters from the centralized server. See the figure below for a diagram of the network topology.

Figure 60: DHCP Example Network Topology



Central Router

```
!do not assign this range to DHCP clients
ip dhcp-excluded address 10.0.0.1 10.0.0.5
!
ip dhcp pool central
! Specifies network number and mask for DHCP clients
 network 10.0.0.0 255.255.255.0
! Specifies the domain name for the client
 domain-name central
! Specifies DNS server that will respond to DHCP clients when they need to correlate host
! name to ip address
 dns-server 10.0.0.2
!Specifies the NETBIOS WINS server
 netbios-name-server 10.0.0.2
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.0
 duplex auto
 speed auto
```

Remote Router

```
ip dhcp pool client
! Imports DHCP option parameters into DHCP server database
 import all
 network 20.0.0.0 255.255.255.0
!
interface GigabitEthernet0/0/0
 ip address dhcp
 duplex auto
 speed auto
```

Configuring DHCP Address Allocation Using Option 82 Example

This example configures two DHCP classes. CLASS1 defines the group of DHCP clients whose address requests contain the relay agent information option with the specified hexadecimal values. CLASS2 defines the group of DHCP clients whose address requests contain the configured relay agent information suboptions.

CLASS3 has no pattern configured and is treated as a “match to any” class. This type of class is useful for specifying a “default” class.

In the following example, the subnet of pool ABC has been divided into three ranges without further subnetting of the 10.0.20.0/24 subnet. If there is a DHCP Discover message from the 10.0.20.0/24 subnet with option 82 matching that of class CLASS1, an available address in the range from 10.0.20.1 to 10.0.20.100 will be allocated. If there is no free address in CLASS1's address range, the DHCP Discover message will be matched against CLASS2, and so on.

Thus, each class in the DHCP pool will be examined for a match in the order configured by the user. In pool ABC, the order of matching is CLASS1, CLASS2, and finally CLASS3. In pool DEF, class CLASS2 does not have any address range configured. By default, the address range for a particular class is the pool's entire subnet(s). Therefore, clients matching CLASS2 may be allocated addresses from 11.0.20.1 to 11.0.20.254.

Multiple pools can be configured with the same class, eliminating the need to configure the same patterns in multiple pools. In the future, further classification method may be implemented. For example, there may be a need to specify that one or more pools should only be used to service a particular class of devices (for example, cable modems and IP phones).

```
! Defines the DHCP classes and relay information patterns
ip dhcp class CLASS1
 relay agent information
  relay-information hex 01030a0b0c02050000000123
  relay-information hex 01030a0b0c02*
  relay-information hex 01030a0b0c02050000000000 bitmask 000000000000000000000FF
ip dhcp class CLASS2
 relay agent information
  relay-information hex 01040102030402020102
  relay-information hex 01040101030402020102
ip dhcp class CLASS3
 relay agent information
! Associates the DHCP pool with DHCP classes
ip dhcp pool ABC
 network 10.0.20.0 255.255.255.0
 class CLASS1
  address range 10.0.20.1 10.0.20.100
class CLASS2
  address range 10.0.20.101 10.0.20.200
 class CLASS3
  address range 10.0.20.201 10.0.20.254
ip dhcp pool DEF
 network 11.0.20.0 255.255.255.0
 class CLASS1
  address range 11.0.20.1 11.0.20.64
 class CLASS2
```

Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP Example

The following example shows how to configure two GigabitEthernet interfaces to obtain the next-hop router IP address from the DHCP server:

```
ip route 10.10.10.0 255.255.255.0 dhcp 200
ip route 10.10.20.1 255.255.255.255 gigaether 1 dhcp
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
DHCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
DHCP conceptual information	"DHCP Overview" module
DHCP relay agent configuration	"Configuring the Cisco IOS XE DHCP Relay Agent" module
DHCP client configuration	"Configuring the Cisco IOS XE DHCP Client" module
DHCP On-Demand Address Pool Manager	"Configuring the DHCP On-Demand Address Pool Manager" module

Standards and RFCs

Standard/RFC	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2132	*DHCP Options and BOOTP Vendor Extensions*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for the Cisco IOS XE DHCP Server

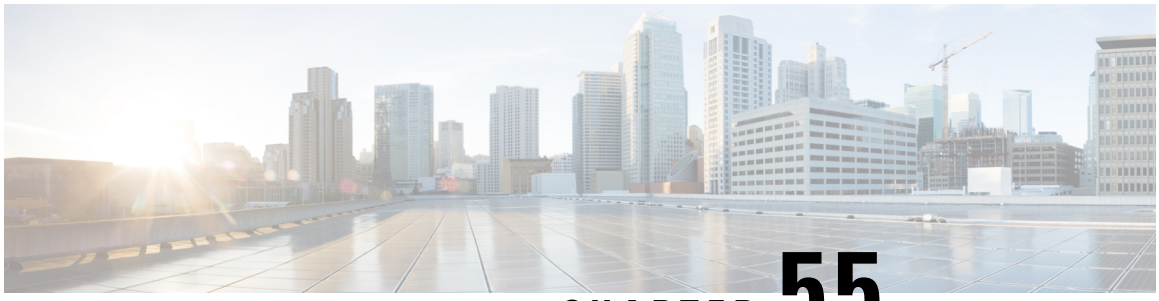
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 83: Feature Information for the Cisco IOS XE DHCP Server

Feature Name	Releases	Feature Configuration Information
DHCP Server	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	Cisco routers running Cisco IOS XE software include Dynamic Host Configuration Protocol (DHCP) server and relay agent software. The Cisco IOS XE DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the domain name system (DNS) server and the default router.
DHCP Address Allocation Using Option 82	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	The Cisco IOS XE DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) information sent by the relay agent. The following commands were introduced by this feature: **address range**, **class**, **ip dhcp class**, **ip dhcp use class**, **relay agent information**, **relay-information hex**.
DHCP Statically Configured Routes Using a DHCP Gateway	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	This feature enables the configuration of static routes that point to an assigned DHCP next hop router. The following commands were modified by this feature: **ip route**, **show ip route**.
DHCP Server Options - Import and Autoconfiguration	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	Options imported by multiple subsystems can co-exist in the DHCP address pool. When the session is terminated or the lease is released, the imported options are cleared.
DHCP Server Multiple Subnet	12.4(15)T 12.2(33)SRB 15.3(1)S Cisco IOS XE Release 3.2SE Cisco IOS XE Release 3.9S	The DHCP Server Multiple Subnet feature enables multiple subnets to be configured under the same DHCP address pool. The following commands were introduced or modified: **network**(DHCP), **override default-router**.

Feature Name	Releases	Feature Configuration Information
DHCP Static Mapping	Cisco IOS XE Release 3.9S	Configuring static mapping pools enables the DHCP server to read the static bindings from a separate text file (similar in format to the DHCP database file) that is stored in special pools. The following commands were introduced or modified: **origin**.
DHCP Server Import All Enhancement	Cisco IOS XE Release 3.9S	The DHCP Server Import All Enhancement feature is an enhancement to the **import all** command. Prior to this feature, the options imported through the **import all** command were overwritten by those imported by another subsystem. Through this feature, options imported by multiple subsystems can coexist in the DHCP address pool. When the session is terminated or the lease is released, the imported options are cleared.
DHCPv4 Client options	Cisco IOS XE Fuji Release 16.9.1	The following features are supported on Cisco 4000 Series ISRs: • Regular Expression support for options 60, 77, 124 and 125 • Generic support to configure all applicable client DHCP options • Import options learnt on specific interface to DHCP pool • Longest Match support for option 60, 77, 124 and 125



CHAPTER 55

Configuring the DHCP Server On-Demand Address Pool Manager

The Cisco IOS XE DHCP server on-demand address pool (ODAP) manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses. When a Cisco IOS XE router is configured as an ODAP manager, pools of IP addresses are dynamically increased or reduced in size depending on the address utilization level. A DHCP pool configured in the router can also be used as an IP address pooling mechanism. The IP address pooling mechanism is configured in the router to specify the source of IP addresses for PPP peers.

Prerequisites for Configuring the DHCP Server On-Demand Address Pool Manager

Before you configure the ODAP manager, you should understand the concepts documented in the “DHCP Overview” module.

You must configure standard Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) unless you intend to use non-MPLS VPNs.

In order for the IP address pooling mechanism to work correctly, the VPN routing and forwarding instance (VRF) of the PPP session must match that configured on the pool. Typically this matching is done either by configuring the **ip vrf forwarding** *vrf-name* command on the virtual template interface, or if AAA is used to authorize the PPP user, it can be part of the user’s profile configuration.

Note For a default session, you can apply access interface VRF and VRF service simultaneously.

Restrictions for Configuring the DHCP Server On-Demand Address Pool Manager

- From Cisco IOS Release 17.6.1, you can use the **ip dhcp excluded-address** *vrf-name* global configuration command to exclude addresses from the VRF associated pools.
- The **vrf** DHCP pool configuration command is currently not supported for host pools.
- Attribute inheritance is not supported on VRF pools.
- A router can be configured as a subnet allocation server and a DHCP server at the same time with one restriction: separate pools must be created for subnet allocation and IP address assignment. An address pool cannot be used by DHCP for both subnet allocation and IP address assignment.

Information About the DHCP Server On-Demand Address Pool Manager

ODAP Manager Operation

ODAPs enable pools of IP addresses to be dynamically increased or reduced in size depending on the address utilization level. Once configured, the ODAP is populated with one or more subnets leased from a source server and is ready to serve address requests from DHCP clients or from PPP sessions. The source server can be a remote DHCP server or a RADIUS server (via AAA). Currently, only the Cisco Access Registrar RADIUS server supports ODAPs. Subnets can be added to the pool when a certain utilization level (high utilization mark) is achieved. When the utilization level falls below a certain level (low utilization mark), a subnet can be returned to the server from which it was originally leased. Summarized routes for each leased subnet must be inserted or removed from the related VRF with each addition or removal of subnets into the ODAP.

ODAPs support address assignment using DHCP for customers using private addresses such as in MPLS VPNs. VPNs allow the possibility that two pools in separate networks can have the same address space, with private network addresses, served by the same DHCP server. These IP addresses can be distinguished by a VPN identifier to help select the VPN to which the client belongs.

Each ODAP is configured and associated with a particular MPLS VPN. Cisco IOS XE software also supports non-MPLS VPN address pools by adding pool name support to the **peer default ip address dhcp-pool** *pool-name*command.

For MPLS VPNs, each VPN is associated with one or more VRFs. The VRF is a key element in the VPN technology because it maintains the routing information that defines a customer VPN site. This customer site is attached to a provider edge (PE) router. A VRF consists of an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

A PPP session belonging to a specific VPN is only allocated an address from the ODAP associated with that VPN. These PPP sessions are terminated on a Virtual Home Gateway (VHG)/PE router where the ODAP is configured. The VHG/PE router maps the remote user to the corresponding MPLS VPNs.

For PPP sessions, individual address allocation from an ODAP follows a First Leased subnet First (FLF) policy. FLF searches for a free address beginning on the first leased subnet, followed by a search on the second leased subnet if no free address is available in the first subnet, and so on. This policy provides the benefit of grouping the leased addresses over time to a set of subnets, which allows an efficient subnet release and route summarization.

However, the FLF policy differs from the normal DHCP address selection policy. Normal DHCP address selection takes into account the IP address of the receiving interface or the gateway address if it is nonzero. To support both policies, the DHCP server needs to be able to distinguish between a normal DHCP address request and an address request for a PPP client. The ODAP manager uses an IP address pooling mechanism for PPP that allows the DHCP server to distinguish between a normal DHCP address request and a request from a PPP client.

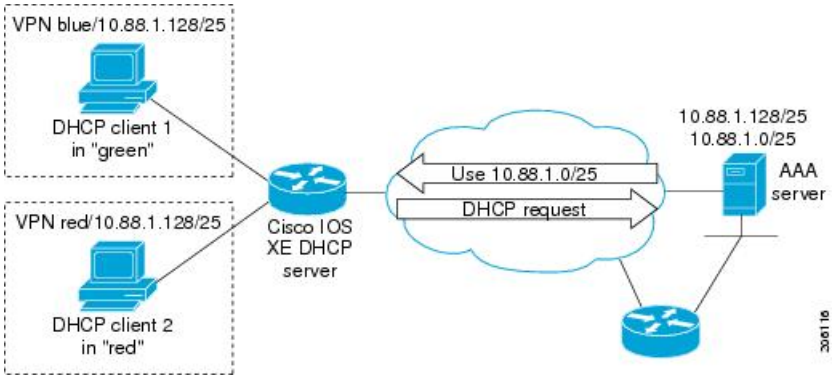
Subnet release from an ODAP follows a Last Leased subnet First (LLF) policy, which prefers the last leased subnet to be released first. This LLF policy searches for a releasable subnet (a subnet with no addresses currently being leased) starting with the last leased subnet. If a releasable subnet is found (candidate subnet), it is released, and the summarized route for that subnet is removed. If more than one releasable subnet exists at that time, only the most recently allocated is released. If there are no releasable subnets, no action is taken. If by releasing the candidate subnet, the high utilization mark is reached, the subnet is not released. The first leased subnet is never released (regardless of the instantaneous utilization level) until the ODAP is disabled.

When a DHCP pool receives multiple subnets from an upstream DHCP server, an address from each subnet is automatically configured on the client connected interface so that the addresses within the subnets can be requested by DHCP clients.

The first address in the first subnet is automatically assigned to the primary address on the interface. The first address of each subsequent subnet is assigned to secondary addresses on the interface. In addition, as client addresses are reclaimed, the count of lease addresses for that subnet is decremented. Once a lease counter for a subnet reaches zero (that is, lease expiry), the subnet is returned to the pool. The previous address on the interface is removed and the first secondary address on the interface is promoted as the primary address of the interface.

The figure below shows an ODAP manager configured on the Cisco IOS XE DHCP server. The ODAP requests an initial pool from the AAA server. Clients make DHCP requests and the DHCP server fulfills requests from the pool. When the utilization rate meets 90 percent, the ODAP manager requests an expansion and the AAA server allocates another subnet from which the ODAP manager can allocate addresses.

Figure 61: ODAP Address Pool Management for MPLS VPNs



Subnet Allocation Server Operation

You can also configure the ODAP manager to allocate subnets instead of individual IP addresses.

This capability allows the network operator to configure a Cisco IOS XE router as a subnet allocation server. The operation of a subnet allocation server is similar to the operation of a DHCP server, except that pools of subnets are created and assigned instead of pools of IP addresses. Subnet allocation pools are created and configured by using the **subnet prefix-length** command in DHCP pool configuration mode. The size of each assigned or allocated subnet is set by the *prefix-length* argument, using standard Common InterDomain Routing (CIDR) bit count notation to determine the number of addresses that are configured in each subnet lease.

When a DHCP server is configured as a subnet allocation server, it provides subnet allocation pools for ODAP manager allocation. In the figure below, Router B is the subnet allocation server and allocates subnets to the ODAP manager based on the demand for IP addresses and subnet availability. Router B is configured to allocate an initial amount of address space in the form of subnets to the ODAP manager. The size of the subnet allocated by the ODAP manager is determined by the subnet size that is configured on the subnet allocation server. The ODAP manager will then assign addresses to clients from these subnets and allocate more subnets as the need for address space increases.

Figure 62: Subnet Allocation Server Topology

When the ODAP manager allocates a subnet, the subnet allocation server creates a subnet binding. This binding is stored in the DHCP database for as long as the ODAP manager requires the address space. The binding is removed and the subnet is returned to the subnet pool only when the ODAP manager releases the subnet as address space utilization decreases.

The subnet allocation server can also be associated with a VRF. A VRF consists of an IP routing table, a derived CEF table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

Benefits of Using ODAPs

Efficient Address Management

The ODAP manager allows customers to optimize their use of IP addresses, thus conserving address space.

Efficient Route Summarization and Update

The ODAP manager inserts a summarized route when a subnet is added to the ODAP.

Multiple VRF and Independent Private Addressing Support

The ODAP manager automatically injects subnet routing information into the appropriate VRF.

How to Configure the DHCP Server On-Demand Address Pool Manager

Defining DHCP ODAPs as the Global Default Mechanism

Perform this task to specify that the global default mechanism to use is on-demand address pooling.

IP addressing allows configuration of a global default address pooling mechanism. The DHCP server needs to be able to distinguish between a normal DHCP address request and an address request for a PPP client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip address-pool dhcp-pool**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip address-pool dhcp-pool** **Example:** `Router(config)# ip address-pool dhcp-pool`	Enables on-demand address pooling as the global default IP address mechanism. • For remote access (PPP) sessions into MPLS VPNs, IP addresses are obtained from locally configured VRF-associated DHCP pools.

Defining DHCP ODAPs on an Interface

Perform this task to configure on-demand address pools on an interface.

The interface on-demand address pooling configuration overrides the global default mechanism on that interface.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number*
4. **peer default ip address dhcp-pool** [*pool-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Virtual-Template1`	Specifies the interface and enters interface configuration mode.
Step 4	**peer default ip address dhcp-pool** [*pool-name*] **Example:** `Router(config-if)# peer default ip address dhcp-pool mypool`	Specifies an IP address from an on-demand address pool to be returned to a remote peer connecting to this interface. • The *pool-name* argument supports non-MPLS VPNs and is mandatory if the session is not associated with any VRF. Multiple pool names can be accepted but must be separated by space.

Configuring the DHCP Pool as an ODAP

Perform this task to configure a DHCP address pool as an ODAP pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **vrf** *name*
5. **origin** {**dhcp** | **aaa**| **ipcp**} [**subnet size initial** *size* [**autogrow** *size*]]
6. **utilization mark low** *percentage-number*
7. **utilization mark high** *percentage-number*
8. **end**
9. **show ip dhcp pool** [*pool-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool red-pool`	Configures a DHCP address pool on a Cisco IOS XE DHCP server and enters DHCP pool configuration mode.
Step 4	**vrf** *name* **Example:** `Router(dhcp-config)# vrf red`	(Optional) Associates the address pool with a VRF name. • Only use this command for MPLS VPNs.
Step 5	**origin** {**dhcp** | **aaa**| **ipcp**} [**subnet size initial** *size* [**autogrow** *size*]] **Example:** `Router(dhcp-config)# origin dhcp subnet size initial /16 autogrow /16`	Configures an address pool as an on-demand address pool. • If you do not configure the pool as an autogrow pool, the pool will not request additional subnets if one subnet is already in the pool. • You can enter size as either the subnet mask (nnnn.nnnn.nnnn.nnnn) or prefix size (/nn). The valid values are /0 and /4 to /30. • When a DHCP pool receives multiple subnets from an upstream DHCP server, an address from each subnet is automatically configured on the client connected interface so that the addresses within the subnets can be requested by DHCP clients. The first address in the first subnet is automatically assigned to the primary address on the interface. The first address of each subsequent subnet is assigned to secondary addresses on the interface. In addition, as client addresses are reclaimed, the count of lease addresses for that subnet is decremented. Once a lease counter for a subnet reaches zero (that is, lease expiry), the subnet is returned to the pool. The previous address on the interface is removed and the first secondary address on the interface is promoted as the primary address of the interface. • If the **origin aaa** option is configured, AAA must be configured.

	Command or Action	Purpose
Step 6	**utilization mark low** *percentage-number* **Example:** `Router(dhcp-config)# utilization mark low 40`	Sets the low utilization mark of the pool size. • This command cannot be used unless the **autogrow** *size* option of the **origin** command is configured. • The default value is 0 percent.
Step 7	**utilization mark high** *percentage-number* **Example:** `Router(dhcp-config)# utilization mark high 60`	Sets the high utilization mark of the pool size. • This command cannot be used unless the **autogrow** *size* option of the **origin** command is configured. • The default value is 100 percent.
Step 8	**end** **Example:** `Router(dhcp-config)# end`	Returns to global configuration mode.
Step 9	**show ip dhcp pool** [*pool-name*] **Example:** `Router# show ip dhcp pool`	(Optional) Displays information about DHCP address pools. • Information about the primary and secondary interface address assignment is also displayed.

Configuring ODAPs to Obtain Subnets Through IPCP Negotiation

Perform this task to configure your router to use subnets obtained through IP Control Protocol (IPCP) negotiation.

You can assign IP address pools to customer premises equipment (CPE) devices, which, in turn, assign IP addresses to the CPE and to a DHCP pool. This functionality has three requirements:

- The Cisco IOS XE CPE device must be able to request and use the subnet.
- The RADIUS server (via AAA) must be able to provide that subnet and insert the framed route into the proper VRF table.
- The PE router must be able to facilitate providing the subnet through (IPCP) negotiation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **import** {**all** | **interface** *interface_name*
5. **origin ipcp**
6. **exit**
7. **interface** *type number*
8. **ip address pool** *pool-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool red-pool`	Configures a DHCP address pool on a Cisco IOS XE DHCP server and enters DHCP pool configuration mode.

Step 4	**import** {**all** | **interface** *interface_name*
	Example:
	`Device(dhcp-config)# import all`
	`Device(dhcp-config) # import interface Ethernet0/0`
		Imports DHCP option parameters into the DHCP server database.

Step 5	**origin ipcp** **Example:** `Router(dhcp-config)# origin ipcp`	Configures an address pool as an on-demand address pool using IPCP as the subnet allocation protocol.
Step 6	**exit** **Example:** `Router(dhcp-config)# exit`	Exits DHCP pool configuration mode.
Step 7	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies the interface and enters interface configuration mode.
Step 8	**ip address pool** *pool-name* **Example:** `Router(config-if)# ip address pool red-pool`	Specifies that the interface IP address will be automatically configured from the named pool, when the pool is populated with a subnet from IPCP.

Configuring AAA

Perform this task to configure AAA.

To allow ODAP to obtain subnets from the AAA server, the AAA client must be configured on the VHG/PE router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authorization configuration default group radius**
5. Do one of the following:
   - **aaa accounting network default start-stop group radius**
   - or
   - **aaa accounting network default stop-only group radius**
6. **aaa session-id common**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA access control.
Step 4	**aaa authorization configuration default group radius** **Example:** `Router(config)# aaa authorization configuration default group radius`	Downloads static route configuration information from the AAA server using RADIUS.
Step 5	Do one of the following: • **aaa accounting network default start-stop group radius** • or • **aaa accounting network default stop-only group radius** **Example:**	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. Sends a "start" accounting notice at the beginning of a process. or Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. Sends a "stop" accounting notice at the end of the requested user process.

	Command or Action	Purpose

	`Router(config)# aaa accounting network default start-stop group radius`
	Example:
	Example:
	`Router(config)# aaa accounting network default stop-only group radius`	

Step 6	**aaa session-id common** **Example:** `Router(config)# aaa session-id common`	Ensures that the same session ID will be used for each AAA accounting service type within a call.

Configuring RADIUS

ODAP AAA Profile

The AAA server sends the RADIUS Cisco AV pair attributes “pool-addr” and “pool-mask” to the Cisco IOS XE DHCP server in the access request and access accept. The pool-addr attribute is the IP address and the pool-mask attribute is the network mask (for example, pool-addr=192.168.1.0 and pool-mask=255.255.0.0). Together, these attributes make up a network address (address/mask) that is allocated by the AAA server to the Cisco IOS XE DHCP server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface** *subinterface-name*
4. **radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number*
5. **radius server attribute 32 include-in-access-req**
6. **radius server attribute 44 include-in-access-req**
7. **radius-server vsa send accounting**
8. **radius-server vsa send authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**ip radius source-interface** *subinterface-name*
	Example:
	`Router(config)#`
	`ip radius source-interface GigabitEthernet0/0/0`
		Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets.

Step 4	**radius-server host** *ip-address* **auth-port** *port-number* **acct-port** *port-number*
	Example:
	`Router(config)#`
	`radius-server host 172.16.1.1 auth-port 1645 acct-port 1646`
		Specifies a RADIUS server host.
		• The *ip-address* argument specifies the IP address of the RADIUS server host.

Step 5	**radius server attribute 32 include-in-access-req**
	Example:
	`Router(config)#`
	`radius server attribute 32 include-in-access-req`
		Sends RADIUS attribute 32 (NAS-Identifier) in an access request or accounting request.

Step 6	**radius server attribute 44 include-in-access-req**
	Example:
	`Router(config)#`
	`radius server attribute 44 include-in-access-req`
		Sends RADIUS attribute 44 (Accounting Session ID) in an access request or accounting request.

Step 7	**radius-server vsa send accounting**
	Example:
	`Router(config)#`
	`radius-server vsa send accounting`
		Configures the network access server (NAS) to recognize and use vendor-specific accounting attributes.

Step 8	**radius-server vsa send authentication**
	Example:
	`Router(config)#`
	`radius-server vsa send authentication`
		Configures the NAS to recognize and use vendor-specific authentication attributes.

What to do next

Disabling ODAPs

This task shows how to disable an ODAP from a DHCP pool.

When an ODAP is disabled, all leased subnets are released. If active PPP sessions are using addresses from the released subnets, those sessions will be reset. DHCP clients leasing addresses from the released subnets will not be able to renew their leases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **no origin** {**dhcp**| **aaa**| **ipcp**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool red-pool`	Configures a DHCP address pool on a Cisco IOS XE DHCP server and enters DHCP pool configuration mode.
Step 4	**no origin** {**dhcp**| **aaa**| **ipcp**} **Example:** `Router(dhcp-config)# no origin dhcp`	Disables the ODAP.

Verifying ODAP Operation

Perform this task to verify ODAP operation.

SUMMARY STEPS

1. **enable**

2. **show ip dhcp pool** [*pool-name*] The following output is for two DHCP pools: Green and Global. Pool Green is configured with a high utilization mark of 50 and a low utilization mark of 30. The pool is also configured to obtain more subnets when the high utilization mark is reached (autogrow). The Subnet size field indicates the values configured in the **origin** command as the initial and incremental subnet sizes that would be requested by the pool named Green. The Total addresses field is a count of all the usable addresses in the pool. The Leased addresses field is a total count of how many bindings were created from the pool. The Pending event field shows subnet request, which means that a subnet request is pending for the pool. The subnet request was scheduled because the Leased addresses count has exceeded the high utilization level of the pool. Subnets currently added to pool Green are shown in sequence. The Current index column shows the address that would be allocated next from this subnet. The IP address range column shows the range of usable addresses from the subnet. The Leased addresses column shows individual count of bindings created from each subnet. Three subnets are currently added to pool Green. The first two subnets have used all their addresses and thus the Current index is showing 0.0.0.0. Notice that pool Green and pool Global can have the same subnet (172.16.0.1-172.16.0.6) because pool Green is configured to be in VRF Green, while pool Global is configured to be in the global address space.
3. **show ip dhcp binding** The following output shows the bindings from pool Green. The Type field shows On-demand, which indicates that the address binding was created for a PPP session. The Lease expiration field shows Infinite, which means that the binding is valid as long as the session is up. If a subnet must be released back to the leasing server while the session is still up, the session is reset so that it will be forced to obtain a new IP address. The Hardware address column for an On-demand entry shows the identifier for the session as detected by PPP. There are no bindings shown under the Bindings from all pools not associated with VRF field because the Global pool has not allocated any addresses.

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 show ip dhcp pool [*pool-name*]

The following output is for two DHCP pools: Green and Global. Pool Green is configured with a high utilization mark of 50 and a low utilization mark of 30. The pool is also configured to obtain more subnets when the high utilization mark is reached (autogrow). The Subnet size field indicates the values configured in the **origin** command as the initial and incremental subnet sizes that would be requested by the pool named Green. The Total addresses field is a count of all the usable addresses in the pool. The Leased addresses field is a total count of how many bindings were created from the pool. The Pending event field shows subnet request, which means that a subnet request is pending for the pool. The subnet request was scheduled because the Leased addresses count has exceeded the high utilization level of the pool. Subnets currently added to pool Green are shown in sequence. The Current index column shows the address that would be allocated next from this subnet. The IP address range column shows the range of usable addresses from the subnet. The Leased addresses column shows individual count of bindings created from each subnet. Three subnets are currently added to pool Green. The first two subnets have used all their addresses and thus the Current index is showing 0.0.0.0. Notice that pool Green and pool Global can have the same subnet (172.16.0.1-172.16.0.6) because pool Green is configured to be in VRF Green, while pool Global is configured to be in the global address space.

Example:

```
Router# show ip dhcp pool
Pool Green :
 Utilization mark (high/low)    : 50 / 30
```

```
 Subnet size (first/next)       : 24 / 24 (autogrow)
 VRF name                       : Green
 Total addresses                : 18
 Leased addresses               : 13
 Pending event                  : subnet request
 3 subnets are currently in the pool :
 Current index        IP address range                    Leased addresses
 0.0.0.0              172.16.0.1       - 172.16.0.6       6
 0.0.0.0              172.16.0.9       - 172.16.0.14      6
 172.16.0.18          172.16.0.17      - 172.16.0.22      1
Pool Global :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 24 / 24 (autogrow)
 Total addresses                : 6
 Leased addresses               : 0
 Pending event                  : none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased addresses
 172.16.0.1           172.16.0.1       - 172.16.0.6       0
```

Step 3 **show ip dhcp binding** The following output shows the bindings from pool Green. The Type field shows On-demand, which indicates that the address binding was created for a PPP session. The Lease expiration field shows Infinite, which means that the binding is valid as long as the session is up. If a subnet must be released back to the leasing server while the session is still up, the session is reset so that it will be forced to obtain a new IP address. The Hardware address column for an On-demand entry shows the identifier for the session as detected by PPP. There are no bindings shown under the Bindings from all pools not associated with VRF field because the Global pool has not allocated any addresses.

Example:

```
Router# show ip dhcp binding

Bindings from all pools not associated with VRF:
IP address       Hardware address        Lease expiration        Type
Bindings from VRF pool Green:
IP address       Hardware address        Lease expiration        Type
172.16.0.1       5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3930.39
172.16.0.2       5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3839.31
172.16.0.3       5674.312d.7465.7374.    Infinite                On-demand
                 2d36.3432.34
172.16.0.4       5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3236.34
172.16.0.5       5674.312d.7465.7374.    Infinite                On-demand
                 2d34.3331.37
172.16.0.6       5674.312d.7465.7374.    Infinite                On-demand
                 2d37.3237.39
172.16.0.9       5674.312d.7465.7374.    Infinite                On-demand
                 2d39.3732.36
172.16.0.10      5674.312d.7465.7374.    Infinite                On-demand
                 2d31.3637
172.16.0.11      5674.312d.7465.7374.    Infinite                On-demand
                 2d39.3137.36
172.16.0.12      5674.312d.7465.7374.    Infinite                On-demand
                 2d37.3838.30
172.16.0.13      5674.312d.7465.7374.    Infinite                On-demand
                 2d32.3339.37
172.16.0.14      5674.312d.7465.7374.    Infinite                On-demand
                 2d31.3038.31
172.16.0.17      5674.312d.7465.7374.    Infinite                On-demand
                 2d38.3832.38
```

```
172.16.0.18       5674.312d.7465.7374.    Infinite                On-demand
                  2d32.3735.31
```

Troubleshooting Tips

You can determine the client identifier by using the **debug ip dhcp server packet** command. In the following sample output, the client is identified by the value 0b07.1134.a029:

```
Device# debug ip dhcp server packet

DHCPD:DHCPDISCOVER received from client 0b07.1134.a029 through relay 10.1.0.253.
DHCPD:assigned IP address 10.1.0.3 to client 0b07.1134.a029.
.
.
.
```

Monitoring and Maintaining the ODAP

This task shows how to monitor and maintain the ODAP.

Note the following behavior for the **clear ip dhcp binding**, **clear ip dhcp conflict**, and **clear ip dhcp subnet** commands:

- If you do not specify the **pool** *pool-name* option and an IP address is specified, it is assumed that the IP address is an address in the global address space and will look among all the non-VRF DHCP pools for the specified binding/conflict/subnet.
- If you do not specify the **pool** *pool-name* option and the * option is specified, it is assumed that all automatic/ or on-demand bindings/conflicts/subnets in all VRF and non-VRF pools are to be deleted.
- If you specify both the **pool** *pool-name* option and the * option, all automatic or on-demand bindings/conflicts/subnets in the specified pool only will be cleared.
- If you specify the **pool** *pool-name* option and an IP address, the specified binding/conflict or the subnet containing the specified IP address will be deleted from the specified pool.

SUMMARY STEPS

1. **enable**
2. **clear ip dhcp** [**pool** *pool-name*] **binding** {***** | *address*}
3. **clear ip dhcp** [**pool** *pool-name*] **conflict** {***** | *address*}
4. **clear ip dhcp** [**pool** *pool-name*] **subnet**{*****| *address*}
5. **debug dhcp details**
6. **debug ip dhcp server events**
7. **show ip dhcp import**
8. **show ip interface** [*type number*]
9. **show ip dhcp pool** *pool-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip dhcp** [**pool** *pool-name*] **binding** {***** | *address*} **Example:** `Router# clear ip dhcp binding *`	Deletes an automatic address binding or objects from a specific pool from the DHCP server database.
Step 3	**clear ip dhcp** [**pool** *pool-name*] **conflict** {***** | *address*} **Example:** `Router# clear ip dhcp conflict *`	Clears an address conflict or conflicts from a specific pool from the DHCP server database.
Step 4	**clear ip dhcp** [**pool** *pool-name*] **subnet**{*****| *address*} **Example:** `Router# clear ip dhcp subnet *`	Clears all currently leased subnets in the named DHCP pool or all DHCP pools if *name* is not specified.
Step 5	**debug dhcp details** **Example:** `Router# debug dhcp details`	Monitors the subnet allocation/releasing in the on-demand address pools.
Step 6	**debug ip dhcp server events** **Example:** `Router# debug ip dhcp server events`	Reports DHCP server events, like address assignments and database updates.
Step 7	**show ip dhcp import** **Example:** `Router# show ip dhcp import`	Displays the option parameters that were imported into the DHCP server database.
Step 8	**show ip interface** [*type number*] **Example:** `Router# show ip interface`	Displays the usability status of interfaces configured for IP.
Step 9	**show ip dhcp pool** *pool-name* **Example:** `Router# show ip dhcp pool green`	Displays DHCP address pool information.

How to Configure DHCP ODAP Subnet Allocation Server Support

Configuring a Global Pool on a Subnet Allocation Server

Perform this task to configure a global subnet pool on a subnet allocation server.

Global Subnet Pools

Global subnet pools are created in a centralized network. The ODAP manager allocates subnets from the subnet allocation server based on subnet availability. When the ODAP manager allocates a subnet, the subnet allocation server creates a subnet binding. This binding is stored in the DHCP database for as long as the ODAP manager requires the address space. The binding is destroyed and the subnet is returned to the subnet pool only when the ODAP manager releases the subnet as address space utilization decreases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **network** *network-number* [*mask*| / *prefix-length*]
5. **subnet prefix-length** *prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool GLOBAL-POOL`	Enters DHCP pool configuration mode and specifies the subnet pool name.
Step 4	**network** *network-number* [*mask*| / *prefix-length*] **Example:** `Router(dhcp-config)# network 10.0.0.0 255.255.255.0`	Configures the subnet number and mask for a DHCP address pool on a DHCP server. • The subnet mask or the prefix length can be configured in this step. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation. The forward slash character must be used when configuring the *prefix-length* argument.

	Command or Action	Purpose
Step 5	**subnet prefix-length** *prefix-length* **Example:** `Router(dhcp-config)# subnet prefix-length 8`	Configures the subnet prefix length. The range of the *prefix-length* argument is from 1 to 31. • This command configures the number of IP addresses that each subnet is configured to allocate from the subnet pool. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation format.

Configuring a VRF Subnet Pool on a Subnet Allocation Server

VRF Subnet Pools

A subnet allocation server can be configured to assign subnets from VRF subnet allocation pools for MPLS VPN clients. VPN routes between the ODAP manager and the subnet allocation server are configured based on VRF name or VPN ID configuration. The VRF and VPN ID are configured to maintain routing information that defines customer VPN sites. The VPN customer site (or Customer Equipment [CE]) is attached to a provider edge (PE) router. The VRF is used to specify the VPN and consists of an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

Before you begin

The VRF name and VPN ID can be configured on the ODAP manager and subnet allocation server prior to the configuration of the subnet allocation pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **vrf** vrf-name
5. **network** *network-number* [*mask* |*/prefix-length*]
6. **subnet prefix-length** *prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool VRF-POOL`	Enters DHCP pool configuration mode and specifies the subnet pool name.
Step 4	**vrf** vrf-name **Example:** `Router(dhcp-config)# vrf RED`	Associates the on-demand address pool with a VPN routing and forwarding (VRF) instance name (or tag). • The **vrf**keyword and *vrf-name* argument are used to specify the VPN for the VRF pool. The *vrf-name* argument must match the VRF name (or tag) that is configured for the client.
Step 5	**network** *network-number* [*mask* |*/prefix-length*] **Example:** `Router(dhcp-config)# network 10.1.1.0 /24`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS XE DHCP server. • The subnet mask or the prefix length can be configured in this step. The values that can be configured for the *prefix-length*argument follow CIDR bit count notation. The forward slash character must be used when configuring the *prefix-length* argument.
Step 6	**subnet prefix-length** *prefix-length* **Example:** `Router(dhcp-config)# subnet prefix-length 16`	Configures the subnet prefix length. The range of the *prefix-length* argument is from 1 to 31. • This command configures the number of IP addresses that each subnet is configured to allocate from the subnet pool. The values that can be configured for the *prefix-length*argument follow CIDR bit count notation format.

Using a VPN ID to Configure a VRF Subnet Pool on a Subnet Allocation Server

Perform this task to configure a VRF subnet pool, using a VPN ID, on a subnet allocation server.

VRF Pools and VPN IDs

A subnet allocation server can also be configured to assign subnets from VPN subnet allocation pools based on the VPN ID of a client. The VPN ID (or Organizational Unique Identifier [OUI]) is a unique identifier assigned by the IEEE.

Before you begin

The VRF name and VPN ID can be configured on the ODAP manager and subnet allocation server prior to the configuration of the subnet allocation pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf vrf-name**
4. **rd** *route-distinguisher*
5. **route-target both** route-target-number
6. **vpn id** vpn-id
7. **exit**
8. **ip dhcp pool** pool-name
9. **vrf** vrf-name
10. **network** *network-number* [*mask* |*/prefix-length*]
11. **subnet prefix-length** *prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf vrf-name** **Example:** `Router(config)#ip vrf RED`	Creates a VRF routing table and specifies the VRF name (or tag). • The *vrf-name* argument must match the VRF name that is configured for the client and VRF pool in Step 9.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF instance created in Step 3. • There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).
Step 5	**route-target both** route-target-number **Example:** `Router(config-vrf)# route-target both 100:1`	Creates a route-target extended community for the VRF instance that was created in Step 3. • The **both** keyword is used to specify which routes should be imported and exported to the target VPN extended community (or the ODAP manager in this configuration).

	Command or Action	Purpose
		• The *route-target-number*argument follows the same format as the *route-distinguisher* argument in Step 4. These two arguments must match.
Step 6	**vpn id** vpn-id **Example:** `Router(config-vrf)# vpn id 1234:123456`	Configures the VPN ID. • This command is only used if the client (ODAP manager) is also configured with or assigned a VPN ID.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 8	**ip dhcp pool** pool-name **Example:** `Router(config)# ip dhcp pool VPN-POOL`	Enters DHCP pool configuration mode and specifies the subnet pool name. • The **VRF**keyword and *vrf-name* argument are used to specify the VPN for the VRF pool. The *vrf-name* argument must match the VRF name (or tag) that is configured for the client.
Step 9	**vrf** vrf-name **Example:** `Router(dhcp-config)#vrf RED`	Associates the on-demand address pool with a VRF instance name. • The *vrf-name* argument must match the *vrf-name* argument that was configured in Step 3.
Step 10	**network** *network-number* [*mask* |*/prefix-length*] **Example:** `Router(dhcp-config)# network 192.168.0.0 /24`	Configures the subnet number and mask for a DHCP address pool on a Cisco IOS XE DHCP server. • The subnet mask or the prefix length can be configured in this step. The values that can be configured for the *prefix-length*argument follow CIDR bit count notation. The forward slash character must be used when configuring the *prefix-length* argument.
Step 11	**subnet prefix-length** *prefix-length* **Example:** `Router(dhcp-config)# subnet prefix-length 16`	Configures the subnet prefix length. • The range of the *prefix-length* argument is from 1 to 31. • This command configures the number of IP addresses that each subnet is configured to allocate from the subnet pool. The values that can be configured for the *prefix-length* argument follow CIDR bit count notation format.

Verifying the Subnet Allocation and DHCP Bindings

Perform this task to verify subnet allocation and DHCP bindings.

The **show ip dhcp pool** and **show ip dhcp binding**commands do not need to be issued together or even in the same session as there are differences in the information that is provided. These commands, however, can be used to display and verify subnet allocation and DHCP bindings. The **show running-config | begin dhcp** command is used to display the local configuration of DHCP and the configuration of the **subnet prefix-length** command.

SUMMARY STEPS

1. **enable**
2. **show running-config** | begin dhcp
3. **show ip dhcp pool [pool-name]**
4. **show ip dhcp binding [ip-address]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** | begin dhcp **Example:** `Router# show running-config | begin dhcp`	Used to display the local configuration of the router. • The configuration of the **subnet prefix-length** command will be displayed under the DHCP pools, for which subnet lease allocation has been configured. The subnet allocation size will be shown, following this command, in CIDR bit count notation. • The sample output is filtered with the **begin** keyword to start displaying output at the DHCP section of the running configuration.
Step 3	**show ip dhcp pool [pool-name]** **Example:** `Router# show ip dhcp pool`	Displays information about DHCP pools. • This command can be used to verify subnet allocation pool configuration on both the subnet allocation server and the ODAP manager. • The output of this command displays specific address pool information, including the name of the pool, utilization of address space, subnet size, number of total addresses, number of leased address, and pending events.
Step 4	**show ip dhcp binding [ip-address]** **Example:**	Displays information about DHCP bindings.

	Command or Action	Purpose
	`Router# show ip dhcp binding`	• This command can be used to display subnet allocation to DHCP binding mapping information. • The output from this command displays binding information for individual IP address assignment and allocated subnets. The output that is generated for DHCP IP address assignment and subnet allocation is almost identical, except that subnet leases display an IP address followed by the subnet mask (which shows the size of the allocated subnet). Bindings for individual IP address only display an IP address and are not followed by a subnet mask.

Troubleshooting the DHCP ODAP Subnet Allocation Server

Perform this task to troubleshoot the DHCP ODAP subnet allocation server.

SUMMARY STEPS

1. **enable**
2. **debug dhcp** [**detail**]
3. **debug ip dhcp server** {**events** | **packets** | **linkage**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug dhcp** [**detail**] **Example:** `Router# debug dhcp detail`	Displays debugging information about DHCP client activities and monitors the status of DHCP packets. • This example is issued with the **detail**keyword on the ODAP manager. The **detail**keyword is used to display and monitor the lease entry structure of the client and the state transitions of lease entries. This command also displays the values of the op, htype, hlen, hops, server identifier option, xid, secs, flags, ciaddr, yiaddr, siaddr, and giaddr fields of the DHCP packet that are shown in addition to the length of the options field.
Step 3	**debug ip dhcp server** {**events**	**packets**	**linkage**} **Example:** `Router# debug ip dhcp server packets`	Enables DHCP server debugging. • This example is issued with the **packets** and events keywords on the subnet allocation server. The output

	Command or Action	Purpose
	Example: `Router# debug ip dhcp server events`	displays lease transition and reception, as well as database information.

Configuration Examples for DHCP Server On-Demand Address Pool Manager

Defining DHCP ODAPs as the Global Default Mechanism Example

The following example shows how to configure the on-demand address pooling mechanism to be used to serve an address request from a PPP client.

```
ip address-pool dhcp-pool
!
ip dhcp pool Green-pool
```

Defining DHCP ODAPs on an Interface Example

The following example shows how to configure an interface to retrieve an IP address from an on-demand address pool:

```
interface Virtual-Template1
 ip vrf forwarding green
 ip unnumbered loopback1
 ppp authentication chap
 peer default ip address dhcp-pool
!
```

Configuring the DHCP Pool as an ODAP Example

The following example shows two ODAPs configured to obtain their subnets from an external DHCP server:

```
Router# show run
Building configuration...
Current configuration : 3943 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
no logging console
enable password lab
!
username vpn_green_net1 password 0 lab
username vpn_red_net1 password 0 lab
ip subnet-zero
```

```
!
ip dhcp pool green_pool
   vrf Green
   utilization mark high 60
   utilization mark low 40
   origin dhcp subnet size initial /24 autogrow /24
!
ip dhcp pool red_pool
   vrf Red
   origin dhcp
!
ip vrf Green
 rd 200:1
 route-target export 200:1
 route-target import 200:1
!
ip vrf Red
 rd 300:1
 route-target export 300:1
 route-target import 300:1
ip cef
ip address-pool dhcp-pool
!
no voice hpi capture buffer
no voice hpi capture destination
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface Loopback1
 ip vrf forwarding Green
 ip address 100.10.10.1 255.255.255.255
!
interface Loopback2
 ip vrf forwarding Red
 ip address 110.10.10.1 255.255.255.255
!
interface ATM2/0/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
interface ATM3/0/0
 no ip address
 no atm ilmi-keepalive
!
interface GigabitEthernet0/0/0
 ip address 10.0.105.12 255.255.255.224
 duplex half
!
interface GigabitEthernet0/0/1
 ip address 150.10.10.1 255.255.255.0
 duplex half
!
interface GigabitEthernet0/0/2
 ip address 120.10.10.1 255.255.255.0
 duplex half
 tag-switching ip
!
interface Virtual-Template1
 ip vrf forwarding Green
 ip unnumbered Loopback1
 ppp authentication chap
!
```

```
interface Virtual-Template2
 ip vrf forwarding Green
 ip unnumbered Loopback1
 ppp authentication chap
!
interface Virtual-Template3
 ip vrf forwarding Green
 ip unnumbered Loopback1
 ppp authentication chap
!
interface Virtual-Template4
 ip vrf forwarding Red
 ip unnumbered Loopback2
 ppp authentication chap
!
interface Virtual-Template5
 ip vrf forwarding Red
 ip unnumbered Loopback2
 ppp authentication chap
!
interface Virtual-Template6
 ip vrf forwarding Red
 ip unnumbered Loopback2
 ppp authentication chap
!
router ospf 100
 log-adjacency-changes
 redistribute connected
 network 1.1.1.1 0.0.0.0 area 0
 network 120.10.10.0 0.0.0.255 area 0
 network 150.10.10.0 0.0.0.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 !
 address-family ipv4 vrf Red
 redistribute connected
 redistribute static
 no auto-summary
 no synchronization
 network 110.0.0.0
 exit-address-family
 !
 address-family ipv4 vrf Green
 redistribute connected
 redistribute static
 no auto-summary
 no synchronization
 network 100.0.0.0
 exit-address-family
 !
 address-family vpnv4
 neighbor 3.3.3.3 activate
 neighbor 3.3.3.3 send-community extended
 exit-address-family
!
ip classless
ip route 172.19.0.0 255.255.0.0 10.0.105.1
no ip http server
ip pim bidir-enable
!
```

```
call rsvp-sync
!
mgcp profile default
!
dial-peer cor custom
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 password lab
 login
!
end
```

Configuring the DHCP Pool as an ODAP for Non-MPLS VPNs Example

The following example shows how to configure an interface to retrieve an IP address from an on-demand address pool. In this example, two non-VRF ODAPs are configured. There are two virtual-templates and two DHCP address pools, usergroup1 and usergroup2. Each virtual-template interface is configured to obtain IP addresses for the peer from the associated address pool.

```
!
ip dhcp pool usergroup1
 origin dhcp subnet size initial /24 autogrow /24
 lease 0 1
!
ip dhcp pool usergroup2
 origin dhcp subnet size initial /24 autogrow /24
 lease 0 1
!
interface virtual-template1
 ip unnumbered loopback1
 peer default ip address dhcp-pool usergroup1
!
interface virtual-template2
 ip unnumbered loopback1
 peer default ip address dhcp-pool usergroup2
```

Configuring AAA and RADIUS Example

The following example shows one pool “Green” configured to obtain its subnets from the AAA (RADIUS) server located at IP address 172.16.1.1:

```
!
aaa new-model
!
aaa authorization configuration default group radius
aaa accounting network default start-stop group radius
aaa session-id common
!
ip subnet-zero
!
ip dhcp ping packets 0
!
```

```
ip dhcp pool Green
 vrf Green
 utilization mark high 50
 utilization mark low 30
 origin aaa subnet size initial /28 autogrow /28
!
ip vrf Green
 rd 300:1
 route-target export 300:1
 route-target import 300:1
!
interface GigabitEthernet0/1/1
 ip address 172.16.1.12 255.255.255.0
 duplex half
!
interface Virtual-Template1
 ip vrf forwarding Green
 no ip address
!
ip radius source-interface GigabitEthernet0/1/1
!
!IP address of the RADIUS server host
radius-server host 172.16.1.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server attribute 32 include-in-access-req
radius-server attribute 44 include-in-access-req
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
```

Configuring a Global Pool for a Subnet Allocation Server Example

The following example shows how to configure a router to be a subnet allocation server and create a global subnet allocation pool named “GLOBAL-POOL” that allocates subnets from the 10.0.0.0/24 network. The configuration of the **subnet prefix-length** command in this example configures the size of each subnet that is allocated from the subnet pool to support 254 host IP addresses.

```
ip dhcp pool GLOBAL-POOL
 network 10.0.0.0 255.255.255.0
 subnet prefix-length 24
!
```

Configuring a VRF Pool for a Subnet Allocation Server Example

The following example shows how to configure a router to be a subnet allocation server and create a VRF subnet allocation pool named “VRF-POOL” that allocates subnets from the 172.16.0.0/16 network and configures the VPN to match the VRF named “RED.” The configuration of the **subnet prefix-length** command in this example configures the size of each subnet that is allocated from the subnet pool to support 62 host IP addresses.

```
 ip dhcp pool VRF-POOL
 vrf RED
 network 172.16.0.0 /16
 subnet prefix-length 26
!
```

Using a VPN ID to Configure a VRF Pool on a Subnet Allocation Server Example

The following example shows how to configure a router to be a subnet allocation server and create a VRF subnet allocation pool named “VRF-POOL” that allocates subnets from the 192.168.0.0/24 network and configures the VRF named “RED.” The VPN ID must match the unique identifier that is assigned to the client site. The route target and route distinguisher are configured in the as-number:network-number format. The route target and route distinguisher must match. The configuration of the **subnet prefix-length** command in this example configures the size of each subnet that is allocated from the subnet pool to support 30 host IP addresses.

```
ip vrf RED
 rd 100:1
 route-target both 100:1
 vpn id 1234:123456
 exit
 ip dhcp pool VPN-POOL
  vrf RED
  network 192.168.0.0 /24
  subnet prefix-length /27
  exit
```

Verifying Local Configuration on a Subnet Allocation Server Example

The following example is output from the **show running-config**command. This command can be used to verify the local configuration on a subnet allocation server. The output from this command displays the configuration of the subnet prefix-length command under the DHCP pool named “GLOBAL-POOL.” The total size of the subnet allocation pool is set to 254 addresses with the **network** command. The configuration of the **subnet prefix-length** command configures this pool to allocate a subnet that will support 254 host IP addresses. Because the total pool size supports only 254 addresses, only one subnet can be allocated from this pool.

```
Router# show running-config | begin dhcp
ip dhcp pool GLOBAL-POOL
   network 10.0.0.0 255.255.255.0
   subnet prefix-length 24
!
```

Verifying Address Pool Allocation Information Example

The following examples are output from the **show ip dhcp pool**command. This command can be used to verify subnet allocation pool configuration on the subnet allocation server and the ODAP manager. The output from this command displays information about the address pool name, utilization level, configured subnet size, total number of addresses (from subnet), pending events, and specific subnet lease information.

The following sample output shows that the configured subnet allocation size is /24 (254 IP addresses), that there is a pending subnet allocation request, and there are no subnets in the pool:

```
Router> show ip dhcp pool ISP-1
Pool ISP-1 :
 Utilization mark (high/low)    :100 / 0
 Subnet size (first/next)       :24 / 24 (autogrow)
 Total addresses                :0
 Leased addresses               :0
```

```
 Pending event                  :subnet request
 0 subnet is currently in the pool
```

The next example shows that the configured subnet allocation size is /24 (254 IP address), the configured VRF name is “RED”, and a subnet containing 254 IP addresses has been allocated but no IP addresses have been leased from the subnet:

```
Router> show ip dhcp pool SUBNET-ALLOC
Pool SUBNET-ALLOC :
 Utilization mark (high/low)    :100 / 0
 Subnet size (first/next)       :24 / 24 (autogrow)
 VRF name                       :RED
 Total addresses                :254
 Leased addresses               :0
 Pending event                  :none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased addresses
 10.0.0.1              10.0.0.1          - 10.0.0.254          0
```

Verifying Subnet Allocation and DHCP Bindings Example

The following example is from the **show ip dhcp binding** command. This command can be used to display subnet allocation to DHCP binding mapping information. The output of this command shows the subnet lease to MAC address mapping, the lease expiration, and the lease type (subnet lease bindings are configured to be automatically created and released by default). The output that is generated for DHCP IP address assignment and subnet allocation is almost identical, except that subnet leases display an IP address followed by the subnet mask (which shows the size of the allocated subnet) in CIDR bit count notation. Bindings for individual IP address only display an IP address and are not followed by a subnet mask.

```
Router# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.0.0.0/26         0063.6973.636f.2d64.    Mar 29 2009 04:36 AM    Automatic
                    656d.6574.6572.2d47.
                    4c4f.4241.4c
```

Additional References

The following sections provide references related to configuring the DHCP ODAP manager.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	“DHCP Overview” module
DHCP server configuration	“Configuring the Cisco IOS XE DHCP Server” module

Related Topic	Document Title
DHCP client configuration	"Configuring the Cisco IOS XE DHCP Client" module
DHCP relay agent configuration	"Configuring the Cisco IOS XE DHCP Relay Agent" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module

Standards

Standards	Title
No new or modified standards are supported by this functionality.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2685	*Virtual Private Networks Identifier*
RFC 3046	*DHCP Relay Information Option*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/public/support/tac/home.shtml

Feature Information for the DHCP Server On-Demand Address Pool Manager

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 84: Feature Information for the DHCP On-Demand Address Pool Manager

Feature Name	Releases	Feature Configuration Information
DHCP Server On-Demand Address Pool Manager for Non-MPLS VPNs	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	This feature was enhanced to provide ODAP support for non-MPLS VPNs. The following command was modified by this feature: **peer default ip address**
DHCP ODAP Server Support	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	This feature introduces the capability to configure a DHCP server (or router) as a subnet allocation server. This capability allows the Cisco IOS DHCP server to be configured with a pool of subnets for lease to ODAP clients. The following commands were introduced or modified by this feature: **subnet prefix-length** and **show ip dhcp binding**

Feature Name	Releases	Feature Configuration Information
DHCP Server On-Demand Address Pool Manager	Cisco IOS XE Release 2.3	The ODAP manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses. When a Cisco IOS router is configured as an ODAP manager, pools of IP addresses are dynamically increased or reduced in size depending on the address utilization level. The following commands were introduced by this feature: **aaa session-id**, **clear ip dhcp subnet**, **ip address pool**, **ip dhcp aaa default username**, **origin**, **show ip dhcp pool**, **utilization mark high**, **utilization mark low**, **vrf**. The following commands were modified by this feature: **clear ip dhcp binding**, **clear ip dhcp conflict**, **ip address-pool**, **peer default ip address**.

Glossary

AAA --authentication, authorization, and accounting. Suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server.

Cisco Access Registrar --A RADIUS server that supports service provider deployment of access services by centralizing AAA information and simplifying provisioning and management.

client --A host trying to configure its interface (obtain an IP address) using DHCP or BOOTP protocols.

DHCP --Dynamic Host Configuration Protocol.

incremental subnet size --The desired size of the second and subsequent subnets requested for an on-demand pool.

initial subnet size --The desired size of the first subnet requested for an on-demand pool.

IPCP --IP Control Protocol. Protocol that establishes and configures IP over PPP.

MPLS --Multiprotocol Label Switching. Emerging industry standard upon which tag switching is based.

ODAP --on-demand address pool.

PE router --provider edge router.

PPP --Point-to-Point Protocol.

RADIUS -- Remote Authentication Dial-In User Service. Database for authenticating modem and ISDN connections and for tracking connection time.

relay agent --A router that forwards DHCP and BOOTP messages between a server and a client on different subnets.

releasable subnet --A leased subnet that has no address leased from it.

server --DHCP or BOOTP server.

VHG --Virtual Home Gateway. A Cisco IOS software component that terminates PPP sessions. It is owned and managed by the service provider on behalf of its customer to provide access to remote users of that

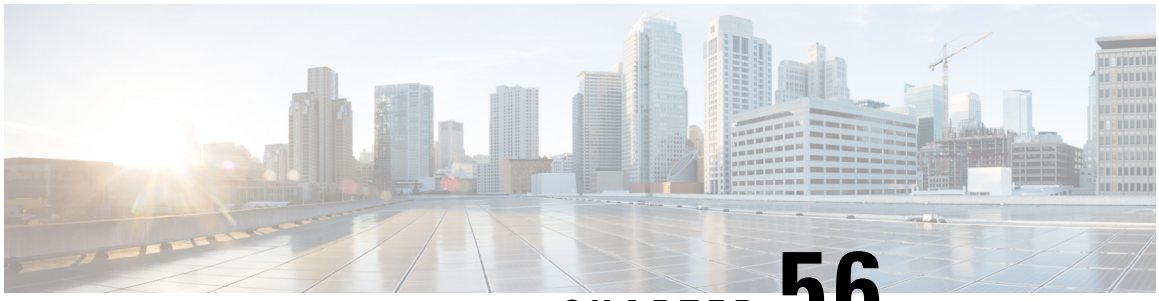
customers network. A single service provider device (router) can host multiple VHGs of different customers. A VHG can be dynamically brought up and down based on the access pattern of the remote users. Note that there is no single IOS feature called the VHG; it is a collection of function and features.

VHG/PE router--A device that terminates PPP sessions and maps the remote users to the corresponding MPLS VPNs.

VPN --Virtual Private Network. Enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

VPN information --In this document, VPN information refers to VRF name or VPN ID.

VRF --VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router. Each VPN instantiated on the PE router has its own VRF.



CHAPTER 56

IPv6 Access Services: DHCPv6 Relay Agent

A Dynamic Host Configuration Protocol for IPv6 (DHCPv6) relay agent, which may reside on the client's link, is used to relay messages between the client and the server.

DHCPv6 Relay Agent

A DHCPv6 relay agent, which may reside on the client's link, is used to relay messages between the client and the server. The DHCPv6 relay agent operation is transparent to the client. A DHCPv6 client locates a DHCPv6 server using a reserved, link-scoped multicast address. For direct communication between the DHCPv6 client and the DHCPv6 server, both of them must be attached to the same link. However, in some situations where ease of management, economy, or scalability is a concern, it is desirable to allow a DHCPv6 client to send a message to a DHCPv6 server that is not connected to the same link. IPv6 enable is required for IPv6 DHCP relay, although IPv6 address is configured.

DHCPv6 Relay Agent Notification for Prefix Delegation

The DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is relayed by the relay agent to the client. When a prefix delegation option is found by the relay agent, the relay agent extracts the information about the prefix that is being delegated and inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

No user configuration is required for this feature. Static route management is done automatically by the relay agent.

IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

The DHCP—DHCPv6 Relay Agent Notification for Prefix Delegation feature leaves a static IPv6 route on the routing table of the relay agent. The registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the device doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that remains in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. Static routes will be removed when a DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote-ID for Gigabit Ethernet and Fast Ethernet Interfaces

The DHCPv6 Ethernet Remote ID Option feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, which includes port information, the system's DUID, and the VLAN ID. This information can be used to uniquely identify both the relay and the port on the relay through which the client packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem.

The addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically and no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. The Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is removed from the packet before it is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface ID

The DHCPv6 Relay—Reload Persistent Interface ID Option feature makes the interface ID option persistent. The interface ID is used by relay agents to decide which interface should be used to forward a RELAY-REPLY packet. A persistent interface-ID option will not change if the device acting as a relay agent goes offline during a reload or a power outage. When the device acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as, when the relay agent reboots and the number of interfaces in the interface index changes, or when the relay agents boot up and has more virtual interfaces than it did before the reboot). This feature prevents such scenarios from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as the short form of the interface name. The interface name as the DHCPv6 interface ID helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

DHCPv6 Relay Chaining

DHCPv6 messages can be relayed through multiple relay agents. This configuration is called *relay chaining*. A relay chaining configuration can be supported only when each relay agent adds information to DHCPv6 messages before relaying them. The information helps in relaying the DHCPv6 reply back to the DHCPv6 client through the same path.

The delegated IPv6 prefix must be routable in order to be useful. The actual DHCPv6 Prefix Delegation (PD) client may not be permitted to inject routes into the delegating network. In service provider (SP) networks, for example, an edge device typically acts as a DHCPv6 relay agent, and this edge device often has the responsibility to maintain routes within the SP network for clients' PD bindings. In the event that DHCPv6 requests and responses are relayed through a chain of DHCPv6 relays, there may be a need to introduce

appropriate routes (particularly with DHCPv6 PD) in the Forwarding Information Base (FIB) so that routing is handled transparently.

DHCPv6 Relay Agent Notification for Prefix Delegation

The DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is relayed by the relay agent to the client. When a prefix delegation option is found by the relay agent, the relay agent extracts the information about the prefix that is being delegated and inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

No user configuration is required for this feature. Static route management is done automatically by the relay agent.

IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

The DHCP—DHCPv6 Relay Agent Notification for Prefix Delegation feature leaves a static IPv6 route on the routing table of the relay agent. The registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the device doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that remains in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. Static routes will be removed when a DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote ID for Ethernet Interfaces

This feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, including port information, the system's DUID, and the VLAN ID. Collectively, this information can be used to uniquely identify both the relay and the port on the relay through which the client's packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem. This feature works only for Ethernet interfaces at this time.

This feature introduces no user configuration. Because the addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically, no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is stripped out of the packet before the packet is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface ID Option

This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. A persistent interface-ID option will not change

if the router acting as a relay agent goes offline (such as during a reload or a power outage). When the router acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as cases where the relay agent reboots and has a change in the number of interfaces in the interface index, or the relay agents boots up and has more virtual interfaces than it did before the reboot). This feature prevents this scenario from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as simply the short form of the interface name. This syntax helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

DHCPv6 Relay Chaining

This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. A persistent interface-ID option will not change if the router acting as a relay agent goes offline (such as during a reload or a power outage). When the router acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as cases where the relay agent reboots and has a change in the number of interfaces in the interface index, or the relay agents boots up and has more virtual interfaces than it did before the reboot). This feature prevents this scenario from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as simply the short form of the interface name. This syntax helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

How to Configure IPv6 Access Services: DHCPv6 Relay Agent

Configuring the DHCPv6 Relay Agent

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 enable**
5. **ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 4/2/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 5	**ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*] **Example:** `Device(config-if) ipv6 dhcp relay destination FE80::250:A2FF:FEBF:A056 gigabitethernet 4/3/0`	Specifies a destination address to which client packets are forwarded and enables the DHCPv6 relay service on the interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for IPv6 Access Services: DHCPv6 Relay Agent

Example: Configuring the DHCPv6 Relay Agent

```
Device# show ipv6 dhcp interface

Ethernet1/0 is in relay mode
  Relay destinations:
    3FFB:C00:C18:6:A8BB:CCFF:FE03:2701
Serial3/0 is in relay mode
  Relay destinations:
    3FFB:C00:C18:6:A8BB:CCFF:FE03:2600
    FE80::A8BB:CCFF:FE03:2801 on Serial3/0
    FF05::1:3
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

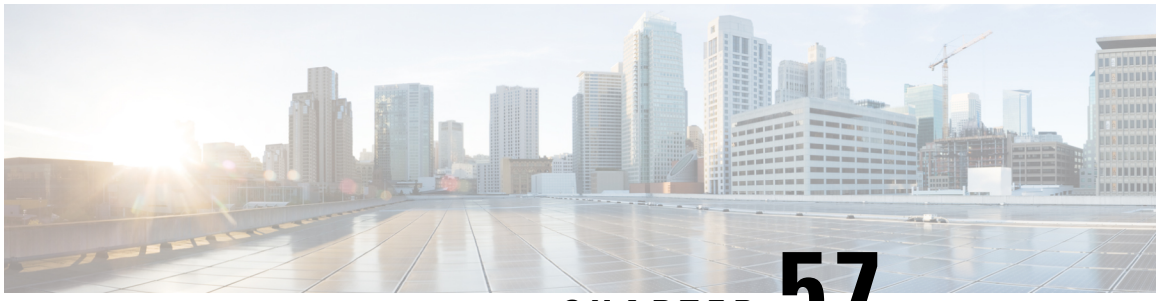
Feature Information for IPv6 Access Services: DHCPv6 Relay Agent

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 85: Feature Information for IPv6 Access Services: DHCPv6 Relay Agent

Feature Name	Releases	Feature Information
IPv6 Access Services: DHCPv6 Relay Agent		A DHCP relay agent, which may reside on the client's link, is used to relay messages between the client and server. The following commands were introduced or modified: **ipv6 dhcp relay destination**, **show ipv6 dhcp interface**.
DHCPv6 Relay Agent Notification for Prefix Delegation		DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 packet that is being relayed by the relay agent to the client.
DHCPv6 Relay: Reload Persistent Interface ID Option		This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent.



CHAPTER 57

DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

The DHCP Relay Server ID Override and Link Selection Option 82 Suboptions feature enables the relay agent to be part of all Dynamic Host Configuration Protocol (DHCP) message exchanges by supporting the use of two suboptions of the relay agent information option (option 82). This design allows DHCPv4 to operate in networks where direct communication between the client and server is not possible or desired. These two suboptions used together enable the deployment of an architecture where having all DHCP traffic flow through the relay agent is desirable, allowing for greater control of DHCP communications.

This feature also introduces the capability to manually configure the interface for the relay agent to use as the source IP address for messages relayed to the DHCP server. This configuration allows the network administrator to specify a stable, hardware-independent IP address (such as a loopback interface).

Restrictions for DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

If the DHCP Relay Server ID Override and Link Selection Option 82 Suboptions feature and the DHCP Relay MPLS VPN Support feature are both configured, the DHCP Relay MPLS VPN Support feature takes precedence.

Information About DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

Server ID Override Suboption

The server identifier (ID) override suboption allows the DHCP relay agent to specify a new value for the server ID option, which is inserted by the Dynamic Host Configuration Protocol (DHCP) server in the reply packet. This suboption allows the DHCP relay agent to act as the actual DHCP server such that the renew requests will come to the relay agent rather than the DHCP server directly. The server ID override suboption contains the incoming interface IP address, which is the IP address on the relay agent that is accessible from the client. The DHCP client uses this information to send all renew and release request packets to the relay agent. The relay agent adds all of the appropriate suboptions and then forwards the renew and release request packets to the original DHCP server.

Link Selection Suboption

The link selection suboption provides a mechanism to separate the subnet/link on which the DHCP client resides from the the gateway address (giaddr), which can be used to communicate with the relay agent by the DHCP server. The relay agent will set the suboption to the correct subscriber subnet and the DHCP server will use that value to assign an IP address rather than the giaddr value. The relay agent will set the giaddr to its own IP address so that DHCP messages are routable over the network.

DHCP Relay Server ID Override and Link Selection Option 82 Suboptions Feature Design

The Dynamic Host Configuration Protocol (DHCP) IPv4 deployment model assumes a single routing domain between the DHCP client and DHCP server. In some network designs, the DHCP server cannot directly communicate with DHCP clients. Customers may choose this design to make critical infrastructure servers inaccessible and to protect the DHCP server from client attacks.

Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet. In all cases, the DHCP relay agent must be able to communicate directly with both the DHCP server and DHCP client. By using the relay agent information option (option 82), the DHCP relay agent can include additional information about itself when forwarding client-originated DHCP packets to a DHCP server.

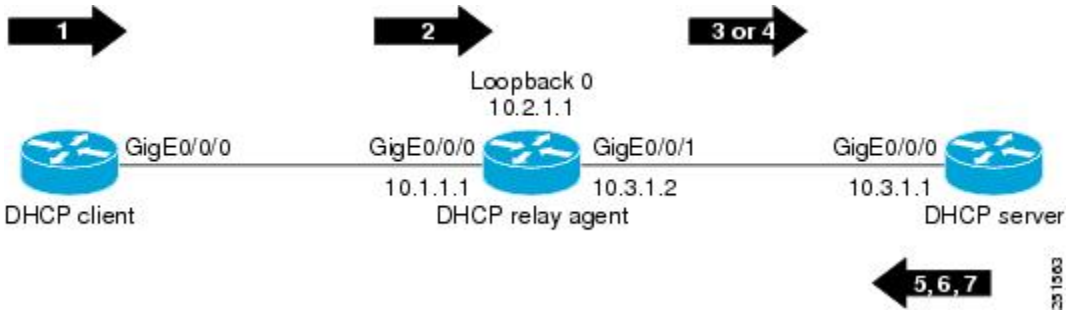
The DHCP Relay Server ID Override and Link Selection Option 82 Suboptions feature enables the relay agent to be part of all DHCP message exchanges by supporting the use of two suboptions of option 82: server ID override and link selection. This design results in all DHCP messages flowing through the relay agent, allowing for greater control of DHCP communications.

Communication from the DHCP server through the relay agent can be an issue. If the server needs to reach the client, it must do so through the relay agent. The IP address of the relay agent might not be ideal. For example, if the network is renumbered or if the interface at the relay agent is down for some reason, the server may not be able to reach the client. This feature introduces the capability to manually configure the interface for the relay agent to use as the source IP address for messages relayed to the DHCP server. This configuration

allows the network administrator to specify a stable, hardware-independent IP address (such as a loopback interface).

The figure and the numbered list that follows it shows the processing that occurs on the DHCP relay agent and DHCP server when this feature is configured.

Figure 63: DHCP Relay Agent and DHCP Server Processing of Option 82 Suboptions



1. The DHCP client generates a DHCP request and broadcasts it on the network.
2. The DHCP relay agent intercepts the broadcast DHCP request packet and inserts a server ID override suboption and link selection suboption to its relay agent information option in the DHCP packet. The server ID override and link selection suboptions contain the incoming interface IP address, which is the IP address on the relay agent that is accessible from the client (10.1.1.1 in this case).
3. The relay agent sets the gateway IP address (giaddr) to the IP address of an interface that is reachable by the DHCP server (typically the server-facing interface that will be used to transmit the message, 10.3.1.2 in this case).
4. If the source interface is explicitly configured on a loopback interface (using the **ip dhcp-relay source-interface** command), the relay agent will use that address as the source IP address (giaddr) for messages relayed to the DHCP server (10.2.1.1 in this case).

The following processing occurs on the DHCP server after receiving the forwarded packets from the relay agent:

1. The DHCP server uses the link selection suboption to locate the correct address pools for the DHCP client.
2. The DHCP server sets the server ID option to the value specified by the server ID override suboption of the DHCP packet.
3. The DHCP server sends the reply message to the IP address specified in the giaddr.

The DHCP client will see the relay agent address as the server ID and use that address when unicasting RENEW messages.

This DHCP server supports all the AireOS remote-id format combinations on eWLC along with delimiter '**:**' support.

How to Configure Support for the DHCP Relay Server ID Override and Link Selection Suboptions

Configuring the DHCP Relay Agent to Insert the DHCP Server ID Override and Link Selection Suboptions into Option 82

Note If the DHCP Relay Server ID Override and Link Selection Option 82 Suboptions feature and the DHCP Relay MPLS VPN Support feature are both configured, the DHCP Relay MPLS VPN Support feature takes precedence.

The eWLC does not support the ap-group and flex-connect group options, so "*ap-group-name"* and "*flex-group-name"* CLI is not supported. However, the eWLC supports policy-tag CLI which adds the site-tag to the remote-id. This CLI can be used as an alternative.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp-relay information option server-override**
4. **ip dhcp-relay source-interface** *type number*
5. **interface** *type number*
6. **ip dhcp relay information option server-id-override**
7. **ipdhcpcompatibilitysuboptionlink-selection<standard/cisco>**
8. **ipdhcpcompatibilitysuboptionserver-override<standard/cisco>**
9. **ip dhcp relay source-interface** *type number*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp-relay information option server-override** **Example:**	Enables the system to globally insert the server ID override and link selection suboptions into the DHCP relay agent

	Command or Action	Purpose
	`Device(config)# ip dhcp-relay information option server-override`	information option in forwarded BOOTREQUEST messages to a Dynamic Host Configuration Protocol (DHCP) server. • If the **ip dhcp relay information option server-id-override** command is configured on an interface, it overrides the global configuration on that interface only.
Step 4	**ip dhcp-relay source-interface** *type number* **Example:** `Device(config)# ip dhcp-relay source-interface loopback 0`	(Optional) Globally configures the source interface for the relay agent to use as the source IP address for relayed messages. • This command allows the network administrator to specify a stable, hardware-independent IP address (such as a loopback interface). • If the **ip dhcp relay source-interface** command is configured on an interface, it overrides the global configuration on that interface only.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	(Optional) Configures an interface and enters interface configuration mode.
Step 6	**ip dhcp relay information option server-id-override** **Example:** `Device(config-if)# ip dhcp relay information option server-id-override`	(Optional) Enables the system to insert the server ID override and link selection suboptions on a specific interface into the DHCP relay agent information option in forwarded BOOTREQUEST messages to a DHCP server.
Step 7	**ipdhcpcompatibilitysuboptionlink-selection<standard/cisco>** **Example:** `Device(config-if)# ip dhcp compatibility suboption link-selection <standard/cisco>`	Enables the system to configure corresponding link selection suboptions to the outgoing DHCP packets.
Step 8	**ipdhcpcompatibilitysuboptionserver-override<standard/cisco>** **Example:** `Device(config-if)# ip dhcp compatibility suboption server-override <standard/cisco>`	Enables the system to configure corresponding server-override sub-option values to the outgoing DHCP packets.
Step 9	**ip dhcp relay source-interface** *type number* **Example:** `Device(config-if)# ip dhcp relay source-interface loopback 2`	(Optional) Configures the source interface for the relay agent to use as the source IP address for relayed messages.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

Example: DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

In the following example, the IP address of the loopback interface is used as the source IP address for relayed messages. The client initiates IP address negotiation from GigabitEthernet interface 0/0/0. The Dynamic Host Configuration Protocol (DHCP) relay agent is configured globally to insert the server ID override suboption and link selection suboption into the relay agent information option of the DHCP packet. The relay agent uses the server ID override suboption to force the DHCP server to use that value as the server ID in the DHCP message. The DHCP server uses the link selection suboption to determine from which subnet to assign an IP address.

DHCP Client

```
interface GigabitEthernet 0/0/0
 ip address dhcp
```

DHCP Relay Agent

```
ip dhcp-relay information option server-override
ip dhcp-relay source-interface loopback 0
!
interface Loopback0
 ip address 10.2.1.1 255.255.255.0
!
interface GigabitEthernet 0/0/0
 ip address 10.1.1.1 255.255.255.0
 ip helper-address 10.3.1.1
!
interface GigabitEthernet 1/0/0
 ip address 10.3.1.2 255.255.255.0
```

DHCP Compatibility Suboption

```
ip dhcp compatibility suboption link-selection <standard/cisco>
ip dhcp compatibility suboption server-override <standard/cisco>
```

DHCP Server

```
ip dhcp excluded-address 10.3.0.1
ip dhcp pool pool1
 network 10.1.1.0 255.255.255.0
 lease 0 0 1
!
interface GigabitEthernet 0/0/0
 ip address 10.3.1.1 255.255.255.0
```

Additional References for DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IP addressing commands	Cisco IOS IP Addressing Services Command Reference
DHCP conceptual information	*DHCP Overview*
DHCP server configuration tasks, examples, and conceptual information	*Configuring the Cisco IOS DHCP Server*
DHCP relay agent configuration tasks, examples, and conceptual information	*Configuring the Cisco IOS DHCP Relay Agent*

Standards and RFCs

Standard/RFC	Title
RFC 3527	*Link Selection Suboption*
RFC 5107	*DHCP Server Identifier Override Suboption*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 86: Feature Information for DHCP Relay Server ID Override and Link Selection Option 82 Suboptions

Feature Name	Releases	Feature Configuration Information
DHCP Relay Server ID Override and Link Selection Option 82 Suboptions		The DHCP Relay Server ID Override and Link Selection Option 82 Suboptions feature enables the relay agent to be part of all Dynamic Host Configuration Protocol (DHCP) message exchanges by supporting the use of two suboptions of the relay agent information option (option 82). This design allows DHCPv4 to operate in networks where direct communication between the client and server is not possible or desired. These two suboptions used together enable the deployment of an architecture where having all DHCP traffic flow through the relay agent is desirable, allowing for greater control of DHCP communications. The following commands were introduced or modified: **ip dhcp relay information option server-id-override, ip dhcp relay source-interface, ip dhcp-relay information option server-override, ip dhcp-relay source-interface.**

Glossary

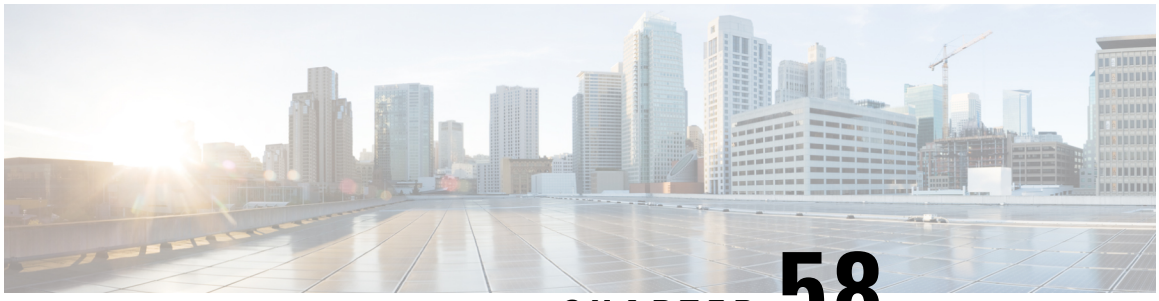
client—A host trying to configure its interface (obtain an IP address) using DHCP or BOOTP protocols.

DHCP—Dynamic Host Configuration Protocol.

DHCP options and suboptions—Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. Options provide a method of appending additional information. Vendors that want to provide additional information to their client not designed into the protocol can use options.

giaddr—Gateway IP address field of the DHCP packet. The giaddr provides the DHCP server with information about the IP address subnet in which the client resides. The giaddr also provides the DHCP server with an IP address where the DHCP response messages can be sent.

relay agent—A router that forwards DHCP and BOOTP messages between a server and a client on different subnets.



CHAPTER 58

DHCP Server RADIUS Proxy

The Dynamic Host Configuration Protocol (DHCP) Server RADIUS Proxy is a RADIUS-based address assignment mechanism in which a DHCP server authorizes remote clients and allocates addresses based on replies from a RADIUS server.

Prerequisites for DHCP Server RADIUS Proxy

Before you can configure the DHCP Server RADIUS Proxy, you must be running DHCPv4 or a later version. For information about release and platform support, see "Feature Information for DHCP Server RADIUS Proxy".

Restrictions for DHCP Server RADIUS Proxy

The DHCP Server RADIUS Proxy supports only one address authorization pool on the router.

Information About DHCP Server RADIUS Proxy

DHCP Server RADIUS Proxy Overview

The DHCP Server RADIUS Proxy feature is an address allocation mechanism for RADIUS-based authorization of DHCP leases. This feature supports DHCP options 60 and 121.

1. The DHCP server passes client information to a RADIUS server.

2. The RADIUS server returns all required information to the DHCP server as RADIUS attributes.
3. The DHCP server translates the RADIUS attributes into DHCP options, and sends this information back to RADIUS in a DHCP OFFER message.
4. DHCP binding is synchronized after the RADIUS server authorizes the client session.

If a local pool and an authorization pool are configured on the router, the DHCP server can assign addresses from both pools for different client interfaces.

DHCP Server RADIUS Proxy Architecture

The allocation of addresses in a DHCP and RADIUS solution occurs as follows:

1. The client accesses the network from a residential gateway and sends a DHCP DISCOVER broadcast message to the relay agent. The DHCP DISCOVER message contains the client IP address, hostname, vendor class identifier, and client identifier.
2. The relay agent sends a DHCP DISCOVER unicast message containing the following information to the router:
    - Relay agent information (option 82) with the remote ID suboption containing the inner and outer VLAN IDs
    - Client information in the DHCP DISCOVER packet

The router determines the address of the DHCP server from the IP helper address on the interface that receives the DHCP packet.

1. RADIUS receives an access-request message to translate the DHCP options to RADIUS attributes.
2. RADIUS responds with an access-accept message, and delivers the following attributes to the DHCP server:
    - Framed-IP-Address
    - Framed-IP-Netmask
    - Session-Timeout
    - Session-Duration
3. The DHCP server sends an OFFER unicast message containing the following translations from the RADIUS server access-accept message to the client:
    - Framed-IP-Address inserted into the DHCP header.
    - Framed-IP-Netmask inserted into DHCP option 1 (subnet mask).
    - Session-Timeout inserted into DHCP option 51 (IP address lease time).
    - Framed-Route that is translated from the standard Cisco Framed-Route format into DHCP option 121 or the DHCP default gateway option (if the network and netmask are appropriate for a default route).
    - A copy of relay agent information (option 82). Before the DHCP client receives the packet, the relay removes option 82.
    - T1 time set to the Session-Timeout and T2 time set to the Session-Duration.
4. The client returns a formal request for the offered IP address to the DHCP server in a DHCP REQUEST broadcast message.

5. The DHCP confirms that the IP address is allocated to the client by returning a DHCP ACK unicast message containing lease information and the DHCP options to the client.
6. A RADIUS server accounting request starts, followed by a RADIUS server accounting response that is used by the AAA subsystem.

When a RADIUS server attribute is not present in an access-accept message, the corresponding DHCP option is not sent to the DHCP client. If the required information to produce a particular RADIUS server attribute is not available to the DHCP server, the DHCP server does not include information in the RADIUS packet. Non-inclusion can be in the form of not sending an attribute (if there is no information at all), or omitting information from the attribute (in the case of CLI-based format strings).

If a DHCP option is provided to the DHCP server but is invalid, the DHCP server may not transmit the corresponding RADIUS attribute in the access-request, or may transmit an invalid RADIUS server attribute.

DHCP Server and RADIUS Translations

The table below lists the translations of DHCP options in a DHCP DISCOVER message to attributes in a RADIUS server access-request message.

Table 87: DCHP DISCOVER to RADIUS Access-Request Translations

DHCP DISCOVER	RADIUS Access-Request
Virtual MAC address of the residential gateway	User-Name
Not Applicable	User-Password as configured on the DHCP server
Gateway address of the relay agent (giaddr field of a DHCP packet)	NAS-identifier
Hostname	Cisco AV pair client-hostname that equals the value of DHCP option 12
Vendor class	Cisco AV pair dhcp-vendor-class that equals a hexadecimal-encoded value of DHCP option 60
Client identifier	Cisco AV pair dhcp-client-id that equals the hexadecimal-encoded value of DHCP option 61
DHCP relay information option that can contain VLAN parameter on the D-router	Cisco AV pair dhcp-relay-info that equals the hexadecimal-encoded value of DHCP option 82

The table below lists the translations of attributes in a RADIUS server access-accept message to DHCP options in a DHCP OFFER message.

Table 88: RADIUS Access-Accept to DHCP OFFER Translations

RADIUS Access-Accept	DHCP OFFER
Framed-IP-Address	IP address of the residential gateway
Framed-IP-Netmask	Subnet mask (option 1)

RADIUS Access-Accept	DHCP OFFER
Session-Timeout	IP address lease time (option 51)
Cisco AV pair session-duration in seconds, where seconds is greater than or equal to the number of seconds in the Session-Timeout attribute.	Provides session control on the DHCP server. This attribute is not transmitted to the DHCP client.
Framed-Route (RADIUS attribute 22). One route for each DHCP option is allowed with a maximum of 16 Framed-Route options for a RADIUS packet.	Contains up to 16 classless routes in one option (option 121)

RADIUS Profiles for DHCP Server RADIUS Proxy

When you configure RADIUS server user profiles for DHCP server RADIUS proxy, use the following guidelines:

- The Session-Timeout attribute must contain a value, in seconds. If this attribute is not present, the DHCP OFFER is not sent to the client.
- A RADIUS user profile must contain the following attributes:
  - Framed-IP-Address
  - Framed-IP-Netmask
  - Framed-Route
  - Session-Timeout
  - Session-Duration--Session-Duration is the Cisco AV pair session-duration = seconds, where seconds is the maximum time for the duration of a lease including all renewals. The value for Session-Duration must be greater than or equal to the Session-Timeout attribute value, and it cannot be zero.
- Additional RADIUS server attributes are allowed but are not required. The DHCP server ignores additional attributes that it does not understand. If a RADIUS server user profile contains a required attribute that is empty, the DHCP server does not generate the DHCP options.

How to Configure DHCP Server RADIUS Proxy

Configuring the DHCP Server for RADIUS-based Authorization

Perform this task on the DHCP server to configure address allocation for RADIUS-based authorization of DHCP leases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service dhcp**
4. aaa new-model
5. **aaa group server radius** *group-name*

6. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
7. **exit**
8. **aaa authorization network** *method-list-name* **group** *group-name*
9. **aaa accounting network** *method-list-name* **start-stop group** *group-name*
10. **ip dhcp pool** name
11. **accounting** *method-list-name*
12. **authorization method** *method-list-name*
13. **authorization shared-password** password
14. **authorization username** string
15. exit
16. **interface** *type slot* / *subslot* / *port* [**.** *subinterface*]
17. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[**,** *vlan-id*[- *vlan-id*]]}
18. **ip address** *address mask*
19. no shutdown
20. **radius-server host** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
21. **radius-server key** {*0 string* | *7 string* | *string*}
22. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**service dhcp** **Example:** `Router(config)# service dhcp`	Enables DHCP server and relay agent features on the router. By default, these features are enabled on the router.
Step 4	aaa new-model **Example:** `Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access control system.
Step 5	**aaa group server radius** *group-name* **Example:** `Router(config)# aaa group server radius group1`	Specifies the name of the server host list to group RADIUS server hosts. Enters server-group configuration mode. *group-name* --Character string to name the server group. The following words cannot be used as group name: • auth-guest

	Command or Action	Purpose
		• enable • guest • if-authenticated • if-needed • krb5 • krb-instance • krb-telnet • line • local • none • radius • rcmd • tacacs • tacacsplus
Step 6	**server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router(config-sg)# server 10.1.1.1 auth-port 1700 acct-port 1701`	*Specifies the IP address of the RADIUS server host for the defined server group. Repeat this command for each RADIUS server host to associate with the server group.* • *ip-address*-- IP address of the RADIUS server host. • **auth-port** *port-number*-- (Optional) Specifies the UDP destination port for authentication requests. Default value is 1645. • **acct-port** *port-number*-- (Optional) Specifies the UDP destination port for accounting requests. Default value is 1646.
Step 7	**exit** **Example:** `Router(config-sg)# exit`	Exits server-group configuration mode.
Step 8	**aaa authorization network** *method-list-name* **group** *group-name* **Example:** `Router(config)# aaa authorization network auth1 group group1`	Specifies the methods list and server group for DHCP authorization. • *method-list-name* --Character string to name the authorization methods list. • **group** --Specifies a server group.

	Command or Action	Purpose
		• *group-name* --Name of the server group to apply to DHCP authorization.
Step 9	**aaa accounting network** *method-list-name* **start-stop group** *group-name* **Example:** `Router(config)# aaa accounting network acct1 start-stop group group1`	Specifies that AAA accounting runs for all network service requests. • *method-list-name* --Character string to name the accounting methods list. • **start-stop** --Sends a start accounting notice at the beginning of a process and a stop accounting notice at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether or not the start accounting notice is received by the accounting server. • **group** --Specifies a server group. • *group-name* --Name of the server group to apply to DHCP accounting.
Step 10	**ip dhcp pool** name **Example:** `Router(config)# ip dhcp pool pool1`	Specifies a name for the DHCP server address pool. Enters DHCP pool configuration mode. • *name* --Name of the pool.
Step 11	**accounting** *method-list-name* **Example:** `Router(config-dhcp)# accounting acct1`	Enables DHCP accounting. • *method-list-name* --Name of the accounting methods list.
Step 12	**authorization method** *method-list-name* **Example:** `Router(config-dhcp)# authorization method auth1`	Enables DHCP authorization. • *method-list-name* --Name of the authorization methods list.
Step 13	**authorization shared-password** password **Example:** `Router(config-dhcp)# authorization shared-password cisco`	Specifies the password that is configured in the RADIUS user profile.

	Command or Action	Purpose
Step 14	**authorization username** string **Example:** `Router(config-dhcp)# authorization username %%c-user1`	Specifies the parameters that RADIUS sends to a DHCP server when downloading configuration information for a DHCP client. The *string* command argument contains the following formatting characters to insert DHCP client information: • **%c-** --Ethernet address of the DHCP client (chaddr field) • **%i-** --Inner VLAN ID from the DHCP relay information (option 82) • %o---Outer VLAN ID from the DHCP relay information (option 82) • **%p** --Port number from the DHCP relay information (option 82) • **%g** --Gateway address of the DHCP relay agent (giaddr field) • **%%** --Transmits the percent sign (%) character in the string sent to the RADIUS server **Note** The percent (%) is a marker to insert the DHCP client information associated with the specified character. The % is not sent to the RADIUS server unless you specify the %% character.
Step 15	exit **Example:** `Router(config-dhcp)# exit`	Exits DHCP pool configuration mode.
Step 16	**interface** *type slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface ethernet 1/10.0`	Configures an interface or subinterface that allows the DHCP client to obtain an IP address from the DHCP server. Enters interface or subinterface configuration mode.
Step 17	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[**,** *vlan-id*[**-** *vlan-id*]]} **Example:** `Router(config-subif)# encapsulation dot1q 100 second-dot1q 200`	(Optional) Enables IEEE 802.1Q encapsulation of traffic on a subinterface in a virtual LAN (VLAN). • *vlan-id* --VLAN ID, integer in the range 1 to 4094. To separate the starting and ending VLAN ID values that are used to define a range of VLAN IDs, enter a hyphen. (Optional) To separate each VLAN ID range from the next range, enter a comma. • second-dot1q--Supports the IEEE 802.1Q-in-Q VLAN Tag Termination feature to configure an inner VLAN ID.

	Command or Action	Purpose
		• **any** --Any second tag in the range 1 to 4094.
Step 18	**ip address** *address mask* **Example:** `Router(config-if)# ip address 192.168.1.1 255.255.255.0`	Specifies an IP address for an interface or subinterface. • *address* is the IP address of the interface or subinterface. • *mask* is the subnet address for the IP address.
Step 19	no shutdown **Example:** `Router(config-if)# no shutdown`	Enables the interface or subinterface.
Step 20	**radius-server host** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router(config)# radius-server host 10.1.1.1`	Specifies a RADIUS server host. • *ip-address* is the IP address of the RADIUS server host. • **auth-port** *port-number*-- (Optional) Specifies the UDP destination port for authentication requests. Default value is 1645. • **acct-port** *port-number*-- (Optional) Specifies the UDP destination port for accounting requests. Default value is 1646.
Step 21	**radius-server key** {*0 string* | *7 string* | *string*} **Example:** `Router(config)# radius-server key cisco`	Specifies the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. • **0** *string*-- Specifies an unencrypted (cleartext) shared key • **7** *string* -- Specifies a hidden shared key. **Note** Any key you enter must match the key on the RADIUS daemon. All leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 22	**exit**	Exits global configuration mode.

Monitoring and Maintaining the DHCP Server

Perform this task to verify and monitor DHCP server information:

SUMMARY STEPS

1. **enable**
2. **debug ip dhcp server packet**
3. **debug ip dhcp server events**
4. **show ip dhcp binding** [*address*]
5. **show ip dhcp server statistics**
6. **show ip dhcp pool** [*name*]
7. **show ip route dhcp** [*address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip dhcp server packet** **Example:** `Router# debug ip dhcp server packet`	(Optional) Enables DHCP server debugging.
Step 3	**debug ip dhcp server events** **Example:** `Router# debug ip dhcp server events`	(Optional) Reports DHCP server events, such as address assignments and database updates.
Step 4	**show ip dhcp binding** [*address*] **Example:** `Router# show ip dhcp binding`	(Optional) Displays a list of all bindings created on a specific DHCP server. • Use the **show ip dhcp binding** command to display the IP addresses that have already been assigned. Verify that the address pool has not been exhausted. If necessary, re-create the pool to create a larger pool of addresses. • Use the **show ip dhcp binding** command to display the lease expiration date and time of the IP address of the host.
Step 5	**show ip dhcp server statistics** **Example:** `Router# show ip dhcp server statistics`	(Optional) Displays count information about server statistics and messages sent and received.
Step 6	**show ip dhcp pool** [*name*] **Example:**	(Optional) Displays the routes added to the routing table by the DHCP server and relay agent.

	Command or Action	Purpose
	`Router# show ip dhcp pool`	
Step 7	**show ip route dhcp** [*address*] **Example:** `Router# show ip route dhcp [address]`	(Optional) Displays information about DHCP address pools.

Configuration Examples for DHCP Server Radius Proxy

Configuring the DHCP Server Example

The following example shows how to configure a DHCP server for RADIUS-based authorization of DHCP leases. In this example, DHCP clients can attach to Ethernet interface 4/0/1 and Ethernet subinterface 4/0/3.10. The username string (%c-user1) specifies that the RADIUS server sends the Ethernet address of DHCP client named user1 to the DHCP server.

```
Router> enable
Router# configure terminal
Router(config)# service dhcp
Router(config)# aaa new-model
Router(config)# aaa group server radius rad1
Router(config-sg)# server 10.1.1.1
Router(config-sg)# server 10.1.5.10
Router(config-sg)# exit
Router(config)# aaa authorization network auth1 group group1
Router(config)# aaa accounting network acct1 start-stop group group1
Router(config)# aaa session-id common
Router(config)# ip dhcp database tftp://172.16.1.1/router-dhcp write-delay 100 timeout 5
!
Router(config)# ip dhcp pool pool_common
Router(config-dhcp)# accounting acct1
Router(config-dhcp)# authorization method auth1
Router(config-dhcp)# authorization shared-password cisco
Router(config-dhcp)# authorization username %c-user1
Router(config-dhcp)# exit
!
Router(config)# interface ethernet4/0/1
Router(config-if)# ip address 15.0.0.1 255.255.255.0
Router(config-if)# exit
Router(config-if)# interface ethernet4/0/3.10

Router(config-if)# encapsulation dot1q 100 second-dot1q 200
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# radius-server host 10.1.3.2
Router(config)# radius-server key cisco
Router(config)# exit
```

Configuring RADIUS Profiles Example

The following example shows how to configure a typical RADIUS user profile to send attributes in an access-accept message to the DHCP server:

```
DHCP-00059A3C7800 Password = “metta”
Service-Type = Framed,
Framed-Ip-Address = 10.3.4.5,
Framed-Netmask = 255.255.255.0,
Framed-Route = "0.0.0.0 0.0.0.0 10.3.4.1",
Session-Timeout = 3600,
Cisco:Cisco-Avpair = "session-duration=7200”
```

Additional References

The following sections provide references related to the DHCP Server RADIUS Proxy feature.

Related Documents

Related Topic	Document Title
DHCP relay configuration	*Configuring the Cisco IOS XE DHCP Relay Agent*
DHCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this functionality.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs was not modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for DHCP Server RADIUS Proxy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 89: Feature Information for the Cisco IOS XE DHCP Relay Agent

Feature Name	Releases	Feature Configuration Information
DHCP Server RADIUS Proxy	Cisco IOS XE Release 2.4 Cisco IOS XE Release 3.9S	DHCP Server RADIUS Proxy enables a server to authorize remote clients and allocate addresses based on replies from the server. In Cisco IOS XE 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified by this feature: **authorization method (dhcp)**, **authorization shared-password, authorization username (dhcp)**.

Glossary

client --A host trying to configure its interface (obtain an IP address) using DHCP or BOOTP protocols.

DHCP --Dynamic Host Configuration Protocol.

giaddr --Gateway IP address. The giaddr field of the DHCP message provides the DHCP server with information about the IP address subnet on which the client is to reside. It also provides the DHCP server with an IP address where the response messages are to be sent.

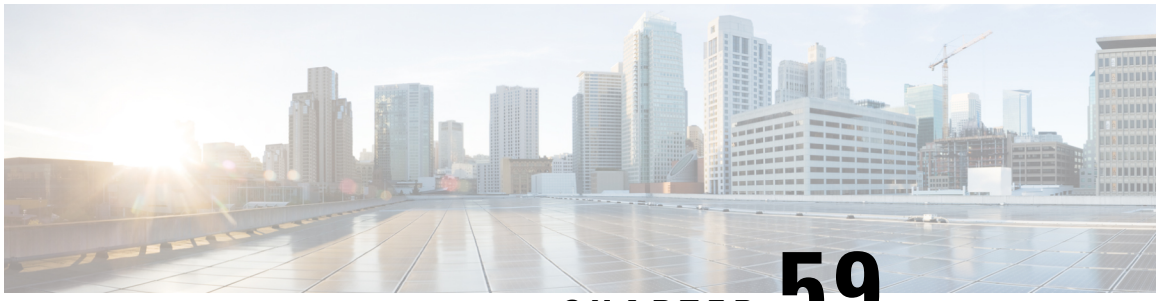
MPLS --Multiprotocol Label Switching. Emerging industry standard upon which tag switching is based.

relay agent --A router that forwards DHCP and BOOTP messages between a server and a client on different subnets.

server --DHCP or BOOTP server.

VPN --Virtual Private Network. Enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

VRF --VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router. Each VPN instantiated on the PE router has its own VRF.



CHAPTER 59

Configuring the Cisco IOS XE DHCP Client

Cisco IOS XE Dynamic Host Configuration Protocol (DHCP) client software provides the flexibility to include various configuration options for the DHCP client. A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address. This module describes the concepts and tasks needed to configure the Cisco IOS XE DHCP client.

Feature Information for the Cisco IOS XE DHCP Client

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 90: Feature Information for the Cisco IOS XE DHCP Client

Feature Name	Releases	Feature Configuration Information
DHCP Client	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address. The following command was introduced by this feature: **ip address dhcp**

Feature Name	Releases	Feature Configuration Information
Configurable DHCP Client	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	The configurable DHCP client functionality allows a DHCP client to use a user-specified client identifier, class identifier, or suggested lease time when requesting an address from a DHCP server. The following commands were introduced by this feature: **ip dhcp client class-id**, **ip dhcp client client-id**, **ip dhcp client hostname**, **ip dhcp client lease**, **ip dhcp client request**
DHCPv4 Client Options	Cisco IOS XE Fuji 16.9.1	The DHCP Client supports configuration of all 1-254 options.
DHCP Client Options using unicast mode	Cisco IOS XE Amsterdam 17.2.1	Introduces support for unicast mode on DHCP. This helps with splitting the horizon therefore improving security of the network.

Information About the DHCP Client

DHCP Client Operation

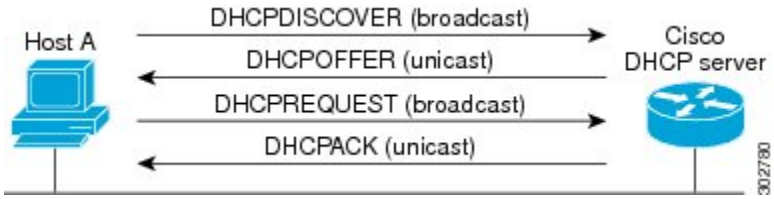
When a Dynamic Host Configuration Protocol (DHCP) client requests an IP address from a DHCP server on a Cisco IOS XE platform, the default process includes:

- DHCPDISCOVERY (broadcast)
- DHCPOFFER (broadcast)
- DHCPREQUEST (broadcast)
- DHCPACK (unicast)

The DHCP on Cisco IOS XE platform supports only broadcast mode with the DHCPOFFER. From Cisco IOS XE Amsterdam Release 17.2, the DHCP on IOS XE platform also supports unicast mode. The DHCP unicast mode helps to split the horizon for security consideration. The DHCP broadcast mode is enabled by default. To enable the DHCP unicast mode, configure the **ip dhcp client broadcast-flag clear** command on the DHCP client. After configuring the command, the DHCPOFFER is sent as a unicast message.

The DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. A DHCP client is an Internet host using DHCP to obtain configuration parameters such as an IP address. The following figure shows the basic steps that occur when a DHCP client requests an IP address from a DHCP server. The client, Host A, sends a DHCPDISCOVER broadcast message to locate a DHCP server. A DHCP server offers configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a DHCPOFFER unicast/broadcast message.

Figure 64: DHCP Request for an IP Address from a DHCP Server



A DHCP client may receive offers from multiple DHCP servers. However, it can accept any one of the offers; the client usually accepts the first offer it receives. The offer from the DHCP server is not a guarantee that the IP address is allocated to the client. However, the server usually reserves the address until the client has had a chance to formally request the address .

The client returns a formal request for the offered IP address to the DHCP server in a DHCPREQUEST broadcast message. The DHCP server confirms that the IP address is allocated to the client by returning a DHCPACK unicast message to the client.

DHCP Client Overview

The configurable dynamic host configuration protocol client functionality allows a DHCP client to use a user-specified client identifier, class identifier, or suggested lease time when requesting an address from a DHCP server.

Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 12—This option specifies the name of the client. The name may or may not be qualified with the local domain.
- Option 51—This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address.
- Option 55—This option allows the DHCP client to request certain options from the DHCP server. The **ip dhcp client request** command allows the system administrator to turn off some of the requested options, thus removing them from the request list.
- Option 60—This option allows the user to configure the vendor class identifier string to use in the DHCP interaction.
- Option 61—This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.
- Option 77—This option is used by a DHCP clients to optionally identify the type or category of user or applications it represents. The information contained in this option represents the user class of which the client is a member. Based on this class, a DHCP server selects the appropriate address pool to assign an address to the client and the appropriate configuration parameters.
- Option 120—This option is used to specify a 32-bit (binary) IPv4 address to be used by the Session Initiation Protocol (SIP) client to locate a SIP server.

- Option 121—This option is used to configure classless static routes by specifying classless network destinations; that is, each routing table entry includes a subnet mask. Upto ten classless static routes are supported using option 121 on the DHCP client.

Note If a request includes both static routes and classless static routes, the client uses only the classless static routes. If the DHCP server returns both a classless static route option and a router option, the DHCP client ignores the router option.

- Option 124—This option is used by DHCP clients and servers to exchange vendor-class information.
- Option 125—This option is used by DHCP clients and servers to exchange vendor-specific information.

How to Configure the DHCP Client

Configuring the DHCP Client

Cisco devices running Cisco software include the Dynamic Host Configuration Protocol (DHCP) server and relay agent software, which are enabled by default. Your device can act as both the DHCP client and the DHCP server. Use the **ip address dhcp** command to obtain IP address information for the configured interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address dhcp**
5. **end**
6. **debug dhcp detail**
7. **debug ip dhcp server packets**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address dhcp** **Example:** `Device(config-if)# ip address dhcp`	Acquires an IP address on an interface from DHCP.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**debug dhcp detail** **Example:** `Device# debug dhcp detail`	Displays the DHCP packets that were sent and received.
Step 7	**debug ip dhcp server packets** **Example:** `Device# debug ip dhcp server packets`	Displays the server side of the DHCP interaction.

Troubleshooting Tips

To verify the configuration, you can use the **debug dhcp detail** EXEC command to display the DHCP packets that were sent and received. To display the server side of the DHCP interaction, use the **debug ip dhcp server packets** command.

Configure Administrative Distance

To configure the default Dynamic Host Configuration Protocol (DHCP) Administrative Distance (AD), use the **ip dhcp client default-router distance** command in interface configuration or global configuration mode:

Configuration in Interface Configuration Mode:

When you use this command for a interface, AD is applied to the route received in DHCP process of that particular interface.

```
Router # configure terminal
Router(config)# interface FastEthernet 0/2
Router(config-if)# ip dhcp client default-router distance 2
```

Configuration in Global Configuration Mode:

Use this command to configure AD is on the route received in DHCP process of any interface.

```
Router # configure terminal
Router(config)#ip dhcp-client default-router distance 10
```

```
Router(config)#interface e0/0
Router(config-if)#ip address dhcp
Router(config-if)#no shut
Router(config-if)#end
```

To disable the configuration, use the **no** form of this command.

Note When you install the **ip dhcp client default-router distance** command in interface configuration or global configuration mode, default route given by DHCP is installed with specified AD. But, when you use the same command with different AD value, it does not take effect immediately. It takes effect when a new connection is made or when a new Discover-Offer-Request-Ack (DORA) cycle happens. It can be done in any one of the following ways:

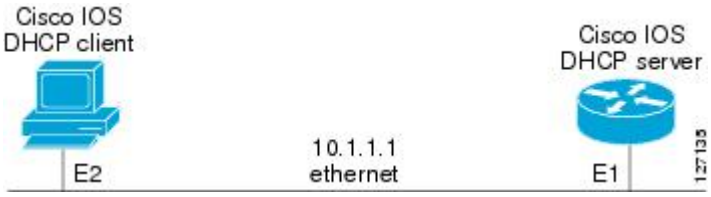
- Bounce the interface, which means execute shutdown followed by no shutdown of the interface.
- Release (using **release dhcp** *interface*) command and renew DHCP (using **renew dhcp** *interface*) command from exec mode.

Configuration Examples for the DHCP Client

Configuring the DHCP Client Example

The figure below shows a simple network diagram of a DHCP client on an Ethernet LAN.

Figure 65: Topology Showing DHCP Client with GigabitEthernet Interface



On the DHCP server, the configuration is as follows:

```
ip dhcp pool 1
 network 10.1.1.0 255.255.255.0
 lease 1 6
```

On the DHCP client, the configuration is as follows on interface GigabitEthernet 0/0/0:

```
interface GigabitEthernet 0/0/0
 ip address dhcp
```

This configuration allows the DHCP client to acquire an IP address from the DHCP server through GigabitEthernet interface 0/0/0.

Customizing the DHCP Client Configuration Example

The following example shows how to customize the DHCP client configuration with various options on GigabitEthernet interface 0/0/1:

```
interface GigabitEthernet 0/0/1
 ip dhcp client client-id ascii my-test1
 ip dhcp client class-id my-class-id
 ip dhcp client lease 0 1 0
 ip dhcp client hostname sanfran
 no ip dhcp client request tftp-server-address
 ip address dhcp
```

The following example shows DHCP Client configuration on GigabitEthernet 0/0/1 to generically request options:

```
!
interface GigabitEthernet 0/0/1
ip dhcp client request option 4 5 7 8 9 10 11 17 18 40 41 42 66 68 69 70 71 72 73 74 75 76
 124 138 141 142 160
no ip address
shutdown
!
```

The following example shows how to configure DHCP Client options with parameters, IP address and string:

```
!
interface GigabitEthernet 0/0/1
ip dhcp client option 1 ip 10.0.0.1
ip dhcp client option 13 ascii test13
ip dhcp client option 14 ascii test14
ip dhcp client option 16 ip 10.0.0.16
ip dhcp client option 46 ascii test46
ip dhcp client option 47 ascii test47
ip dhcp client option 50 ip 10.0.0.50
ip dhcp client option 51 ascii test51
ip dhcp client option 52 ascii test52
ip dhcp client option 54 ascii test54
ip dhcp client option 58 ascii test58
ip dhcp client option 59 ascii test59
ip dhcp client option 60 ascii test60
ip dhcp client option 61 ascii test61
ip dhcp client option 62 ascii test62
ip dhcp client option 63 ip 10.0.0.63
ip dhcp client option 64 ascii test64
ip dhcp client option 65 ip 10.0.0.65
ip dhcp client option 67 ascii test67
ip dhcp client option 90 ascii test90
ip dhcp client option 116 ascii test116
ip dhcp client option 118 ip 10.0.0.118
ip dhcp client option 220 ip 10.0.0.220
ip dhcp client option 221 ascii test221
ip address dhcp
shutdown
!
```

The following example shows how to configure DHCP Client options with class:

By default, Option-124 carries the PID of the device. The following example shows the overriding option-124 default value in DHCPv4:

```
Router(config-if)# ip dhcp client ?
information Configure information refresh option
pd Prefix-Delegation
request Request
vendor-class Configure vendor class data, Product ID by default (Option 124)
```

The following configuration example overrides PID with mac-address:

```
Router(config-if)# ip dhcp client vendor-class mac-address
```

The following configuration example overrides PID with user defined string in ascii format:

```
Router(config-if)# ip dhcp client vendor-class ascii cisco
```

The following configuration example overrides PID with user defined string in hex format:

```
Router(config-if)# ip dhcp client vendor-class hex aabbcc
```

The following configuration example is used to disable sending option-124 in DHCPv4 messages:

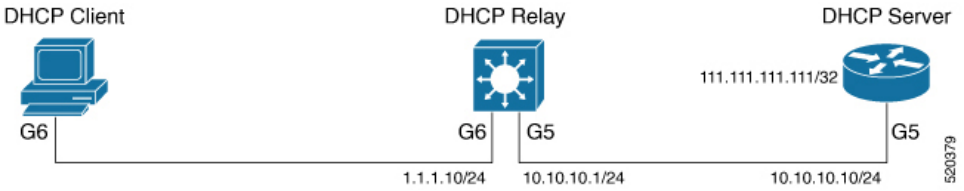
```
Router(config-if)# ip dhcp client vendor-class disable
```

Example: Configuring the DHCP Client in Unicast Mode

The following example shows how to configure DHCP Client in unicast mode:

The figure below shows a simple network diagram of a DHCP client in unicast mode.

Figure 66: Topology Showing DHCP Client with GigabitEthernet Interface



Client:

```
interface GigabitEthernet6
ip address dhcp
ip dhcp client broadcast-flag clear
```

Relay:

```
interface GigabitEthernet6
ip address 1.1.1.10 255.255.255.0
ip helper-address 111.111.111.111

!
interface GigabitEthernet5
  ip address 10.10.10.1 255.255.255.0

ip route 111.111.111.111 255.255.255.255 GigabitEthernet5
```

Server:

```
interface Loopback10
ip address 111.111.111.111 255.255.255.255
no shutdown
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet5

ip dhcp pool Cisco
```

```
network 11.11.11.0 255.255.255.0
default-router 11.11.11.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples.	Cisco IOS IP Addressing Services Command Reference
DHCP conceptual information	"DHCP Overview" module
DHCP server configuration	"Configuring the Cisco IOS XE DHCP Server" module
DHCP server on-demand address pools	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP relay agent configuration	"Configuring the Cisco IOS XE DHCP Relay Agent" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module

RFCs

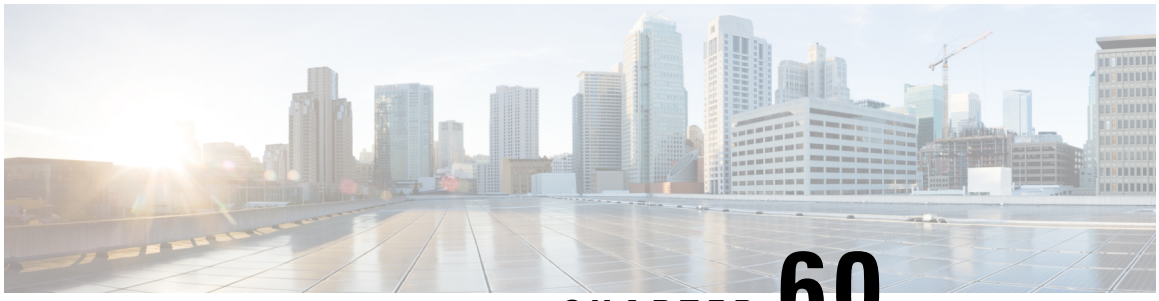
RFCs	Title
RFC 2131	Dynamic Host Configuration Protocol
RFC 2132	DHCP Options and BOOTP Vendor Extensions

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport



CHAPTER 60

Configuring DHCP Services for Accounting and Security

Cisco IOS XE software supports several capabilities that enhance DHCP security, reliability, and accounting in Public Wireless LANs (PWLANs). This functionality can also be used in other network implementations. This module describes the concepts and tasks needed to configure DHCP services for accounting and security.

Prerequisites for Configuring DHCP Services for Accounting and Security

Before you configure DHCP services for accounting and security, you should understand the concepts documented in the “DHCP Overview” module.

Information About DHCP Services for Accounting and Security

DHCP Operation in Public Wireless LANs

The configuration of DHCP in a public wireless LAN (PWLAN) simplifies the configuration of wireless clients and reduces the overhead necessary to maintain the network. DHCP clients are leased IP addresses by the DHCP server and then authenticated by the Service Selection Gateway (SSG), which allows the clients to access network services. The DHCP server and client exchange DHCP messages for IP address assignments. When a DHCP server assigns an IP address to a client, a DHCP binding is created. The IP address is leased to the client until the client explicitly releases the IP address and disconnects from the network. If the client disconnects without releasing the address, the server terminates the lease after the lease time is over. In either case, the DHCP server removes the binding and the IP address is returned to the pool.

Security Vulnerabilities in Public Wireless LANs

As more people start using PWLANs, security becomes an important concern. Most implementations of PWLANs rely on DHCP for users to obtain an IP address while in a hot spot (such as a coffee shop, airport terminal, hotel, and so on) and use this IP address provided by the DHCP server throughout their session.

IP spoofing is a common technique used by hackers to spoof IP addresses. For example, customer A obtains an IP address from DHCP and has already been authenticated to use the PWLAN, but a hacker spoofs the IP address of customer A and uses this IP address to send and receive traffic. Customer A will still be billed for the service even though he or she is not using the service.

Address Resolution Protocol (ARP) table entries are dynamic by design. Request and reply ARP packets are sent and received by all the networking devices in a network. In a DHCP network, the DHCP server stores the leased IP address to the MAC address or the client-identifier of the client in the DHCP binding. But as ARP entries are learned dynamically, an unauthorized client can spoof the IP address given by the DHCP server and start using that IP address. The MAC address of this unauthorized client will replace the MAC address of the authorized client in the ARP table allowing the unauthorized client to freely use the spoofed IP address.

DHCP Services for Security and Accounting Overview

DHCP security and accounting features have been designed and implemented to address the security concerns in PWLANs but also can be used in other network implementations.

DHCP accounting provides authentication, authorization, and accounting (AAA) and Remote Authentication Dial-In User Service (RADIUS) support for DHCP. The AAA and RADIUS support improves security by sending secure START and STOP accounting messages. The configuration of DHCP accounting adds a layer of security that allows DHCP lease assignment and termination to be triggered for the appropriate RADIUS START and STOP accounting records so that the session state is properly maintained by upstream devices, such as an SSG. This additional security can help to prevent unauthorized clients or hackers from gaining illegal entry to the network by spoofing authorized DHCP leases.

The DHCP Secured IP Address Assignment feature prevents IP spoofing by synchronizing the database of the DHCP server with the ARP table to avoid address hijacking. This secure ARP functionality adds an entry to the ARP table for a client when an address is allocated that can be deleted by the DHCP server only when a binding expires.

The third feature is ARP autologoff, which adds finer control for probing when authorized users log out. The **arp probe interval** command specifies when to start a probe (the timeout), how frequent a peer is probed (the interval), and the maximum number of retries (the count).

DHCP Lease Limits

You can control the number of subscribers globally or on a per-interface basis by configuring a DHCP lease limit. This functionality allows an Internet service provider (ISP) to limit the number of leases available to clients per household or connection.

How to Configure DHCP Services for Accounting and Security

Configuring AAA and RADIUS for DHCP Accounting

Perform this task to configure AAA and RADIUS for DHCP accounting.

RADIUS provides the accounting capability for the transmission of secure START and STOP messages. AAA and RADIUS are enabled prior to the configuration of DHCP accounting but can also be enabled to secure an insecure DHCP network. The configuration steps in this section are required for configuring DHCP accounting in a new or existing network.

DHCP accounting introduces the attributes shown in the table below. These attributes are processed directly by the RADIUS server when DHCP accounting is enabled. These attributes can be monitored in the output of the debug radius command. The output will show the status of the DHCP leases and specific configuration details about the client. The accounting keyword can be used with the debug radius command to filter the output and display only DHCP accounting messages.

Table 91: RADIUS Accounting Attributes

Attribute	Description
Calling-Station-ID	The output from this attribute displays the MAC address of the client.
Framed-IP-Address	The output from this attribute displays the IP address that is leased to the client.
Acct-Terminate-Cause	The output from this attribute displays the message “session-timeout” if a client does not explicitly disconnect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *group-name*
5. **server** *ip-address* **auth-port** *port-number* **acct-port** *port-number*
6. **exit**
7. **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** group-name
8. **aaa session-id** {**common** | **unique**}
9. **ip radius source-interface** *type number* [**vrf** *vrf-name*]
10. **radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*]
11. **radius-server attribute 31 send nas-port-detail mac-only**
12. **radius-server retransmit number-of-retries**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the AAA access control model. • DHCP accounting functions only in the access control model. **Note** TACACS and extended TACACS commands are not available after this command is configured and are not supported by DHCP accounting.
Step 4	**aaa group server radius** *group-name* **Example:** `Device(config)# aaa group server radius RGROUP-1`	Creates a server group for AAA or TACACS+ services and enters server group configuration mode. • The server group is created in this step so that accounting services can be applied.
Step 5	**server** *ip-address* **auth-port** *port-number* **acct-port** *port-number* **Example:** `Device(config-sg-radius)# server 10.0.0.1 auth-port 1645 acct-port 1646`	Specifies the servers that are members of the server group that was created in Step 4. • You must open port numbers for authorization and accounting. 1645 is the default port number for authorization, and 1646 is the default port number for accounting. The range of port numbers that can be specified is from 0 to 65535. • The values entered for the auth-port port-number and acct-port port-number keywords and arguments must match the port numbers that will be configured in Step 10.
Step 6	**exit** **Example:** `Device(config-sg-radius)# exit`	Exits server group configuration mode and enters global configuration mode.
Step 7	**aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** group-name **Example:** `Device(config)# aaa accounting network RADIUS-GROUP1 start-stop group RGROUP-1`	Configures RADIUS accounting for the specified server group. • The RADIUS accounting server is specified in the first list-name argument (RADIUS-GROUP1), and the target server group is specified in the second group-name argument (RGROUP-1).

	Command or Action	Purpose
		• This command enables start and stop accounting for DHCP accounting. The start-stop keyword enables the transmission of both START and STOP accounting messages. The stop-only keyword will enable the generation and verification of STOP accounting messages only.
Step 8	**aaa session-id** {**common** | **unique**} **Example:** `Device(config)# aaa session-id common`	Specifies whether the same session ID will be used for each AAA accounting service type within a call or whether a different session ID will be assigned to each accounting service type.
Step 9	**ip radius source-interface** *type number* [**vrf** *vrf-name*] **Example:** `Device(config)# ip radius source-interface GigabitEthernet 0/0/0`	Forces RADIUS to use the IP address of the specified interface for all outgoing RADIUS packets.
Step 10	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Device(config)# radius-server host 10.1.1.1 auth-port 1645 acct-port 1646`	Specifies the radius server host. • The values entered for the auth-port port-number and acct-port port-number keywords and arguments must match the port numbers that were configured in Step 5.
Step 11	**radius-server attribute 31 send nas-port-detail mac-only** **Example:** `Device(config)# radius-server attribute 31 send nas-port-detail mac-only`	(Optional) Allows the MAC address of the client to be included in the Calling-Station-ID attribute. The Calling-Station-ID attribute is processed by the RADIUS server when DHCP accounting is enabled.
Step 12	**radius-server retransmit number-of-retries** **Example:** `Device(config)# radius-server retransmit 3`	Specifies the number of times that Cisco IOS XE software will look for RADIUS server hosts.

Troubleshooting Tips

To monitor and troubleshoot the configuration of RADIUS accounting, use the following command:

Command	Purpose
debug radius accounting `Device# debug radius accounting`	The **debug radius** command is used to display RADIUS events on the console of the device. These events provide information about RADIUS processes. DHCP accounting information can be filtered with the accounting keyword. START and STOP accounting message information will also be displayed.

Configuring DHCP Accounting

Perform this task to configure DHCP accounting.

DHCP accounting is enabled with the **accounting** command. This command configures DHCP to operate with AAA and RADIUS to enable secure START and STOP accounting messages. This configuration adds a layer of security that allows DHCP lease assignment and termination to be triggered for the appropriate RADIUS START and STOP accounting records so that the session state is properly maintained by upstream devices, such as the SSG.

DHCP accounting is configured on a per-client or per-lease basis. Separate DHCP accounting processes can be configured on a per-pool basis.

Before you begin

You must configure an SSG for client authentication. AAA and RADIUS must be enabled before DHCP accounting will operate.

Note The following restrictions apply to DHCP accounting:

- DHCP accounting can be configured only for DHCP network pools in which bindings are created automatically and destroyed upon lease termination or when the client sends a DHCPRELEASE message.
- DHCP bindings are destroyed when the **clear ip dhcp binding** or **no service dhcp** commands are entered, which also triggers an accounting STOP message. You should exercise caution when entering these commands if a pool is configured with DHCP accounting, as these commands will clear active leases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **accounting** *method-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:**	Configures a DHCP address pool and enters DHCP pool configuration mode.

	Command or Action	Purpose
	`Device(config)# ip dhcp pool WIRELESS-POOL`	
Step 4	**accounting** *method-list-name* **Example:** `Device(dhcp-config)# accounting RADIUS-GROUP1`	Enables DHCP accounting if the specified server group is configured to run RADIUS accounting. • The example configures DHCP accounting START and STOP messages to be sent if RADIUS-GROUP1 is configured as a start-stop group. STOP messages will only be sent if RADIUS-GROUP1 is configured as a stop-only group. See Step 7 in the "Configuring AAA and RADIUS for DHCP Accounting" configuration task table for more details.

Verifying DHCP Accounting

Perform this task to verify the DHCP accounting configuration.

The debug radius, debug ip dhcp server events, debug aaa accounting, debug aaa id commands do not need to be issued together or in the same session as there are differences in the information that is provided. These commands, however, can be used to display DHCP accounting start and stop events, AAA accounting messages, and information about AAA and DHCP hosts and clients. See the "RADIUS Accounting Attributes" section of this module for a list of AAA attributes that have been introduced by DHCP accounting. The show running-config | begin dhcp command can be used to display the local DHCP configuration including the configuration of DHCP accounting.

SUMMARY STEPS

1. **enable**
2. **debug radius accounting**
3. **debug ip dhcp server events**
4. **debug aaa accounting**
5. **debug aaa id**
6. **show running-config | begin dhcp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius accounting** **Example:** `Device# debug radius accounting`	Displays RADIUS events on the console of the device. • These events provide information about RADIUS processes. DHCP accounting information can be filtered with the accounting keyword. START and

	Command or Action	Purpose
		STOP accounting messages will be displayed in the output.
Step 3	**debug ip dhcp server events** **Example:** `Device# debug ip dhcp server events`	Displays DHCP IP address assignments, DHCP lease expirations, and DHCP database changes.
Step 4	**debug aaa accounting** **Example:** `Device# debug aaa accounting`	Displays AAA accounting events. • START and STOP accounting messages will be displayed in the output.
Step 5	**debug aaa id** **Example:** `Device# debug aaa id`	Displays AAA events as they relate to unique AAA session IDs.
Step 6	**show running-config** | **begin dhcp** **Example:** `Device# show running-config | begin dhcp`	The **show running-config** command is used to display the local configuration of the device. The sample output is filtered with the **begin** keyword to start displaying output at the DHCP section of the running configuration.

Securing ARP Table Entries to DHCP Leases

Perform this task to secure ARP table entries to DHCP leases in the DHCP database.

When the **update arp** command is used, ARP table entries and their corresponding DHCP leases are secured automatically for all new leases and DHCP bindings. However, existing active leases are not secured. These leases are still insecure until they are renewed. When the lease is renewed, it is treated as a new lease and will be secured automatically. If this command is disabled on the DHCP server, all existing secured ARP table entries will automatically change to dynamic ARP entries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool -name*
4. **update arp**
5. **renew deny unknown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool -name* **Example:** `Device(config)# ip dhcp pool WIRELESS-POOL`	Configures a DHCP address pool and enters DHCP pool configuration mode.
Step 4	**update arp** **Example:** `Device(dhcp-config)# update arp`	Secures insecure ARP table entries to the corresponding DHCP leases. • Existing active DHCP leases will not be secured until they are renewed. Using the **no update arp** command will change secured ARP table entries back to dynamic ARP table entries.
Step 5	**renew deny unknown** **Example:** `Device(dhcp-config)# renew deny unknown`	(Optional) Configures the renewal policy for unknown clients. • See the Troubleshooting Tips, on page 805 section for information about when to use this command.

Troubleshooting Tips

Use the **debug ip dhcp server class** command to display the class matching results.

Configuring a DHCP Lease Limit to Control the Number of Subscribers on an Interface

Perform this task to limit the number of DHCP leases allowed on an interface.

This feature allows an ISP to limit the number of leases available to clients per household or connection on an interface.

If this feature is enabled on the Cisco IOS XE DHCP server directly connected to clients through unnumbered interfaces, the server allocates addresses and increments the number of leases per subinterface. If a new client tries to obtain an IP address, the server will not offer an IP address if the number of leases on the subinterface has already reached the configured lease limit.

Note This feature is not supported on numbered interfaces. The lease limit can be applied only to ATM with RBE unnumbered interfaces or serial unnumbered interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp limit lease log**
4. **interface** *type number*
5. **ip dhcp limit lease** *lease-limit*
6. **end**
7. **show ip dhcp limit lease** [*type number*]
8. **show ip dhcp server statistics** [*type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp limit lease log** **Example:** `Device(config)# ip dhcp limit lease log`	(Optional) Enables DHCP lease violation logging when a DHCP lease limit threshold is exceeded. • If this command is configured, any lease limit violations will display in the output of the **show ip dhcp limit lease** command.
Step 4	**interface** *type number* **Example:** `Device(config)# interface Serial0/0/0`	Enters interface configuration mode.
Step 5	**ip dhcp limit lease** *lease-limit* **Example:** `Device(config-if)# ip dhcp limit lease 6`	Limits the number of leases offered to DHCP clients per interface. • The interface configuration will override any global setting specified by the **ip dhcp limit lease per interface** global configuration command.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits the configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show ip dhcp limit lease** [*type number*] **Example:** `Device# show ip dhcp limit lease Serial0/0/0`	(Optional) Displays the number of times the lease limit threshold has been violated. • You can use the **clear ip dhcp limit lease** privileged EXEC command to manually clear the stored lease violation entries.
Step 8	**show ip dhcp server statistics** [*type number*] **Example:** `Device# show ip dhcp server statistics Serial 0/0/0`	(Optional) Displays DHCP server statistics.

Troubleshooting Tips

You can use the **debug ip dhcp server packet** and **debug ip server events** commands to troubleshoot the DHCP lease limit.

Configuration Examples for DHCP Services for Accounting and Security

Example: Configuring AAA and RADIUS for DHCP Accounting

The following example shows how to configure AAA and RADIUS for DHCP accounting:

```
aaa new-model
aaa group server radius RGROUP-1
 server 10.1.1.1 auth-port 1645 acct-port 1646
 exit
aaa accounting network RADIUS-GROUP1 start-stop group RGROUP-1
aaa session-id common
ip radius source-interface GigabitEthernet0/0/0
radius-server host 10.1.1.1 auth-port 1645 acct-port 1646
radius-server attribute 31 send nas-port-detail mac-only
radius-server retransmit 3
exit
```

Example: Configuring DHCP Accounting

DHCP accounting is configured on a per-client or per-lease basis. Separate DHCP accounting processes can be configured on a per-pool basis. The following example shows how to configure DHCP accounting START and STOP messages to be sent if RADIUS-GROUP1 is configured as a start-stop group.

```
ip dhcp pool WIRELESS-POOL
 accounting RADIUS-GROUP1
 exit
```

Example: Verifying DHCP Accounting

DHCP accounting is enabled after both RADIUS and AAA for DHCP are configured. DHCP START and STOP accounting generation information can be monitored with the **debug radius accounting** and **debug ip dhcp server events** commands. See the "RADIUS Accounting Attributes" section of this module for a list of AAA attributes that have been introduced by DHCP accounting.

The following is sample output from the **debug radius accounting** command. The output shows the DHCP lease session ID, the MAC address, and the IP address of the client interface.

```
00:00:53: RADIUS: Pick NAS IP for uid=2 tableid=0 cfg_addr=10.0.18.3 best_addr=0.0.0.0
00:00:53: RADIUS(00000002): sending
00:00:53: RADIUS(00000002): Send to unknown id 21645/1 10.1.1.1 :1646, Accounting-Request,
 len 76
00:00:53: RADIUS: authenticator C6 FE EA B2 1F 9A 85 A2 - 9A 5B 09 B5 36 B5 B9 27
00:00:53: RADIUS: Acct-Session-Id [44] 10 "00000002"
00:00:53: RADIUS: Framed-IP-Address [8] 6 10.0.0.10
00:00:53: RADIUS: Calling-Station-Id [31] 16 "00000c59df76"
00:00:53: RADIUS: Acct-Status-Type [40] 6 Start [1]
00:00:53: RADIUS: Service-Type [6] 6 Framed [2]
00:00:53: RADIUS: NAS-IP-Address [4] 6 10.0.18.3
00:00:53: RADIUS: Acct-Delay-Time [41] 6 0
```

The following is sample output from the **debug ip dhcp server events** command. The output was generated on a DHCP server and shows an exchange of DHCP messages between the client and server to negotiate a DHCP lease. The acknowledgment that confirms to the DHCP server that the client has accepted the assigned IP address triggers the accounting START message. It is shown in the last line of the following output:

```
00:45:50:DHCPD:DHCPDISCOVER received from client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31 on
interface GigabitEthernet0/0/0.
00:45:52:DHCPD:assigned IP address 10.10.10.16 to client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31.
00:45:52:DHCPD:Sending DHCPOFFER to client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31(10.10.10.16)
00:45:52:DHCPD:broadcasting BOOTREPLY to client 0001.42c9.ec75.
00:45:52:DHCPD:DHCPREQUEST received from client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31.
00:45:52:DHCPD:Sending DHCPACK to client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31
(10.10.10.16).
00:45:52:DHCPD:broadcasting BOOTREPLY to client 0001.42c9.ec75.
00:45:52:DHCPD:triggered Acct Start for 0001.42c9.ec75 (10.10.10.16).
```

The following is sample output from the **debug ip dhcp server events** command. The output was generated on a DHCP server and shows the receipt of an explicit release message from the DHCP client. The DHCP server triggers an accounting STOP message and then returns the IP address to the DHCP pool. Information about the accounting STOP message is shown in the third line of the following output:

```
00:46:26:DHCPD:DHCPRELEASE message received from client
0063.6973.636f.2d30.3030.312e.3432.6339.2e65.6337.352d.4574.31 (10.10.10.16)
00:46:26:DHCPD:triggered Acct Stop for (10.10.10.16).
00:46:26:DHCPD:returned 10.10.10.16 to address pool WIRELESS-POOL.
```

Example: Configuring a DHCP Lease Limit

In the following example, 5 DHCP clients are allowed to receive IP addresses. If a sixth client tries to obtain an IP address, the DHCPDISCOVER messages will not be forwarded to the DHCP server and a trap will be sent to the SNMP manager.

```
ip dhcp limit lease log
!
ip dhcp pool pool1
 network 10.1.1.0 255.255.255.0
!
interface loopback0
 ip address 10.1.1.1 255.255.255.0
!
interface serial 0/0.2 point-to-point
 ip dhcp limit lease 5
 ip unnumbered loopback0
 exit
snmp-server enable traps dhcp interface
```

Additional References

The following sections provide references related to configuring DHCP services for accounting and security.

Related Documents

Related Topic	Document Title
ARP commands: complete command syntax, command modes, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP commands: complete command syntax, command modes, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP conceptual information	"DHCP Overview" module
DHCP server configuration	"Configuring the Cisco IOS XE DHCP Server" module
DHCP ODAP configuration	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP client configuration	"Configuring the Cisco IOS XE DHCP Client" module
DHCP relay agent configuration	"Configuring the Cisco IOS XE DHCP Relay Agent" module
AAA and RADIUS configuration tasks	*Cisco IOS Security Configuration Guide*
AAA and RADIUS commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

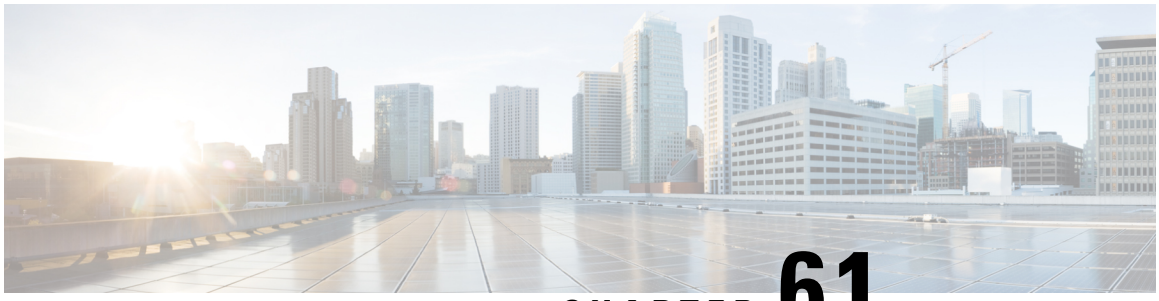
Feature Information for DHCP Services for Accounting and Security

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 92: Feature Information for DHCP Services for Accounting and Security

Feature Name	Releases	Feature Configuration Information
DHCP Per Interface Lease Limit and Statistics	Cisco IOS XE Release 2.1	This feature limits the number of DHCP leases offered to DHCP clients on an interface. DHCP server statistics reporting was enhanced to display interface-level statistics. The following commands were introduced or modified by this feature: **ip dhcp limit lease**, **ip dhcp limit lease log**, **clear ip dhcp limit lease**, **show ip dhcp limit lease**, and **show ip dhcp server statistics**.
DHCP Accounting	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	DHCP accounting introduces AAA and RADIUS support for DHCP configuration. The following command was introduced by this feature: **accounting**.
DHCP Secured IP Address Assignment	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	DHCP secure IP address assignment provides the capability to secure ARP table entries to DHCP leases in the DHCP database. This feature secures and synchronizes the MAC address of the client to the DHCP binding, preventing unauthorized clients or hackers from spoofing the DHCP server and taking over a DHCP lease of an authorized client. The following command was introduced by this feature: **update arp**. The following command was modified by this feature: **show ip dhcp server statistics**.
ARP Auto-logoff	Cisco IOS XE Release 3.9S	The ARP Auto-logoff feature enhances DHCP authorized ARP by providing finer control and probing of authorized clients to detect a logoff. The following command was introduced by this feature: **arp probe interval**.



CHAPTER 61

ISSU and SSO--DHCP High Availability Features

Cisco IOS XE Release 2.1 and 2.3 introduce the following series of Dynamic Host Configuration Protocol (DHCP) High Availability features:

- ISSU--DHCP Server
- SSO--DHCP Server
- ISSU--DHCP Relay on Unnumbered Interface
- SSO--DHCP Relay on Unnumbered Interface
- ISSU--DHCP Proxy Client
- SSO--DHCP Proxy Client
- ISSU--DHCP ODAP Client and Server
- SSO--DHCP ODAP Client and Server

These features are enabled by default when the redundancy mode of operation is set to Stateful Switchover (SSO).

Prerequisites for DHCP High Availability

- The Cisco IOS XE In-Service Software Upgrade (ISSU) process must be configured and working properly. See the “Cisco IOS XE In-Service Software Upgrade Process” feature module for more information.
- Stateful Switchover (SSO) must be configured and working properly. See the “Stateful Switchover” feature module for more information.

- Nonstop Forwarding (NSF) must be configured and working properly. See the "Cisco Nonstop Forwarding" feature module for more information.

Restrictions for DHCP High Availability

The DHCP high availability features do not support DHCP accounting or DHCP authorized Address Resolution Protocol (ARP).

Information About DHCP High Availability

ISSU

The ISSU process allows Cisco IOS XE software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco IOS XE software to be modified while packet forwarding continues, which increases network availability and reduces downtime caused by planned software upgrades.

SSO

SSO refers to the implementation of Cisco IOS XE software that allows applications and features to maintain a defined state between an active and standby Route Processor (RP).

In specific Cisco networking devices that support dual RPs, SSO takes advantage of RP redundancy to increase network availability. The SSO feature takes advantage of RP redundancy by establishing one of the RPs as the active RP while the other RP is designated as the standby RP, and then synchronizing critical state information between them. Following an initial synchronization between the two processors, SSO dynamically maintains RP state information between them.

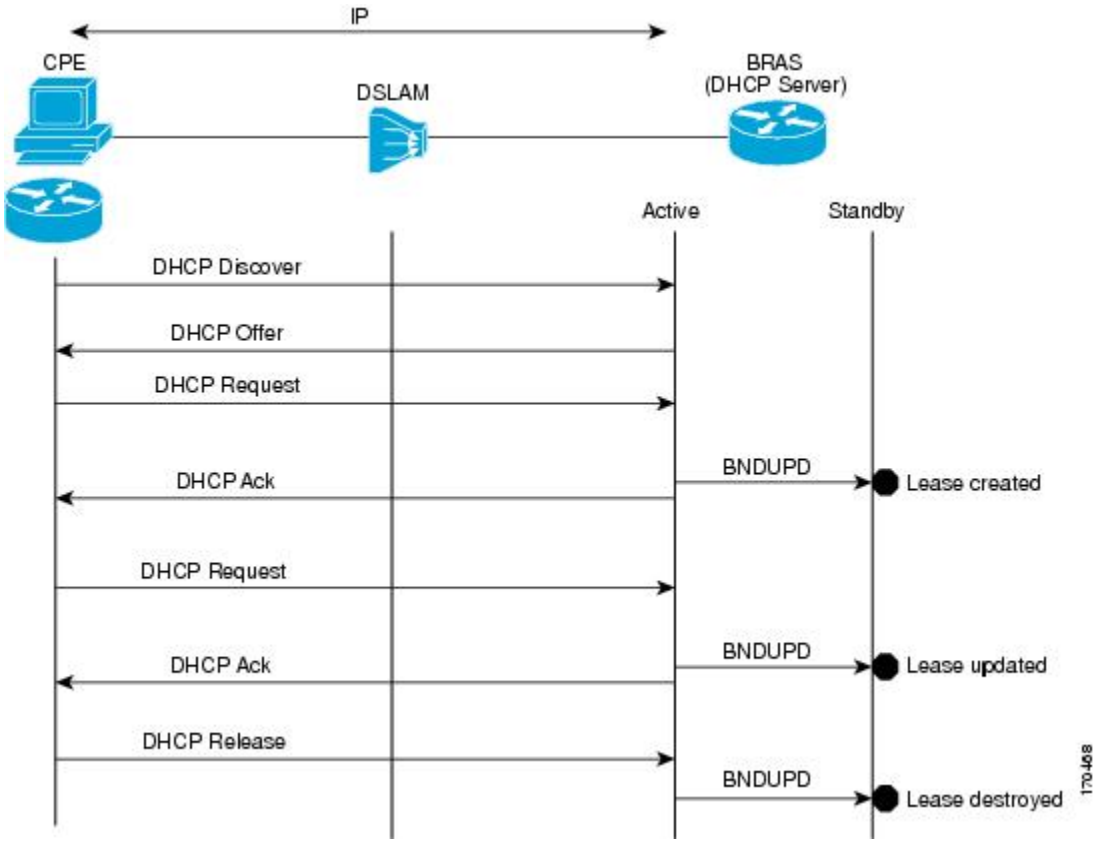
A switchover from the active to the standby processor occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance.

ISSU and SSO--DHCP Server

The DHCP server that is ISSU and SSO aware is able to detect when a router is failing over to the standby RP and preserve the DHCP lease across a switchover event.

Each DHCP binding is synchronized and re-created from the active RP to the standby RP upon lease commit. The figure below illustrates this process. The lease extension and release are also synchronized to the standby RP.

Figure 67: DHCP Server Maintaining States Between the Active and Standby Route Processor



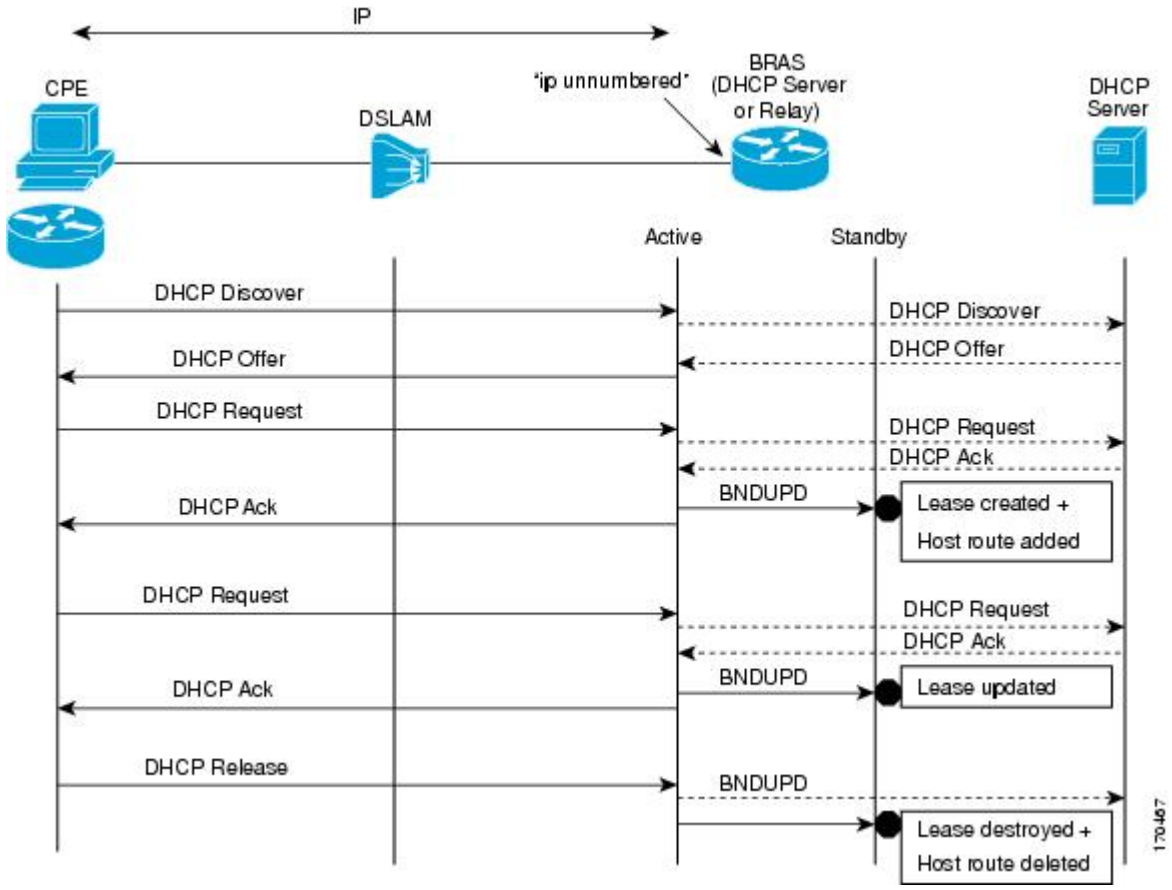
ISSU and SSO--DHCP Relay on Unnumbered Interface

The DHCP relay agent supports the use of unnumbered interfaces. For DHCP clients connected through the unnumbered interfaces, the DHCP relay agent automatically adds a static host route once the DHCP client obtains an address, specifying the unnumbered interface as the outbound interface. The route is automatically removed once the lease time expires or when the client releases the address.

The **ip helper-address** interface configuration command must be configured on the unnumbered interface to enable the Cisco IOS XE DHCP relay agent on unnumbered interfaces. See the "Configuring the Cisco IOS XE DHCP Relay Agent" configuration module for more information.

The ISSU and SSO DHCP relay on unnumbered interface functionality adds high availability support for host routes to clients connected through unnumbered interfaces. The DHCP relay agent can now detect when a router is failing over to the standby RP and keep the states related to unnumbered interfaces. The figure below illustrates the process.

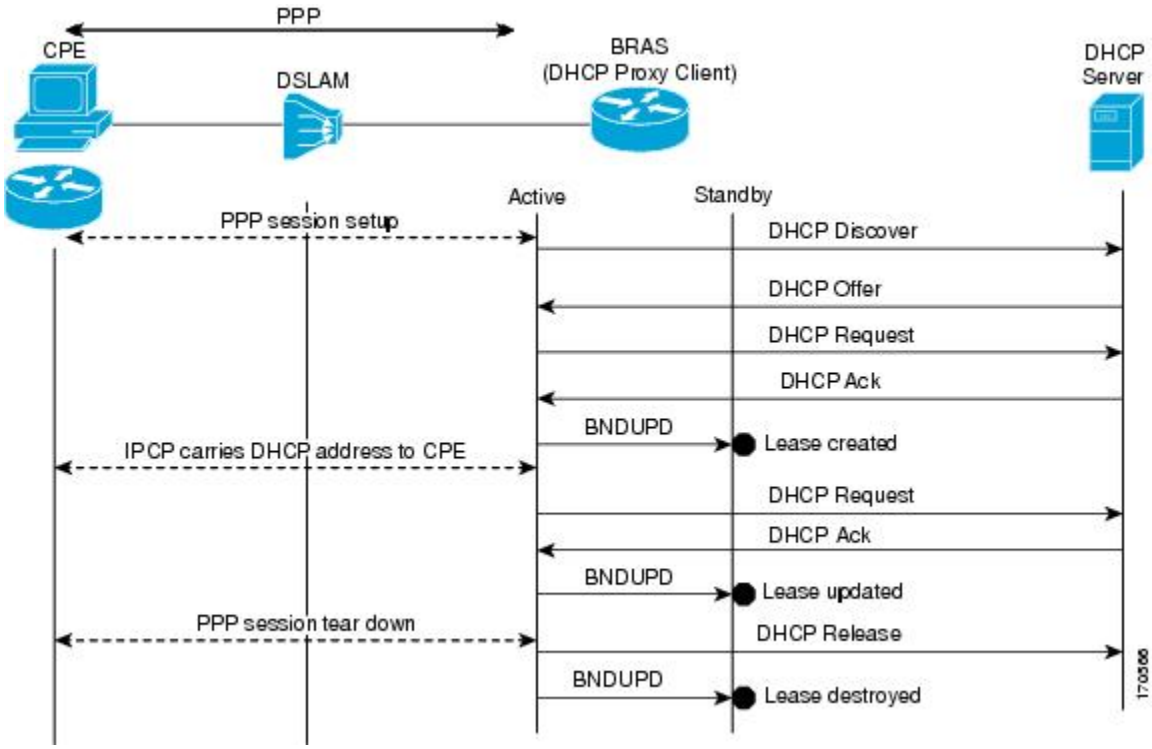
Figure 68: DHCP Maintaining States with an IP Unnumbered Interface



ISSU and SSO--DHCP Proxy Client

The DHCP proxy client enables the router to obtain a lease for configuration parameters from a DHCP server for a remote Point-to-Point Protocol (PPP) client. The DHCP proxy client that is ISSU and SSO aware is able to request a lease from the DHCP server and the state of the lease is synchronized between the active and standby RP. The figure below illustrates the process.

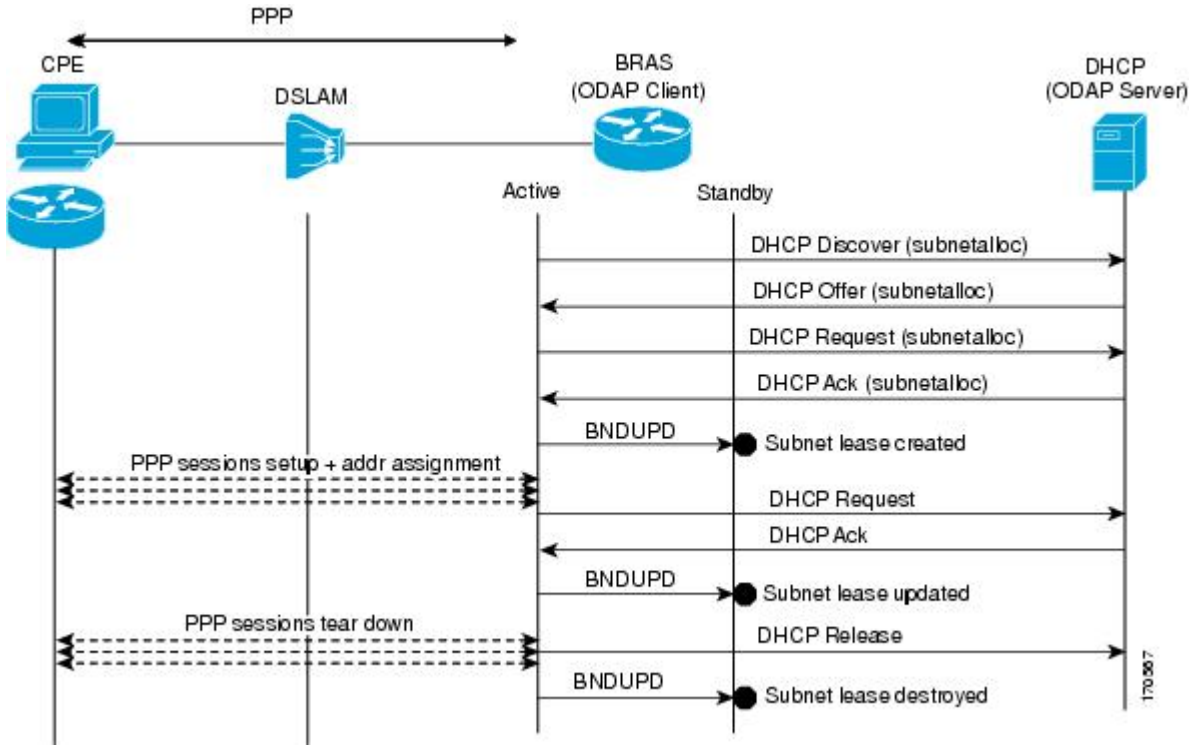
Figure 69: DHCP Proxy Client Lease Synchronization



ISSU and SSO--DHCP ODAP Client and Server

The DHCP on-demand address pool (ODAP) client that is ISSU and SSO aware can request a lease for a subnet from the DHCP ODAP server. After the DHCP ODAP server allocates the subnet to the client, the state of the lease is synchronized between the active and standby RP through binding updates. Following a switchover event, the DHCP ODAP client can continue to allocate IP addresses from the same subnets and also continue to renew the subnets from the DHCP ODAP server. The figure below illustrates the process.

Figure 70: ODAP Subnet Lease Synchronization



How to Configure DHCP High Availability

There are no configuration tasks. The DHCP high availability features are enabled by default when the redundancy mode of operation is set to SSO.

Configuration Examples for DHCP High Availability

There are no configuration examples for DHCP high availability features.

Additional References

The following sections provide references related to DHCP high availability features.

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Related Topic	Document Title
DHCP conceptual and configuration information	*Cisco IOS XE IP Addressing Services Configuration Guide*
In-Service Software Upgrade process conceptual and configuration information	"Cisco IOS XE In Service Software Upgrade Process" module
Nonstop Forwarding conceptual and configuration information	"Cisco Nonstop Forwarding" module
Stateful switchover conceptual and configuration information	"Stateful Switchover" module

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for DHCP High Availability Features

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 93: Feature Information for DHCP High Availability Features

Feature Name	Releases	Feature Information
ISSU--DHCP Server	Cisco IOS XE Release 2.1	The DHCP server has been enhanced to support ISSU.
SSO--DHCP Server	Cisco IOS XE Release 2.1	The DHCP server has been enhanced to support SSO.
ISSU--DHCP Relay on Unnumbered Interface	Cisco IOS XE Release 2.3	The DHCP relay on unnumbered interface has been enhanced to support ISSU.
SSO--DHCP Relay on Unnumbered Interface	Cisco IOS XE Release 2.1	The DHCP relay on unnumbered interface has been enhanced to support SSO.
ISSU--DHCP Proxy Client	Cisco IOS XE Release 2.3	The DHCP proxy client has been enhanced to support ISSU.
SSO--DHCP Proxy Client	Cisco IOS XE Release 2.3	The DHCP proxy client has been enhanced to support SSO.
ISSU--DHCP ODAP Client and Server	Cisco IOS XE Release 2.3	The DHCP ODAP client and server have been enhanced to support ISSU.
SSO--DHCP ODAP Client and Server	Cisco IOS XE Release 2.3	The DHCP ODAP client and server have been enhanced to support SSO.

Glossary

CPE --customer premises equipment. Terminating equipment, such as terminals, telephones, and modems, supplied by the service provider, installed at customer sites, and connected to the network.

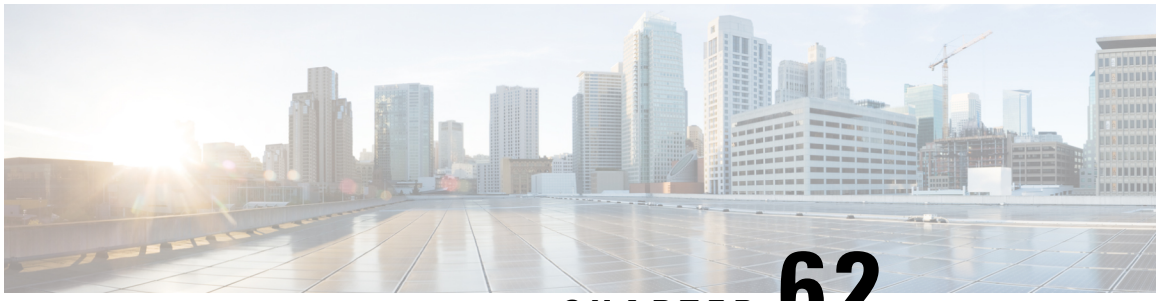
DSLAM --digital subscriber line access multiplexer. A device that connects many digital subscriber lines to a network by multiplexing the DSL traffic onto one or more network trunk lines.

ISSU --In Service Software Upgrade. ISSU is a process that allows Cisco IOS software to be updated or otherwise modified while packet forwarding continues.

ODAP --On-Demand Address Pool. ODAPs enable pools of IP addresses to be dynamically increased or reduced in size depending on the address utilization level. Once configured, the ODAP is populated with one or more subnets leased from a source server and is ready to serve address requests from DHCP clients or from PPP sessions.

RP --Route Processor. A generic term for the centralized control unit in a chassis.

SSO --Stateful Switchover. SSO refers to the implementation of Cisco IOS software that allows applications and features to maintain a defined state between an active and standby RP. When a switching occurs, forwarding and sessions are maintained. SSO makes an RP failure undetectable to the network.



CHAPTER 62

DHCPv6 Relay and Server - MPLS VPN Support

Information About DHCPv6 Relay and Server - MPLS VPN Support

DHCPv6 Server and Relay—MPLS VPN Support

To facilitate managed central services in a Multiprotocol Label Switching (MPLS)-based network, DHCPv6 must be made MPLS-aware so that a single resource can be used to serve multiple VPNs instead of dedicating a resource to a single VPN.

The DHCPv6 server implementation of MPLS VPN allows a per-pool configuration so that DHCPv6 pools can be associated with a VPN routing and forwarding (VRF) instance. The DHCPv6 server differentiates clients from various VRFs and assigns an IPv6 prefix accordingly from the respective VRF pools. Meanwhile, the DHCPv6 bindings store the clients' VRF information.

The DHCPv6 relay implementation allows the configuration of the destination VRF instance to which the relay messages will be forwarded. The relay adds the client's VPN information while forwarding the client's DHCPv6 requests toward the server, and the relay agent then processes the client's VPN information in reply packets from the server.

The relay agent adds IPv6 static routes for delegated prefixes in corresponding clients' VRF, and the relay agent's high availability (HA) functionality synchronizes the VRF information while synchronizing static routes created by the relay process.

The DHCPv6 relay and server VRF-aware features are disabled by default.

How to Configure DHCPv6 Relay and Server - MPLS VPN Support

Configuring a VRF-Aware Relay and Server for MPLS VPN Support

Configuring a VRF-Aware Relay

Note You do not have to configure this feature on specified interfaces. If you want the feature to be enabled globally only on a device, perform steps 1, 2, and 3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp-relay option vpn**
4. **interface** *type number*
5. **ipv6 dhcp relay option vpn**
6. **ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number* | **vrf** *vrf-name* | **global**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp-relay option vpn** **Example:** `Device(config)# ipv6 dhcp-relay option vpn`	Enables the DHCP for IPv6 relay VRF-aware feature globally.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**ipv6 dhcp relay option vpn** **Example:** `Device(config-if)# ipv6 dhcp relay option vpn`	Enables the DHCP for IPv6 relay VRF-aware feature on the specified interface. Enabling this command supersedes the configuration that is enabled by using the **ipv6 dhcp-relay option vpn** command.
Step 6	**ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number* | **vrf** *vrf-name* | **global**] **Example:** `Device(config-if)# ipv6 dhcp relay destination FE80::250:A2FF:FEBF:A056 ethernet 0/0`	Specifies a destination address to which client messages are forwarded.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring a VRF-Aware Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 dhcp server vrf enable**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ipv6 dhcp server vrf enable** **Example:** `Device(config-if)# ipv6 dhcp server vrf enable`	Enables the DHCPv6 server VRF-aware feature on an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for DHCPv6 Server - MPLS VPN Support

Example: Configuring a VRF-Aware Relay

```
Router# show ipv6 dhcp relay binding

Relay Bindings associated with default vrf:
Prefix: 2001:DB8:0:1::/64 (GigabitEthernet0/0/0)
  DUID: 00030001AABBCC006500
  IAID: 196609
  lifetime: 2592000
  expiration: 12:34:28 IST Oct 14 2010
Summary:
  Total number of Relay bindings = 1
  Total number of Relay bindings added by Bulk lease = 0
```

Example: Configuring a VRF-Aware Server

```
Router# show ipv6 dhcp binding

Client: FE80::A8BB:CCFF:FE00:6400
  DUID: 00030001AABBCC006400
  VRF : global
  Interface : GigabitEthernet0/0/0
  IA PD: IA ID 0x00030001, T1 302400, T2 483840
    Prefix: 2001::1/64
            preferred lifetime 604800, valid lifetime 2592000
            expires at Oct 15 2010 03:18 PM (2591143 seconds)

Router# show ipv6 route status

IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
       ND - Neighbor Discovery
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
```

```
S   2001::/64 [1/0]
     via FE80::A8BB:CCFF:FE00:6400, GigabitEthernet0/0/0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

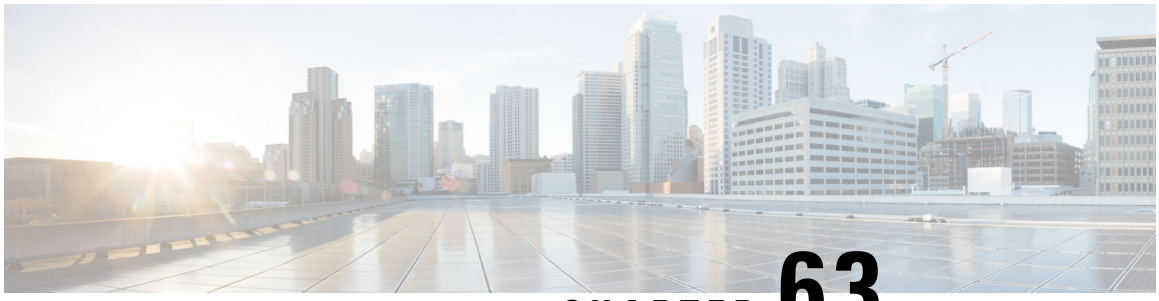
Feature Information for DHCPv6 Relay and Server - MPLS VPN Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 94: Feature Information for DHCPv6 Relay and Server - MPLS VPN Support

Feature Name	Releases	Feature Information
DHCPv6 Relay - MPLS VPN Support	Cisco IOS XE Release 3.3S	The DHCPv6 relay implementation allows the configuration of the destination VRF instance to which the relay messages will be forwarded. The following commands were introduced or modified: **ipv6 dhcp relay destination**, **ipv6 dhcp relay option vpn**, **ipv6 dhcp server vrf enable**, **show ipv6 dhcp relay binding**.
DHCPv6 Server - MPLS VPN Support	Cisco IOS XE Release 3.3S	The DHCPv6 server implementation of MPLS VPN support allows a per-pool configuration so DHCPv6 pools can be associated with a VRF instance. The following commands were introduced or modified: **ipv6 dhcp relay destination**, **ipv6 dhcp relay option vpn**, **ipv6 dhcp server vrf enable**, **show ipv6 dhcp relay binding**.



CHAPTER 63

Information About IPv6 Access Services: DHCPv6 Relay Agent

DHCPv6 Relay Agent

A DHCPv6 relay agent, which may reside on the client's link, is used to relay messages between the client and the server. The DHCPv6 relay agent operation is transparent to the client. A DHCPv6 client locates a DHCPv6 server using a reserved, link-scoped multicast address. For direct communication between the DHCPv6 client and the DHCPv6 server, both of them must be attached to the same link. However, in some situations where ease of management, economy, or scalability is a concern, it is desirable to allow a DHCPv6 client to send a message to a DHCPv6 server that is not connected to the same link. IPv6 enable is required for IPv6 DHCP relay, although IPv6 address is configured.

DHCPv6 Relay Agent Notification for Prefix Delegation

The DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is relayed by the relay agent to the client. When a prefix delegation option is found by the relay agent, the relay agent extracts the information about the prefix that is being delegated and inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

No user configuration is required for this feature. Static route management is done automatically by the relay agent.

IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

The DHCP—DHCPv6 Relay Agent Notification for Prefix Delegation feature leaves a static IPv6 route on the routing table of the relay agent. The registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the device doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that remains in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. Static routes will be removed when a DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote-ID for Gigabit Ethernet and Fast Ethernet Interfaces

The DHCPv6 Ethernet Remote ID Option feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, which includes port information, the system's DUID, and the VLAN ID. This information can be used to uniquely identify both the relay and the port on the relay through which the client packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem.

The addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically and no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. The Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is removed from the packet before it is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface ID

The DHCPv6 Relay—Reload Persistent Interface ID Option feature makes the interface ID option persistent. The interface ID is used by relay agents to decide which interface should be used to forward a RELAY-REPLY packet. A persistent interface-ID option will not change if the device acting as a relay agent goes offline during a reload or a power outage. When the device acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as, when the relay agent reboots and the number of interfaces in the interface index changes, or when the relay agents boot up and has more virtual interfaces than it did before the reboot). This feature prevents such scenarios from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as the short form of the interface name. The interface name as the DHCPv6 interface ID helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

DHCPv6 Relay Chaining

DHCPv6 messages can be relayed through multiple relay agents. This configuration is called *relay chaining*. A relay chaining configuration can be supported only when each relay agent adds information to DHCPv6 messages before relaying them. The information helps in relaying the DHCPv6 reply back to the DHCPv6 client through the same path.

The delegated IPv6 prefix must be routable in order to be useful. The actual DHCPv6 Prefix Delegation (PD) client may not be permitted to inject routes into the delegating network. In service provider (SP) networks, for example, an edge device typically acts as a DHCPv6 relay agent, and this edge device often has the responsibility to maintain routes within the SP network for clients' PD bindings. In the event that DHCPv6 requests and responses are relayed through a chain of DHCPv6 relays, there may be a need to introduce

appropriate routes (particularly with DHCPv6 PD) in the Forwarding Information Base (FIB) so that routing is handled transparently.

DHCPv6 Relay Agent Notification for Prefix Delegation

The DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is relayed by the relay agent to the client. When a prefix delegation option is found by the relay agent, the relay agent extracts the information about the prefix that is being delegated and inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

No user configuration is required for this feature. Static route management is done automatically by the relay agent.

IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

The DHCP—DHCPv6 Relay Agent Notification for Prefix Delegation feature leaves a static IPv6 route on the routing table of the relay agent. The registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the device doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that remains in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. Static routes will be removed when a DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote ID for Ethernet Interfaces

This feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, including port information, the system's DUID, and the VLAN ID. Collectively, this information can be used to uniquely identify both the relay and the port on the relay through which the client's packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem. This feature works only for Ethernet interfaces at this time.

This feature introduces no user configuration. Because the addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically, no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is stripped out of the packet before the packet is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface ID Option

This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. A persistent interface-ID option will not change

if the router acting as a relay agent goes offline (such as during a reload or a power outage). When the router acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as cases where the relay agent reboots and has a change in the number of interfaces in the interface index, or the relay agents boots up and has more virtual interfaces than it did before the reboot). This feature prevents this scenario from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as simply the short form of the interface name. This syntax helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

DHCPv6 Relay Chaining

This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. A persistent interface-ID option will not change if the router acting as a relay agent goes offline (such as during a reload or a power outage). When the router acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as cases where the relay agent reboots and has a change in the number of interfaces in the interface index, or the relay agents boots up and has more virtual interfaces than it did before the reboot). This feature prevents this scenario from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as simply the short form of the interface name. This syntax helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

How to Configure IPv6 Access Services: DHCPv6 Relay Agent

Configuring the DHCPv6 Relay Agent

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 enable**
5. **ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 4/2/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 5	**ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*] **Example:** `Device(config-if) ipv6 dhcp relay destination FE80::250:A2FF:FEBF:A056 gigabitethernet 4/3/0`	Specifies a destination address to which client packets are forwarded and enables the DHCPv6 relay service on the interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for IPv6 Access Services: DHCPv6 Relay Agent

Example: Configuring the DHCPv6 Relay Agent

```
Device# show ipv6 dhcp interface

Ethernet1/0 is in relay mode
  Relay destinations:
    3FFB:C00:C18:6:A8BB:CCFF:FE03:2701
Serial3/0 is in relay mode
  Relay destinations:
    3FFB:C00:C18:6:A8BB:CCFF:FE03:2600
    FE80::A8BB:CCFF:FE03:2801 on Serial3/0
    FF05::1:3
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

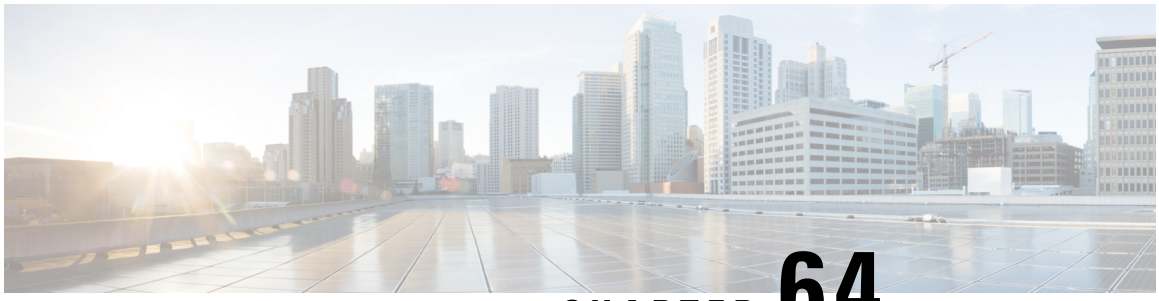
Feature Information for IPv6 Access Services: DHCPv6 Relay Agent

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 95: Feature Information for IPv6 Access Services: DHCPv6 Relay Agent

Feature Name	Releases	Feature Information
IPv6 Access Services: DHCPv6 Relay Agent		A DHCP relay agent, which may reside on the client's link, is used to relay messages between the client and server. The following commands were introduced or modified: **ipv6 dhcp relay destination**, **show ipv6 dhcp interface**.
DHCPv6 Relay Agent Notification for Prefix Delegation		DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 packet that is being relayed by the relay agent to the client.
DHCPv6 Relay: Reload Persistent Interface ID Option		This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent.



CHAPTER 64

IPv6 Access Services: Stateless DHCPv6

The stateless Dynamic Host Configuration Protocol for IPv6 (DHCPv6) feature allows DHCPv6 to be used for configuring a node with parameters that do not require a server to maintain any dynamic state for the node.

Information About IPv6 Access Services: Stateless DHCPv6

Information Refresh Server Option

The DHCPv6 information refresh option can specify an upper boundary for the length of time a client should wait before refreshing information retrieved from DHCPv6. This option is used with stateless DHCPv6, because there are no addresses or other entities with lifetimes that can tell the client when to contact the DHCPv6 server to refresh its configuration.

SIP Server Options

Session initiation protocol (SIP) server options contain either a list of domain names or IPv6 addresses that can be mapped to one or more SIP outbound proxy servers. One option carries a list of domain names, and the other option carries a list of 128-bit IPv6 addresses.

SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions or calls. A SIP system has several logical components: user agents, proxy servers, redirect servers, and registrars. User agents may contain SIP clients; proxy servers always contain SIP clients.

SNTP Server Option

The SNTP server option provides a list of one or more IPv6 addresses of SNTP servers available to the client for synchronization. The clients use these SNTP servers to synchronize their system time to that of the standard time servers. The server may list the SNTP servers in decreasing order of preference, but clients must treat the list of SNTP servers as an ordered list.

How to Configure IPv6 Access Services: Stateless DHCPv6

Configuring the Stateless DHCPv6 Function

The server maintains no state related to clients; for example, no prefix pools and records of allocation are maintained. Therefore, this function is "stateless" DHCPv6.

Configuring the Stateless DHCPv6 Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **dns-server** *ipv6-address*
5. **domain-name** *domain*
6. **exit**
7. **interface** *type number*
8. **ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]
9. **ipv6 nd other-config flag**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool dhcp-pool`	Configures a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**dns-server** *ipv6-address* **Example:** `Device(config-dhcp)# dns-server 2001:DB8:3000:3000::42`	Specifies the Domain Name System (DNS) IPv6 servers available to a DHCPv6 client.

	Command or Action	Purpose
Step 5	**domain-name** *domain* **Example:** `Device(config-dhcp)# domain-name example.com`	Configures a domain name for a DHCPv6 client.
Step 6	**exit** **Example:** `Device(config-dhcp)# exit`	Exits DHCPv6 pool configuration mode, and returns the device to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface serial 3`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 8	**ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**] **Example:** `Device(config-if)# ipv6 dhcp server dhcp-pool`	Enables DHCPv6 on an interface.
Step 9	**ipv6 nd other-config flag** **Example:** `Device(config-if)# ipv6 nd other-config flag`	Sets the "other stateful configuration" flag in IPv6 router advertisements (RAs).
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring the Stateless DHCPv6 Client

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address autoconfig** [**default**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial 3`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 address autoconfig** [**default**] **Example:** `Device(config-if)# ipv6 address autoconfig`	Enables automatic configuration of IPv6 addresses using stateless autoconfiguration on an interface and enables IPv6 processing on the interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Enabling Processing of Packets with Source Routing Header Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 source-route**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 source-route** **Example:**	Enables processing of the IPv6 type 0 routing header.

	Command or Action	Purpose
	`Device(config)# ipv6 source-route`	
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Importing Stateless DHCPv6 Server Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **import dns-server**
5. **import domain-name**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Router(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**import dns-server** **Example:** `Router(config-dhcp)# import dns-server`	Imports the DNS recursive name server option to a DHCPv6 client.
Step 5	**import domain-name** **Example:** `Router(config-dhcp)# import domain-name`	Imports the domain search list option to a DHCPv6 client.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-dhcp)# end`	Returns to privileged EXEC mode.

Configuring the SNTP Server Option

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **sntp address** *ipv6-address*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**sntp address** *ipv6-address* **Example:** `Device(config-dhcp)# sntp address 2001:DB8:2000:2000::33`	Specifies the SNTP server list to be sent to the client.
Step 5	**end** **Example:** `Device(config-dhcp)# end`	Returns to privileged EXEC mode.

Importing SIP Server Information

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **import sip address**
5. **import sip domain-name**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Router(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**import sip address** **Example:** `Router(config-dhcp)# import sip address`	Imports the SIP server IPv6 address list option to the outbound SIP proxy server.
Step 5	**import sip domain-name** **Example:** `Router(config-dhcp)# import sip domain-name`	Imports a SIP server domain-name list option to the outbound SIP proxy server.
Step 6	**end** **Example:** `Router(config-dhcp)# end`	Returns to privileged EXEC mode.

Importing the SNTP Server Option

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **import sntp address** *ipv6-address*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**import sntp address** *ipv6-address* **Example:** `Device(config-dhcp)# import sntp address 2001:DB8:2000:2000::33`	Imports the SNTP server option to a DHCPv6 client.
Step 5	**end** **Example:** `Device(config-dhcp)# end`	Returns to privileged EXEC mode.

Configuration Examples for IPv6 Access Services: Stateless DHCPv6

Example: Configuring the Stateless DHCPv6 Function

The following example shows how to use the DHCPv6 function to configure clients with information about the name lookup system. The server is configured with a DHCP pool, which contains the name lookup information that is to be passed to clients. It does not need to contain a prefix pool. This DHCP pool is attached to the access link to customers (GigabitEthernet0/0/0) using the **ipv6 dhcp server** command. The access link also has the **ipv6 nd other-config-flag** command enabled. RA messages sent from this interface will inform clients that they should use DHCPv6 for "other" (for example, nonaddress) configuration information.

```
ipv6 dhcp pool dhcp-pool
 dns-server 2001:DB8:A:B::1
 dns-server 2001:DB8:3000:3000::42
 domain-name example.com
!
interface GigabitEthernet0/0/0
description Access link down to customers
 ipv6 address 2001:DB8:1234:42::1/64
 ipv6 nd other-config-flag
 ipv6 dhcp server dhcp-pool
```

The client has no obvious DHCPv6 configuration. However, the **ipv6 address autoconfig** command on the uplink to the service provider (GigabitEthernet 0/0/0) causes the following two events:

- Addresses are autoconfigured on the interface, based on prefixes in RA messages received from the server.
- If received RA messages have the "other configuration" flag set, the interface will attempt to acquire the other (for example, nonaddress) configuration from any DHCPv6 servers.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

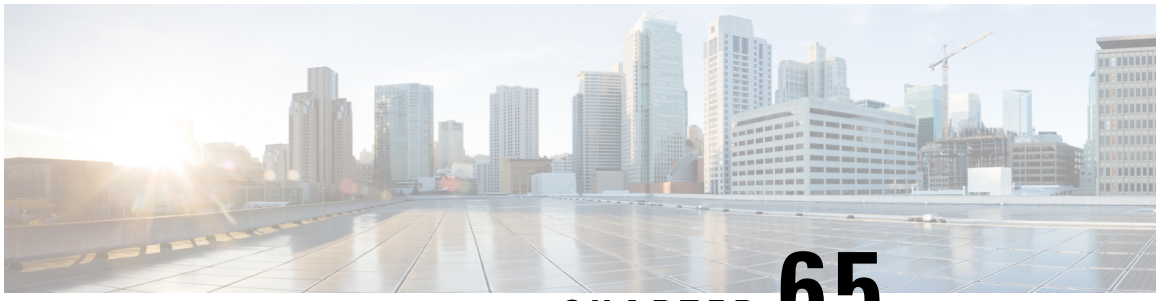
Feature Information for IPv6 Access Services: Stateless DHCPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 96: Feature Information for IPv6 Access Services: Stateless DHCPv6

Feature Name	Releases	Feature Information
IPv6 Access Services: Stateless DHCPv6	Cisco IOS XE Release 2.5 Cisco IOS XE Release 3.9S	Stateless DHCPv6 allows DHCPv6 to be used for configuring a node with parameters that do not require a server to maintain any dynamic state for the node. The following commands were introduced or modified: **dns-server**, **domain-name**, **import dns-server**, **import domain-name**, **import sip address**, **import sip domain-name**, **import sntp address**, **ipv6 address autoconfig**, **ipv6 dhcp pool**, **ipv6 dhcp server**, **ipv6 nd other-config-flag**, **ipv6 source-route**, **sntp address**.



CHAPTER 65

IPv6 Access Services: DHCPv6 Prefix Delegation

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) prefix delegation feature can be used to manage link, subnet, and site addressing changes.

Information About IPv6 Access Services: DHCPv6 Prefix Delegation

DHCPv6 Prefix Delegation

The IPv6 Access Services—DHCPv6 Prefix Delegation feature can be used to manage link, subnet, and site addressing changes. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) can be used in environments to deliver stateful and stateless information, which are defined as follows:

- Stateful prefix delegation—Address assignment is centrally managed and clients must obtain configuration information such as address autoconfiguration and neighbor discovery that is not available through protocols.
- Stateless prefix delegation—Stateless configuration parameters do not require a server to maintain any dynamic state for individual clients, such as Domain Name System (DNS) server addresses and domain search list options.

Extensions to DHCPv6 also enable prefix delegation, through which an ISP can automate the process of assigning prefixes to a customer for use within the customer's network. The prefix delegation occurs between a provider edge (PE) device and customer premises equipment (CPE) using the DHCPv6 prefix delegation option. Once the ISP has delegated prefixes to a customer, the customer may further subnet and assign prefixes to the links in the customer's network.

Configuring Nodes Without Prefix Delegation

Stateless DHCPv6 allows DHCPv6 to be used for configuring a node with parameters that do not require a server to maintain any dynamic state for the node. The use of stateless DHCP is controlled by router advertisement (RA) messages multicasted by routers. The Cisco IOS XE DHCPv6 client will invoke stateless DHCPv6 when it receives an RA. The Cisco IOS XE DHCPv6 server will respond to a stateless DHCPv6 request with configuration parameters, such as the DNS servers and domain search list options.

Client and Server Identification

Each DHCPv6 client and server is identified by a DHCP unique identifier (DUID). The DUID is carried in client identifier and server identifier options. The DUID is unique across all DHCP clients and servers, and it is stable for any specific client or server. DHCPv6 uses DUIDs based on link-layer addresses for both the client and server identifier. The device uses the MAC address from the lowest-numbered interface to form the DUID. The network interface is assumed to be permanently attached to the device.

When a DHCPv6 client requests two prefixes with the same DUID but with different identity association identifiers (IAIDs) on two different interfaces, these prefixes are considered to be for two different clients, and the interface information is maintained for both.

Rapid Commit

The DHCPv6 client can obtain configuration parameters from a server either through a rapid two-message exchange (solicit, reply) or through a four-message exchange (solicit, advertise, request, and reply). By default, the four-message exchange is used. When the rapid-commit option is enabled by both the client and the server, the two-message exchange is used.

DHCPv6 Client, Server, and Relay Functions

The DHCPv6 client, server, and relay functions are mutually exclusive on an interface. When one of these functions is enabled and a user tries to configure a different function on the same interface, one of the following messages is displayed: “Interface is in DHCP client mode,” “Interface is in DHCP server mode,” or “Interface is in DHCP relay mode.”

The following sections describe these functions:

Client Function

The DHCPv6 client function can be enabled on individual IPv6-enabled interfaces.

Note You need APPX license package to enable the DHCPv6 client function on the device.

The DHCPv6 client can request and accept those configuration parameters that do not require a server to maintain any dynamic state for individual clients, such as DNS server addresses and domain search list options.

The DHCPv6 client can also request the delegation of prefixes. The prefixes acquired from a delegating device will be stored in a local IPv6 general prefix pool. The prefixes in the general prefix pool can then be referred to from other applications; for example, the general prefix pool can be used to number device downstream interfaces.

Server Selection

A DHCPv6 client builds a list of potential servers by sending a solicit message and by collecting advertise message replies from servers. These messages are ranked based on the preference value, and servers may add a preference option to their advertise messages explicitly stating their preference value. If the client needs to acquire prefixes from servers, only servers that have advertised prefixes are considered.

IAPD and IAID

An Identity Association for Prefix Delegation (IAPD) is a collection of prefixes assigned to a requesting device. A requesting device may have more than one IAPD; for example, one for each of its interfaces.

Each IAPD is identified by an IAID. The IAID is chosen by the requesting device and is unique among the IAPD IAIDs on the requesting device. IAIDs are made consistent across reboots by using information from the associated network interface, which is assumed to be permanently attached to the device.

Server Function

The DHCPv6 server function can be enabled on individual IPv6-enabled interfaces.

The DHCPv6 server can provide configuration parameters that do not require the server to maintain any dynamic state for individual clients, such as DNS server addresses and domain search list options. The DHCPv6 server may be configured to perform prefix delegation.

All the configuration parameters for clients are independently configured into DHCPv6 configuration pools, which are stored in the NVRAM. A configuration pool can be associated with a particular DHCPv6 server on an interface when it is started. Prefixes that are to be delegated to clients may be specified either as a list of preassigned prefixes for a particular client or as IPv6 local prefix pools that are also stored in the NVRAM. The list of manually configured prefixes or IPv6 local prefix pools can be referenced and used by DHCPv6 configuration pools.

The DHCPv6 server maintains an automatic binding table in memory to track the assignment of some configuration parameters, such as prefixes between the server and its clients. Automatic bindings can be stored permanently in the database agent, such as a remote TFTP server or a local NVRAM file system.

Configuration Information Pool

A DHCPv6 configuration information pool is a named entity that includes information about available configuration parameters and policies that the control assignment of the parameters to clients from the pool. A pool is configured independently and is associated with the DHCPv6 service through the CLI.

Each configuration pool can contain the following configuration parameters and operational information:

- Prefix delegation information, which includes:
  - A prefix pool name and associated preferred and valid lifetimes
  - A list of available prefixes for a particular client and associated preferred and valid lifetimes
- A list of IPv6 addresses of DNS servers
- A domain search list, which is a string containing domain names for the DNS resolution

DHCP for IPv6 Address Assignment

DHCPv6 enables DHCP servers to pass configuration parameters, such as IPv6 network addresses, to IPv6 clients. The DHCPv6 Individual Address Assignment feature manages nonduplicate address assignment in the correct prefix based on the network where the host is connected. Assigned addresses can be from one or

multiple prefix pools. Additional options, such as the default domain and DNS name-server address, can be passed back to the client. Address pools can be assigned for use on a specific interface or on multiple interfaces, or the server can automatically find the appropriate pool.

Prefix Assignment

A prefix-delegating router (DHCPv6 server) selects prefixes to be assigned to a requesting router (DHCPv6 client) upon receiving a request from the client. The server can select prefixes for a requesting client by using static and dynamic assignment mechanisms. Administrators can manually configure a list of prefixes and associated preferred and valid lifetimes for an IAPD of a specific client that is identified by its DUID.

When the delegating router receives a request from a client, it checks if there is a static binding configured for the IAPD in the client's message. If a static binding is present, the prefixes in the binding are returned to the client. If no such binding is found, the server attempts to assign prefixes for the client from other sources.

The Cisco IOS XE DHCPv6 server can assign prefixes dynamically from an IPv6 local prefix pool. When the server receives a prefix request from a client, it attempts to obtain unassigned prefixes from the pool. After the client releases the previously assigned prefixes, the server returns them to the pool for reassignment.

An IPv6 prefix delegating router can also select prefixes for a requesting router based on an external authority such as a RADIUS server using the Framed-IPv6-Prefix attribute.

Automatic Binding

Each DHCPv6 configuration pool has an associated binding table. The binding table contains records of all prefixes in the configuration pool that have been explicitly delegated to clients. Each entry in the binding table contains the following information:

- Client DUID.
- Client IPv6 address.
- A list of IAPDs associated with the client.
- A list of prefixes delegated to each IAPD.
- Preferred and valid lifetimes for each prefix.
- The configuration pool to which this binding table belongs.
- The network interface on which the server that is using the pool is running.

A binding table entry is automatically created whenever a prefix is delegated to a client from the configuration pool, and the entry is updated when the client renews, rebinds, or confirms the prefix delegation. A binding table entry is deleted when the client voluntarily releases all the prefixes in the binding, the valid lifetimes of all prefixes have expired, or administrators run the **clear ipv6 dhcp binding** command.

Binding Database

Each permanent storage to which the binding database is saved is called the database agent. A database agent can be a remote host, such as an FTP server, or a local file system, such as the NVRAM.

Automatic bindings are maintained in the RAM and can be saved to some permanent storage so that information about configurations, such as prefixes assigned to clients, is not lost after a system reload. The bindings are stored as text records for easy maintenance. Each record contains the following information:

- DHCPv6 pool name from which the configuration was assigned to the client.

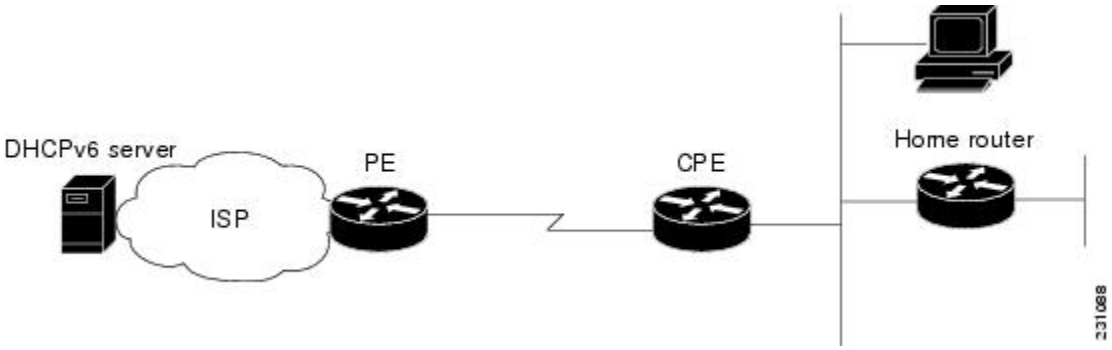
- Interface identifier from which the client requests were received.
- The client IPv6 address.
- The client DUID.
- IAID of the IAPD.
- Prefix delegated to the client.
- The prefix length.
- The prefix preferred lifetime in seconds.
- The prefix valid lifetime in seconds.
- The prefix expiration time stamp.
- Optional local prefix pool name from which the prefix was assigned.

DHCPv6 Server Stateless Autoconfiguration

Hierarchical DHCPv6 for stateless configuration parameters allows a stateless or stateful DHCPv6 client to export configuration parameters (DHCPv6 options) to a local DHCPv6 server pool. The local DHCPv6 server can then provide the imported configuration parameters to other DHCPv6 clients.

The figure below shows a typical broadband deployment.

Figure 71: Broadband Topology



The CPE interface towards the PE can be a stateless or stateful DHCPv6 client. In either case, the ISP-side DHCPv6 server may provide configuration parameters such as DNS server addresses, domain names, and Simple Network Time Protocol (SNTP) servers to the DHCP client on the CPE. Such information can be specific to ISPs.

In addition to being a DHCPv6 client (for example, towards the ISP), the CPE may act as a DHCPv6 server to the home network. For example, neighbor discovery followed by a stateless or stateful DHCPv6 client can occur on the link between the CPE and the home devices (such as the home router or PC). In some cases, the information to be provided to the home network is the same as that obtained from the ISP-side DHCPv6 server. Because this information can be dynamically changed, it cannot be hard-configured in the CPE's configuration. Therefore, the DHCPv6 component on the CPE allows automatic importing of configuration parameters from the DHCPv6 client to the DHCPv6 server pool.

DHCPv6 supports the following options for IPv6 on the server:

Information Refresh Server Option

The DHCPv6 information refresh option can specify a maximum limit for the length of time a client should wait before refreshing the information retrieved from DHCPv6. This option is used with stateless DHCPv6 because there are no addresses or other entities with lifetimes that can tell the client when to contact the DHCPv6 server to refresh its configuration.

NIS- and NIS+-Related Server Options

Users can configure the network information service (NIS) or NIS plus (NIS+) address or domain name of a DHCPv6 server using NIS- and NIS+-related options, and then import that information to the DHCPv6 client.

SIP Server Options

Session Initiation Protocol (SIP) server options contain either a list of domain names or a list of IPv6 addresses that can be mapped to one or more SIP outbound proxy servers. One option carries a list of domain names, and the other option carries a list of 128-bit IPv6 addresses.

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions or calls. A SIP system has several logical components: user agents, proxy servers, redirect servers, and registrars. User agents may contain SIP clients; proxy servers always contain SIP clients.

SNTP Server Option

The SNTP server option provides a list of one or more IPv6 addresses of SNTP servers available to the client for synchronization. Clients use these SNTP servers to synchronize their system time to that of the standard time servers. The DHCPv6 server may list the SNTP servers in decreasing order of preference, but clients treat the list of SNTP servers as an ordered list.

How to Configure IPv6 Access Services: DHCPv6 Prefix Delegation

Configuring the DHCPv6 Server Function

Configuring the DHCPv6 Configuration Pool

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **domain-name** *domain*
5. **dns-server** *ipv6-address*
6. **prefix-delegation** *ipv6-prefix* / *prefix-length client-duid* [**iaid** *iaid*] [*lifetime*]
7. **prefix-delegation pool** *poolname* [**lifetime** *valid-lifetime preferred-lifetime*]
8. **exit**
9. **interface** *type number*

10. **ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**domain-name** *domain* **Example:** `Device(config-dhcp)# domain-name example.com`	Configures a domain name for a DHCPv6 client.
Step 5	**dns-server** *ipv6-address* **Example:** `Device(config-dhcp)# dns-server 2001:DB8:3000:3000::42`	Specifies the DNS IPv6 servers available to a DHCPv6 client.
Step 6	**prefix-delegation** *ipv6-prefix* / *prefix-length client-duid* [**iaid** *iaid*] [*lifetime*] **Example:** `Device(config-dhcp)# prefix-delegation 2001:DB8:1263::/48 0005000400F1A4D070D03`	Specifies a manually configured numeric prefix to be delegated to a specified client's IAPD.
Step 7	**prefix-delegation pool** *poolname* [**lifetime** *valid-lifetime preferred-lifetime*] **Example:** `Device(config-dhcp)# prefix-delegation pool pool1 lifetime 1800 60`	Specifies a named IPv6 local prefix pool from which prefixes are delegated to DHCPv6 clients.
Step 8	**exit** **Example:**	Exits DHCPv6 pool configuration mode, and returns the device to global configuration mode.

	Command or Action	Purpose
	`Device(config-dhcp)# exit`	
Step 9	**interface** *type number* **Example:** `Device(config)# interface serial 3`	Specifies an interface type and number, and enters interface configuration mode.
Step 10	**ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**] **Example:** `Device(config-if)# ipv6 dhcp server pool1`	
Step 11	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring a Binding Database Agent for the Server Function

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp database** *agent* [**write-delay** *seconds*] [**timeout** *seconds*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp database** *agent* [**write-delay** *seconds*] [**timeout** *seconds*] **Example:** `Device(config)# ipv6 dhcp database tftp://10.0.0.1/dhcp-binding`	Specifies DHCPv6 binding database agent parameters.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring the DHCPv6 Client Function

General prefixes can be defined dynamically from a prefix received by a DHCPv6 prefix delegation client. The delegated prefix is stored in a general prefix.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**] **Example:** `Device(config-if)# ipv6 dhcp client pd dhcp-prefix`	Enables the DHCPv6 client process and enables a request for prefix delegation through a specified interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Deleting Automatic Client Bindings from the DHCPv6 Binding Table

SUMMARY STEPS

1. **enable**
2. **clear ipv6 dhcp binding** [*ipv6-address*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 dhcp binding** [*ipv6-address*] [**vrf** *vrf-name*] **Example:** `Device# clear ipv6 dhcp binding`	Deletes automatic client bindings from the DHCPv6 binding table.

Configuration Examples for IPv6 Access Services: DHCPv6 Prefix Delegation

Example: Configuring the DHCPv6 Server Function

DHCPv6 clients are connected to the DHCPv6 server on Gigabit Ethernet interface 0/0/0. The server is configured to use parameters from the DHCP pool called dhcp-pool. This pool provides clients with the IPv6 address of a DNS server and the domain name to be used. It also specifies that prefixes can be delegated from the prefix pool called client-prefix-pool1. The prefixes delegated will have valid and preferred lifetimes of 1800 and 600 seconds respectively. The prefix pool named client-prefix-pool1 has a prefix of length /40 from which it will delegate (sub) prefixes of length /48.

```
ipv6 dhcp pool dhcp-pool
 prefix-delegation pool client-prefix-pool1 lifetime 1800 600
 dns-server 2001:DB8:3000:3000::42
 domain-name example.com
!

 description downlink to clients
 ipv6 address FEC0:240:104:2001::139/64
 ipv6 dhcp server dhcp-pool
!
ipv6 local pool client-prefix-pool1 2001:DB8:1200::/40 48
```

The following example from the **show ipv6 dhcp** command shows the DUID of the device:

```
Router# show ipv6 dhcp

This device's DHCPv6 unique identifier(DUID): 000300010002FCA5DC1C
```

In the following example, the show ipv6 dhcp binding command shows information about two clients, including their DUIDs, IAPDs, prefixes, and preferred and valid lifetimes:

```
Router# show ipv6 dhcp binding

Client: FE80::202:FCFF:FEA5:DC39 (GigabitEthernet2/1/0)
  DUID: 000300010002FCA5DC1C
  IA PD: IA ID 0x00040001, T1 0, T2 0
    Prefix: 3FFE:C00:C18:11::/68
            preferred lifetime 180, valid lifetime 12345
            expires at Nov 08 2002 02:24 PM (12320 seconds)
Client: FE80::202:FCFF:FEA5:C039 (GigabitEthernet2/1/0)
  DUID: 000300010002FCA5C01C
  IA PD: IA ID 0x00040001, T1 0, T2 0
    Prefix: 3FFE:C00:C18:1::/72
            preferred lifetime 240, valid lifetime 54321
            expires at Nov 09 2002 02:02 AM (54246 seconds)
    Prefix: 3FFE:C00:C18:2::/72
            preferred lifetime 300, valid lifetime 54333
            expires at Nov 09 2002 02:03 AM (54258 seconds)
    Prefix: 3FFE:C00:C18:3::/72
            preferred lifetime 280, valid lifetime 51111
```

In the following example, the **show ipv6 dhcp database** command provides information on the binding database agents TFTP, NVRAM, and flash:

```
Router# show ipv6 dhcp database

Database agent tftp://172.19.216.133/db.tftp:
  write delay: 69 seconds, transfer timeout: 300 seconds
  last written at Jan 09 2003 01:54 PM,
     write timer expires in 56 seconds
  last read at Jan 06 2003 05:41 PM
  successful read times 1
  failed read times 0
  successful write times 3172
  failed write times 2
Database agent nvram:/dhcpv6-binding:
  write delay: 60 seconds, transfer timeout: 300 seconds
  last written at Jan 09 2003 01:54 PM,
     write timer expires in 37 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 3325
  failed write times 0
Database agent flash:/dhcpv6-db:
  write delay: 82 seconds, transfer timeout: 3 seconds
  last written at Jan 09 2003 01:54 PM,
    write timer expires in 50 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 2220
  failed write times 614
```

Example: Configuring the DHCPv6 Configuration Pool

In the following example, the **show ipv6 dhcp pool** command provides information on the configuration pool named svr-p1, including the static bindings, prefix information, the DNS server, and the domain names found in the svr-p1 pool:

```
Router# show ipv6 dhcp pool

DHCPv6 pool: svr-p1
  Static bindings:
    Binding for client 000300010002FCA5C01C
      IA PD: IA ID 00040002,
        Prefix: 3FFE:C00:C18:3::/72
                preferred lifetime 604800, valid lifetime 2592000
      IA PD: IA ID not specified; being used by 00040001
        Prefix: 3FFE:C00:C18:1::/72
                preferred lifetime 240, valid lifetime 54321
        Prefix: 3FFE:C00:C18:2::/72
                preferred lifetime 300, valid lifetime 54333
        Prefix: 3FFE:C00:C18:3::/72
                preferred lifetime 280, valid lifetime 51111
  Prefix from pool: local-p1, Valid lifetime 12345, Preferred lifetime 180
  DNS server: 2001:DB8:1001::1
  DNS server: 2001:DB8:1001::2
  Domain name: example1.net
  Domain name: example2.net
  Domain name: example3.net
Active clients: 2
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname Router
!
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
!
!
interface GigabitEthernet0/0/0
 ip address 10.4.9.11 255.0.0.0
 media-type 10BaseT
 ipv6 address 2001:DB8:C18:1::/64 eui-64
```

Example: Configuring the DHCPv6 Client Function

This DHCPv6 client has three interfaces: Gigabit Ethernet interface 0/0/0 is the upstream link to a service provider, which has a DHCPv6 server function enabled. The Fast Ethernet interfaces 0/0/0 and 0/1/0 are links to local networks.

The upstream interface, Gigabit Ethernet interface 0/0/0, has the DHCPv6 client function enabled. Prefixes delegated by the provider are stored in the general prefix called prefix-from-provider.

The local networks, Fast Ethernet interfaces 0/0/0 and 0/1/0, both assign interface addresses based on the general prefix called prefix-from-provider. The bits on the left of the addresses come from the general prefix, and the bits on the right of the addresses are specified statically.

```
interface GigabitEthernet 0/0/0
 description uplink to provider DHCP IPv6 server
 ipv6 dhcp client pd prefix-from-provider
!
interface FastEthernet 0/0/0
 description local network 0
 ipv6 address prefix-from-provider ::5:0:0:0:100/64
```

```
!
interface FastEthernet 0/1/0
 description local network 1
 ipv6 address prefix-from-provider ::6:0:0:0:100/64
```

Example: Configuring a Database Agent for the Server Function

The DHCPv6 server is configured to store table bindings to the file named dhcp-binding on the server at address 10.0.0.1 using the TFTP protocol. The bindings are saved every 120 seconds.

```
ipv6 dhcp database tftp://10.0.0.1/dhcp-binding write-delay 120
```

The following example shows how to specify DHCP for IPv6 binding database agent parameters and store binding entries in bootflash:

```
ipv6 dhcp database bootflash
```

Example: Displaying DHCP Server and Client Information on the Interface

The following is sample output from the **show ipv6 dhcp interface** command. In the first example, the command is used on a router that has an interface acting as a DHCPv6 server. In the second example, the command is used on a router that has an interface acting as a DHCPv6 client:

```
Router1# show ipv6 dhcp interface

 is in server mode
  Using pool: svr-p1
  Preference value: 20
  Rapid-Commit is disabled

Router2# show ipv6 dhcp interface

 is in client mode
  State is OPEN (1)
  List of known servers:
    Address: FE80::202:FCFF:FEA1:7439, DUID 000300010002FCA17400
    Preference: 20
      IA PD: IA ID 0x00040001, T1 120, T2 192
        Prefix: 3FFE:C00:C18:1::/72
                preferred lifetime 240, valid lifetime 54321
                expires at Nov 08 2002 09:10 AM (54319 seconds)
        Prefix: 3FFE:C00:C18:2::/72
                preferred lifetime 300, valid lifetime 54333
                expires at Nov 08 2002 09:11 AM (54331 seconds)
        Prefix: 3FFE:C00:C18:3::/72
                preferred lifetime 280, valid lifetime 51111
                expires at Nov 08 2002 08:17 AM (51109 seconds)
      DNS server: 2001:DB8:1001::1
      DNS server: 2001:DB8:1001::2
      Domain name: example1.net
      Domain name: example2.net
      Domain name: example3.net
    Prefix name is cli-p1
    Rapid-Commit is enabled
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

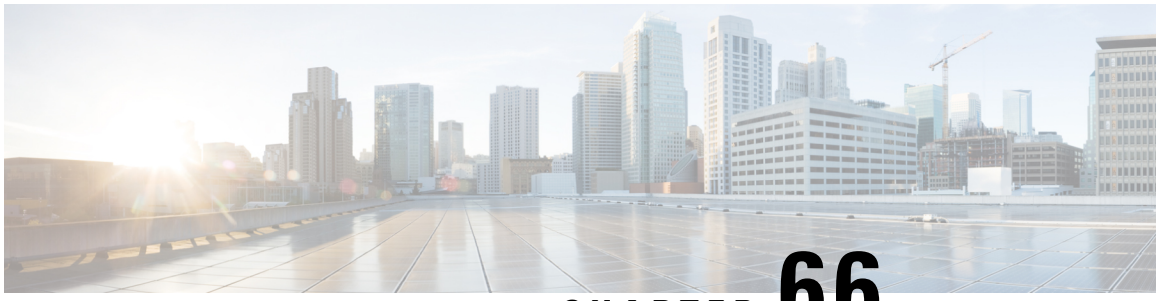
Feature Information for IPv6 Access Services: DHCPv6 Prefix Delegation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 97: Feature Information for IPv6 Access Services: DHCPv6 Prefix Delegation

Feature Name	Releases	Feature Information
IPv6 Access Services: DHCPv6 Prefix Delegation		The DHCPv6 Prefix Delegation feature can be used to manage link, subnet, and site addressing changes. DHCPv6 can be used in environments to deliver stateful and stateless information. The following commands were introduced or modified: **clear ipv6 dhcp binding**, **dns-server**, **domain-name**, **ipv6 dhcp client pd**, **ipv6 dhcp database**, **ipv6 dhcp pool**, **ipv6 dhcp server**, **prefix-delegation**, **prefix-delegation pool**, **show ipv6 dhcp**, **show ipv6 dhcp binding**, **show ipv6 dhcp interface**, **show ipv6 dhcp pool**.



CHAPTER 66

Asymmetric Lease for DHCPv6 Relay Prefix Delegation

The Asymmetric Lease for DHCPv6 Relay Prefix Delegation feature is used to manage or change the lease renewal by the relay agent.

Restrictions for Asymmetric Lease for DHCPv6 Prefix Delegation

- The Asymmetric Lease for DHCPv6 Prefix Delegation is enabled only when the relay destination is configured and IA-PD route option is enabled.
- The short lease value must be less than the allotted T1 value of the server.
- DHCPv6 relay agent must not remember the modified IA-PD option T1 and T2 values.
- Short Lease for Cisco IOS DHCPv6 PD client and IOS DHCPv6 server is not supported.
- Relay agent does not detect the live status of the prefix delegation clients. The Relay agent will not handle the prefix delegation client prefix when the client fails to renew.

Information about Asymmetric Lease for DHCPv6 Relay Prefix Delegation

Asymmetric lease is also referred as short lease which is shorter than the actual lease that is granted by the server. You can configure the short lease on a relay agent which overrides the actual lease. The short lease provides an option to force a lease renewal for clients before the original lease expires. It detects the lease expiry early and helps to keep the clients status live.

The Cisco DHCPv6 prefix delegation client receives the IPv6 prefix from the DHCPv6 server. The prefix delegation client uses the allotted prefix to assign IPv6 addresses to LAN side hosts. The prefix delegation client, relay agent, and server retain the allotted prefix even when the connnection is down in the following scenarios:

1. If the prefix delegation client is down, the relay or DHCPv6 server waits until the granted lease expires to release the allotted prefix.
2. If the relay or server is down, the prefix delegation client fails to renew the lease and retains the prefix until the valid lifetime timer expires.

In both scenarios, the prefix is retained until the valid lifetime timer is expired. The short lease enables the client to reclaim the unused address prefixes earlier. Also, it enables the client to detect the failure of the relay or server earlier and allows prefix delegation client to reinitiate the DHCP6 prefix delegation prefix assignment.

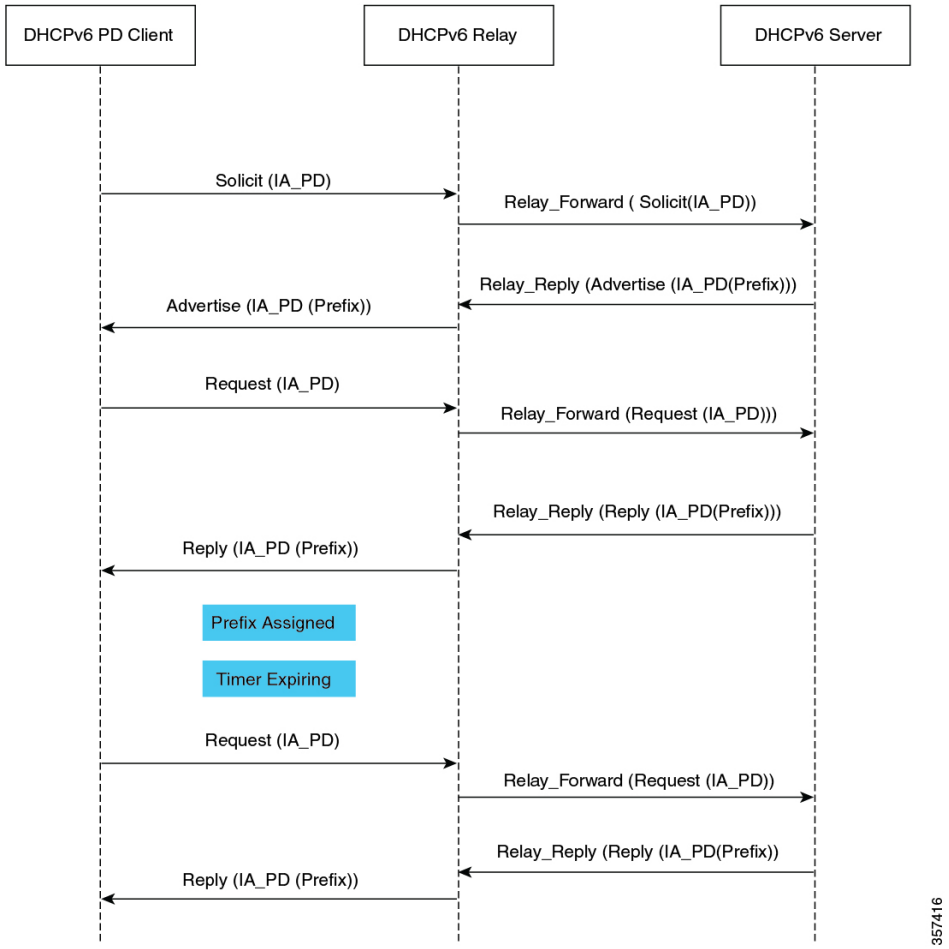
DHCPv6 Prefix Delegation with Asymmetric Lease

In DHCPv6 Prefix Delegation with Asymmetric Lease, the client initiates the prefix assignment by sending the multicast desired packet to the server. The Relay intercepts and encapsulates the desired request message in Relay-Forward and forwards it to the server. The server responds with the advertise or reply message encapsulated in relay-reply to a Relay agent. When the Relay agent receives the relay-reply message, it performs the following:

1. Extracts the encapsulated advertise or reply message destined to the prefix delegation client.
2. Stores the server assigned T1 and T2 values.
3. Validates the configured short lease value with the server that is assigned to T1 and T2 values.
4. Modifies the advertise or reply packet and replaces the T1 and T2 values with the new values defined as per the short lease configuration.

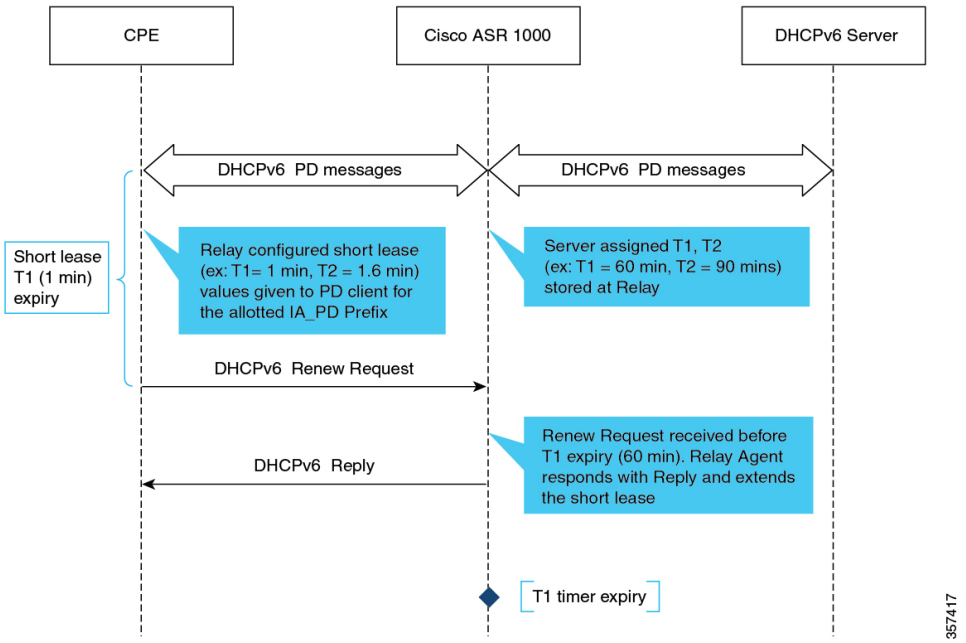
The following sequence diagram describes the prefix delegation flow with short lease.

Figure 72: DHCPv6 PD-Client, Relay and Server with Asymmetric Lease



The following sequence diagram depicts the message sequence between the DHCPv6 PD client, relay, and server.

Figure 73: Message Sequence between DHCPv6 PD Client, Relay, and Server



Deriving IA-PD Option T1 and T2 Values

The IA-PD option contatins the T1 and T2 values. The T1 represents the time at which prefix delegation client renews the IA_PD prefix and lifetime values. It sends the renewal message to the server which is provided with the client's addresses and configuration parameters to extend the lifetimes on the IA_PD prefix assigned to the client.

T2 represents the time at which the prefix delegation client tries to rebind. For example, contact any available DHCPv6 server to extend the lifetime of IA_PD prefix. The rebind message is sent after a client receives no response to a renewal message. Both T1 and T2 are time duration relative to the current time expressed in units of seconds.

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          OPTION_IA_PD           |          option-length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          IAID (4 octets)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               T1                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               T2                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                                 .
.                          IA_PD-options                          .
.                                                                 .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

On the DHCPv6 relay agent, the minimum allowed short lease value is 60 seconds. The T1 can be assigned with the configured short lease value. T2 is derived from T1 as shown below.

T2 = minimum (2 * T1 * 0.8, DHCPv6 Server assigned T2 value)

If the DHCPv6 server sets T1 and T2 to the value 0, the T1 and T2 calculation is determined by the client. In this case, T1 can be assigned with the configured short lease.

T2 shall be calculated as shown below.

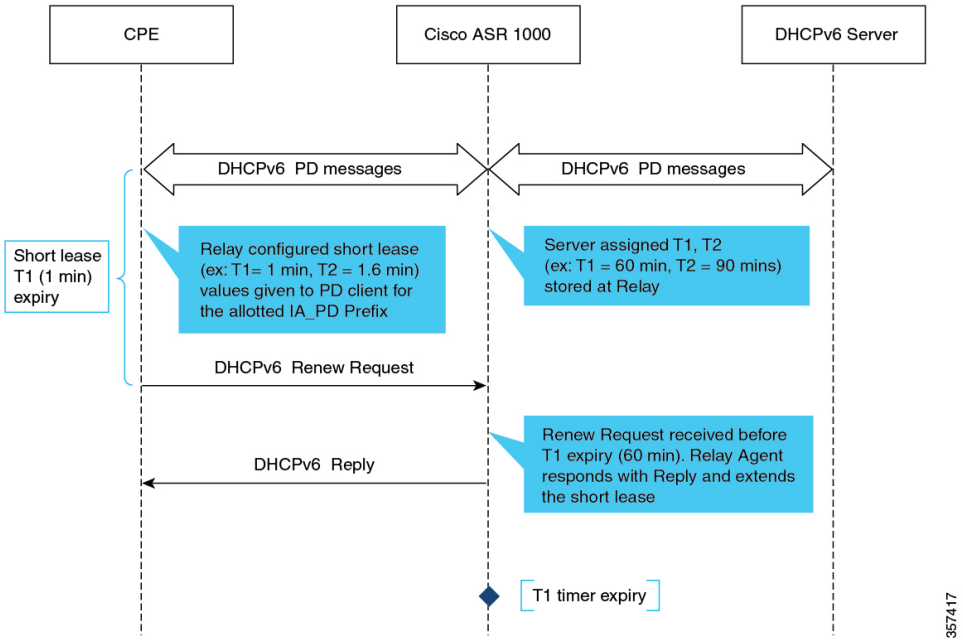
T2 = 2 * T1 * 0.8

Note If the configured short lease value is greater than the server assigned T1 value then this feature will not be applied.

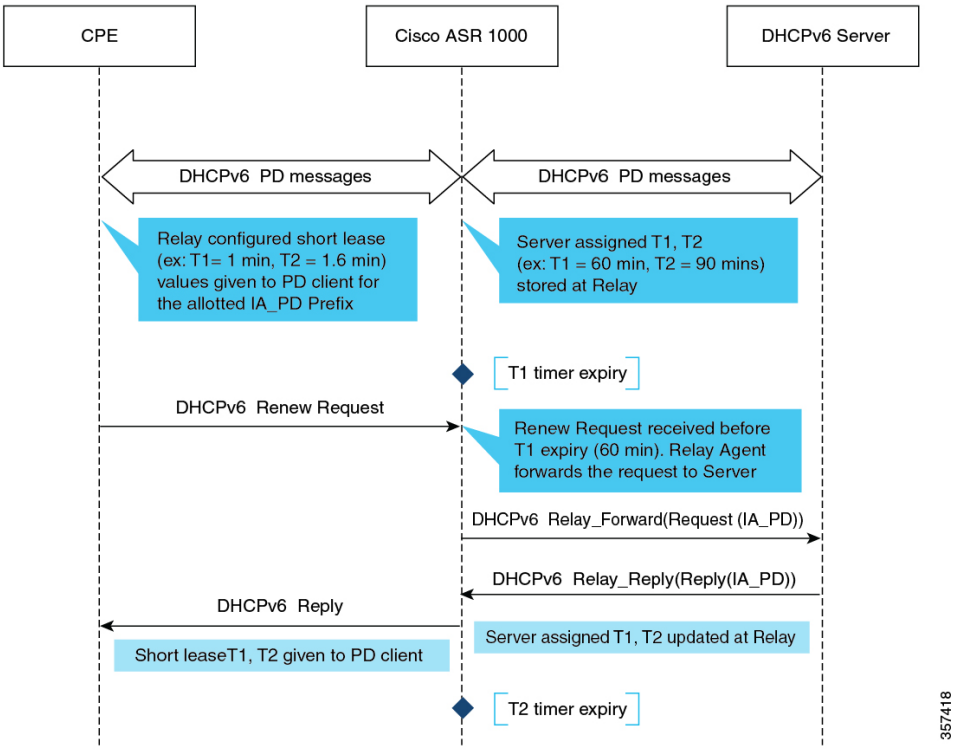
DHCPv6 relay agent uses timer wheel for handling the server assigned T1 and T2 values.

Renewing and Rebinding Scenarios

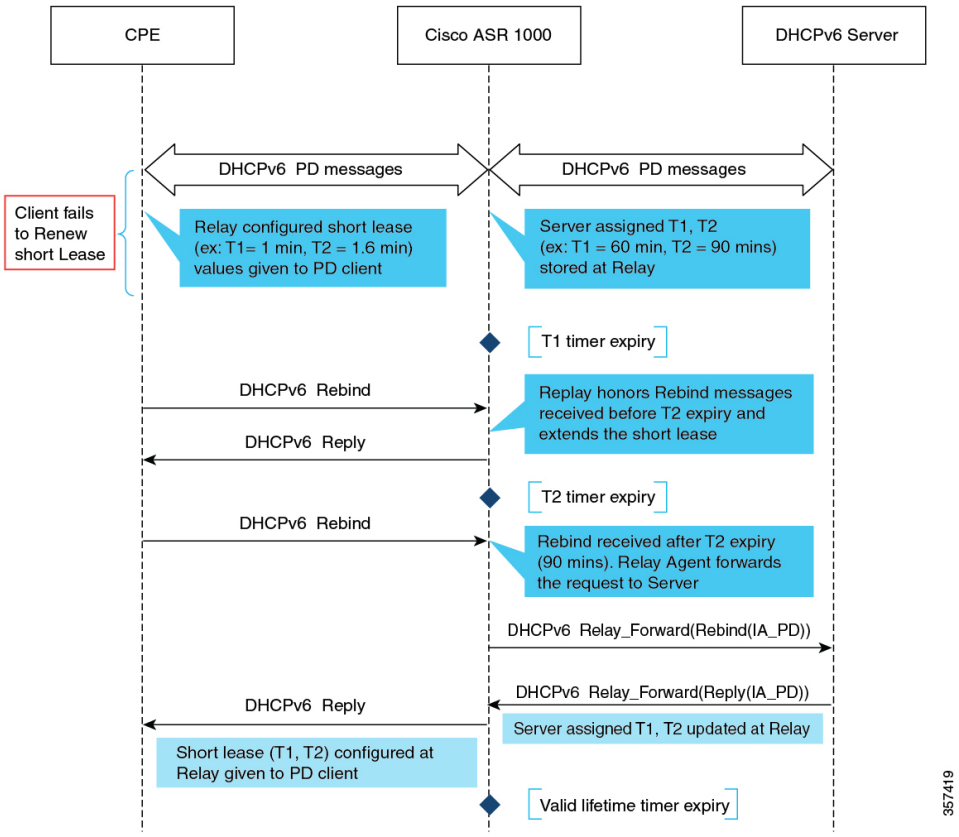
The relay agent stores the actual T1 and T2 values received from DHCPv6 Server. For every reply with IA options received from the server, relay will update the T1 and T2 values which will also take care of any changes in the server side configurations. In this sequence diagrams, CPE is the DHCPv6 Prefix Delegation client and Cisco ASR 1000 Series is the DHCPv6 relay agent.



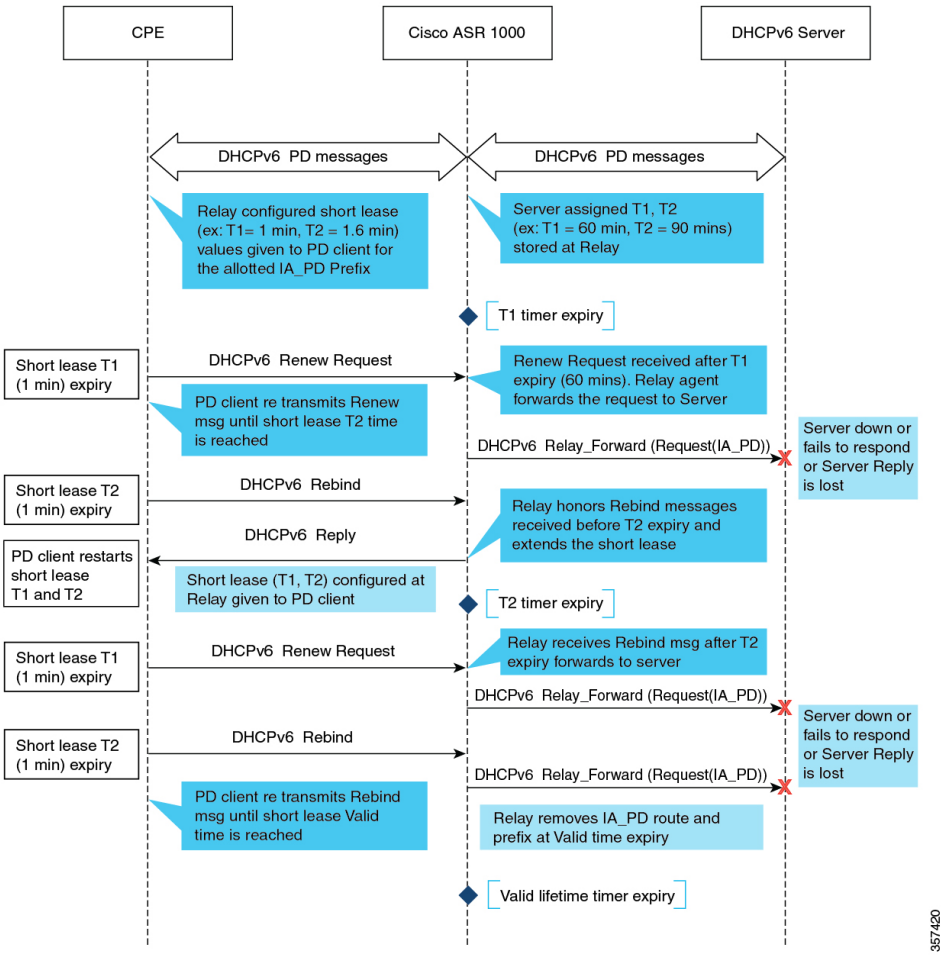
The following sequence diagram depicts the short lease renewal before the expiry of the server assigned T1 value.



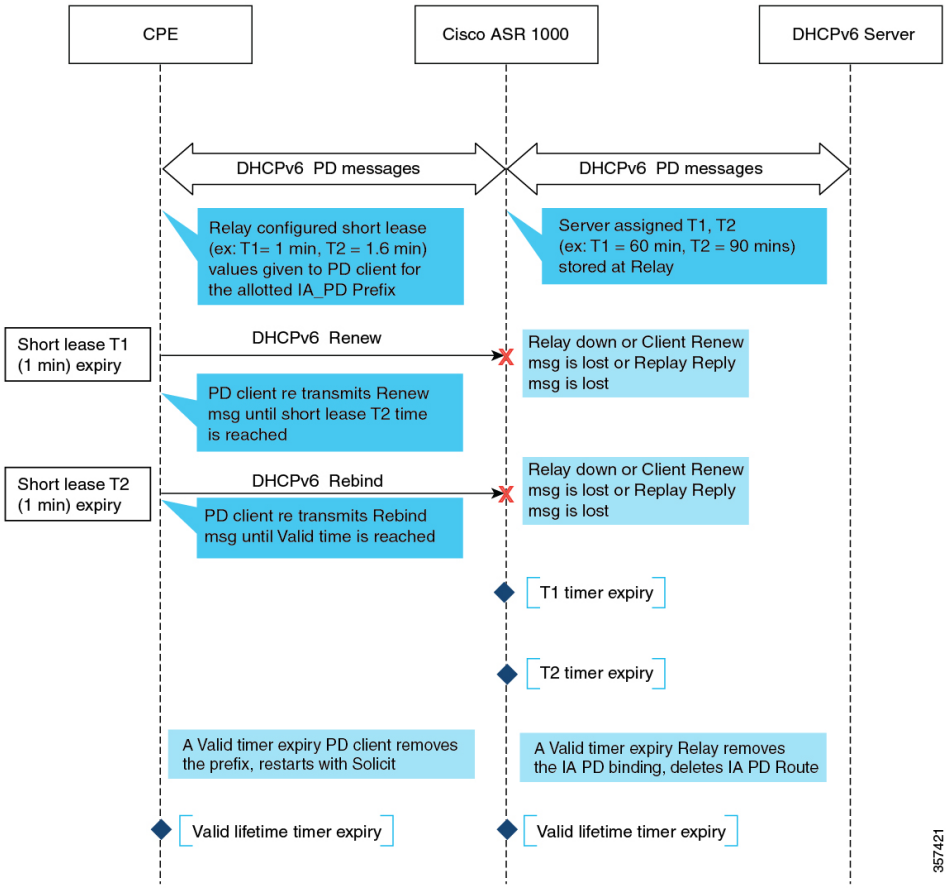
The following message sequence diagram depicts the short lease renewal after the expiry of the server assigned T1 value.



The following message sequence diagram depicts the short lease rebind scenario.



The following message sequence diagram depicts how to handle the short lease renewal and rebind when the server is down.



Reconfiguring Scenario

The DHCPv6 server sends a reconfigure message to the client to inform that the server has a new or updated configuration parameters. With this information, the client initiates a renewal or reply or information-request and reply transaction with the server to receive the updated information.

The server tries to send the reconfigure message to an IPv6 unicast address of the DHCP client. If the server does not have an address to which it can send the reconfigure message directly to the client, the server uses Relay-reply option to send the message to a relay agent that will relay the message to the client.

When responding to a reconfigure message, the client creates and sends the information-request message with the exception that the client includes a server identifier option with the identifier from the reconfigure message to which the client is responding.

Renewing Scenario

When responding to a reconfigure, the client creates and sends the renew message with the exception that the client copies the Option Request option and any IA options from the reconfigure message into the renew message.

Configuring Asymmetric Lease

You can apply the Asymmetric lease configuration on per interface or globally for all interfaces.

Configuring Asymmetric Lease on an Interface

Before you begin

To configure the asymmetric lease on an interface, perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip dhcp relay destination optionshort-lease source-information** *time in seconds*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ip dhcp relay destination optionshort-lease source-information** *time in seconds* **Example:** `Router(config-if)# ip dhcp relay short-lease 500`	Sets and enables the short lease for the client on the interface. You can set the lease time in seconds. The range is from 60 to 3600 seconds.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring Asymmetric Lease in Global Configuration Mode

Before you begin

To configure the asymmetric lease globally:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp relay destination optionshort-lease source-information** *time in seconds*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp relay destination optionshort-lease source-information** *time in seconds* **Example:** `Router(config)# ip dhcp relay short-lease 500`	Sets and enables the short lease for the client globally. You can set the lease time in seconds. The range is from 60 to 3600 seconds.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for the Asymmetric Lease

Example: Configuring the Asymmetric Lease on an Interface

The **show running-config interface Ethernet** command displays the interface where short lease is configured:

```
Router# show running-config interface Ethernet0/0
Building configuration...

Current configuration : 215 bytes
```

```
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:10::1/64
 ipv6 enable
 ipv6 dhcp relay destination 2001:DB8:10::1
 ipv6 dhcp relay short-lease 500

end
```

Verifying the Configuration

To verify the configuration on the interface, use the **show ipv6 dhcp interface GigabitEthernet 2** and **show ipv6 dhcp relay binding** commands.

:

```
Router# show ipv6 dhcp interface GigabitEthernet 2
GigabitEthernet2 is in client mode
  Prefix State is OPEN
  Renew will be sent in 11:38:28
  Address State is IDLE
  List of known servers:
    Reachable via address: FE80::250:56FF:FEAE:17A8
    DUID: 00030001001EE6383500
    Preference: 0
    Configuration parameters:
      IA PD: IA ID 0x00080001, T1 120, T2 240
        Prefix: 8001:DB8::/48
                preferred lifetime 120, valid lifetime 240
                expires at Mar 31 2020 09:24 PM (51 seconds)
      DNS server: 2000:3000:4000::1
      Information refresh time: 0
  Prefix name: pd-client
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled


end
```

```
Device# show ipv6 dhcp relay binding

Relay Bindings associated with default vrf:
Prefix: 8001:DB8::/48 (Ethernet0/0)
  DUID: 00030001AABBCC000A00
  IAID: 131073
  Short Lease T1: 60 (Expired)
  Short Lease T2: 96 (Active)
  T1: 3600
    expiration: 19:47:00 IST Jul 15 2020
  T2: 7200
    expiration: 20:47:00 IST Jul 15 2020
  lifetime: 150
  expiration: 21:17:00 IST Jul 15 2020
Summary:
  Total number of Relay bindings = 1
  Total number of IAPD bindings = 1
  Total number of IANA bindings = 0
  Total number of Relay bindings added by Bulk lease = 0
RELAY#
```

DHCPv6 Short Lease Performance Scaling

The following table provides the performance scaling information for the DHCPv6 short lease.

Table 98: DHCPv6 Short Lease Performance Scaling

Number of Sessions	Short Lease Value	Calls per Second	Performance with ISG	Performance without ISG
32000	60 sec	100	Yes	No
32000	60 sec	100	No	Yes

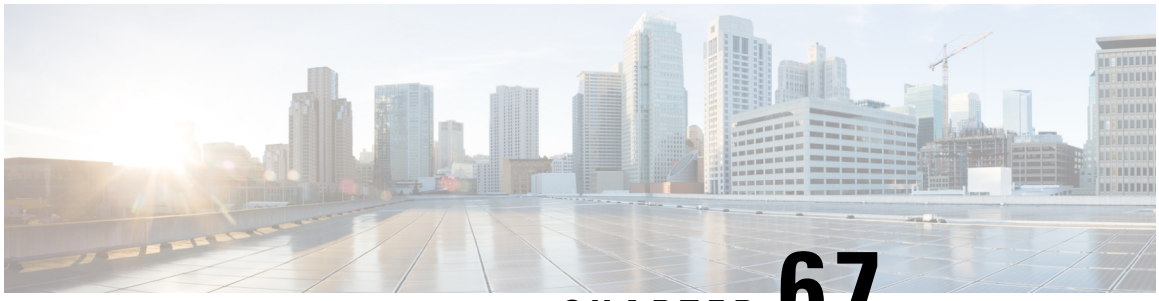
Feature Information for Asymmetric Lease for DHCPv6 Relay Prefix Delegation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 99: Feature Information for Asymmetric Lease for DHCPv6 Relay Prefix Delegation

Feature Name	Releases	Feature Information
Asymmetric Lease for DHCPv6 Relay Prefix Delegation		This feature allows you to manage or change the lease renewal. It provides options to force renewal of lease and also detects when the lease is nearing the expiry date.



CHAPTER 67

Configuration Examples for DHCP for IPv6 Broadband

Information About DHCP for IPv6 Broadband

Prefix Delegation

An IPv6 prefix delegating device selects IPv6 prefixes to be assigned to a requesting device upon receiving a request from the client. The delegating device might select prefixes for a requesting device in the following ways:

- Dynamic assignment from a pool of available prefixes.
- Dynamic assignment from a pool name obtained from the RADIUS server.
- Assignment of prefix obtained from the RADIUS sever.

Contrary to IPv4 address assignment, an IPv6 user will be assigned a prefix, not a single address. Typically the Internet service provider (ISP) assigns a 64- or 48-bit prefix.

Accounting Start and Stop Messages

PPP calls a registry to allow DHCPv6 to append the delegated prefix information to accounting start and stop messages. When accounting is configured for a DHCPv6 pool, accounting interim packets are sent to broadband sessions after binding is provided from the pool.

Forced Release of a Binding

The DHCPv6 server maintains an automatic binding table in memory to track the assignment of some configuration parameters, such as prefixes between the server and its clients. The automatic bindings can be

stored permanently in the database agent, which can be, for example, a remote TFTP server or local NVRAM file system.

DHCPv6 invokes a routine when the virtual interface used by PPP terminates. This routine automatically releases any delegated prefix bindings associated with the PPP virtual interface that is being terminated.

When a PPP virtual interface terminates, the routine runs through the full table of DHCPv6 bindings checking for the matching interface. Because PPP uses a virtual interface, this subroutine clears any related lease information when the PPP connection terminates.

Note In IPv6 broadband deployment using DHCPv6, you must enable release of prefix bindings associated with a PPP virtual interface using the **ipv6 dhcp binding track ppp** command. This ensures that DHCPv6 bindings are tracked together with PPP sessions, and in the event of DHCP REBIND failure, the client initiates DHCPv6 negotiation again.

How to Configure DHCP for IPv6 Broadband

Enabling the Sending of Accounting Start and Stop Messages

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **accounting** *mlist*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCP for IPv6 configuration information pool and enters DHCP for IPv6 pool configuration mode.

	Command or Action	Purpose
Step 4	**accounting** *mlist* **Example:** `Device(config-dhcp)# accounting list1`	Enables accounting start and stop messages to be sent.

Removing Delegated Prefix Bindings

Perform this task to release any delegated prefix bindings associated with the PPP virtual interface that is being terminated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 dhcp bindings track ppp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface VirtualAccess2.2`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 dhcp bindings track ppp** **Example:** `Device(config-if)# ipv6 dhcp bindings track ppp`	Releases any delegated prefix leases associated with the PPP virtual interface that is being terminated.

Configuration Examples for DHCP for IPv6 Broadband

Example: Enabling the Sending of Accounting Start and Stop Messages

This example shows how to enable a device to send accounting start and stop messages.

```
Device(config)# ipv6 dhcp pool pool1
Device(config-dhcp)# accounting list1
```

Example: Configuration for a Prefix Allocated from a Local Pool

DHCPv6 clients are connected to the DHCPv6 server on Gigabit Ethernet interface 0/0/0. The server is configured to use parameters from the DHCP pool called dhcp-pool. This pool provides clients with the IPv6 address of a DNS server and the domain name to be used. It also specifies that prefixes can be delegated from the prefix pool called client-prefix-pool1. The prefixes delegated will have valid and preferred lifetimes of 1800 and 600 seconds respectively. The prefix pool named client-prefix-pool1 has a prefix of length /40 from which it will delegate (sub) prefixes of length /48.

```
ipv6 dhcp pool dhcp-pool
 prefix-delegation pool client-prefix-pool1 lifetime 1800 600
 dns-server 2001:DB8:3000:3000::42
 domain-name example.com
!
interface GigabitEthernet0/0/0
 description downlink to clients
 ipv6 address FEC0:240:104:2001::139/64
 ipv6 dhcp server dhcp-pool
!
ipv6 local pool client-prefix-pool1 2001:DB8:1200::/40 48
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

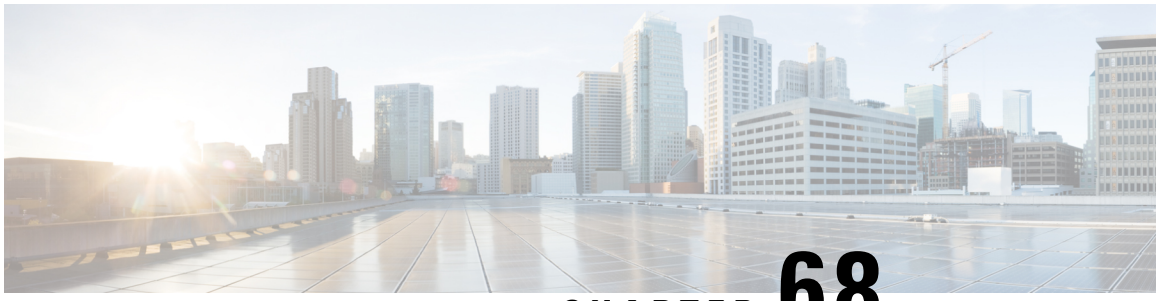
Feature Information for DHCP for IPv6 Broadband

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 100: Feature Information for DHCP for IPv6 Broadband

Feature Name	Releases	Feature Information
DHCP Enhancements to Support IPv6 Broadband Deployments	Cisco IOS XE Release 2.5	The feature highlights the DHCP enhancements that support IPv6 broadband deployments, such as, the different ways a delegating device selects prefixes for a requesting device, enabling accounting messages on a device, and forced release of delegated prefix bindings associated with a PPP virtual interface when the PPP virtual interface is terminated. The following commands were introduced or modified: **accounting**, **ipv6 dhcp bindings track ppp**, **ipv6 dhcp pool**.
DHCPv6 Prefix Delegation RADIUS VSA	Cisco IOS XE Release 2.5	When the user requests a prefix from the prefix delegator, typically the NAS, the prefix is allocated using DHCPv6.
DHCP Accounting Attribute	Cisco IOS XE Release 3.13S	The DHCP Accounting Attribute feature allows DHCPv6 to append delegated prefix information to accounting start and stop messages.



CHAPTER 68

DHCPv6 Server Stateless Autoconfiguration

Hierarchical Dynamic Host Configuration Protocol for IPv6 (DHCPv6) for stateless configuration parameters allows a stateless or stateful DHCPv6 client to export configuration parameters (DHCPv6 options) to a local DHCPv6 server pool. The local DHCPv6 server can then provide the imported configuration parameters to other DHCPv6 clients.

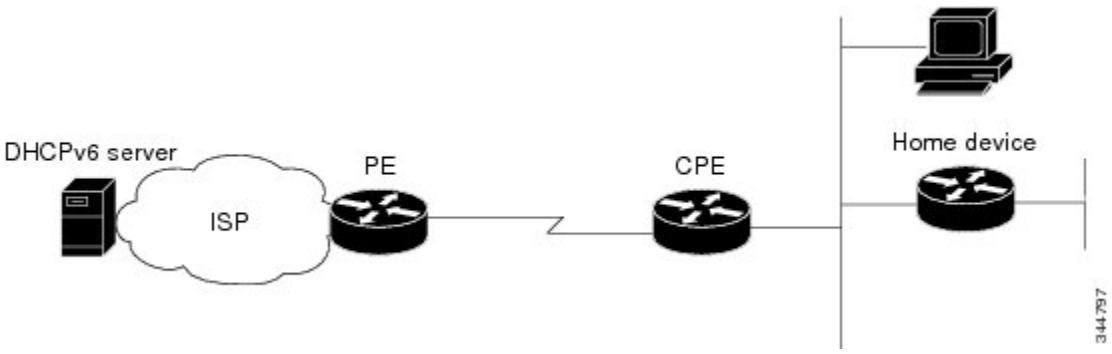
Information About DHCPv6 Server Stateless Autoconfiguration

DHCPv6 Server Stateless Autoconfiguration

Hierarchical Dynamic Host Configuration Protocol for IPv6 (DHCPv6) for stateless configuration parameters allows a stateless or stateful DHCPv6 client to export configuration parameters (DHCPv6 options) to a local DHCPv6 server pool. The local DHCPv6 server can then provide the imported configuration parameters to other DHCPv6 clients.

The figure below shows a typical broadband deployment.

Figure 74: Broadband Topology



The customer premises edge (CPE) interface toward the provider edge (PE) can be a stateless or stateful DHCPv6 client. In either case, the ISP-side DHCPv6 server might provide configuration parameters such as Domain Name System (DNS) server addresses, domain names, and Simple Network Time Protocol (SNTP) servers to the DHCP client on the CPE. Such information can be specific to ISPs.

In addition to being a DHCPv6 client (for example, toward the ISP), the CPE can act as a DHCPv6 server to the home network. For example, neighbor discovery followed by a stateless or stateful DHCPv6 client can occur on the link between the CPE and the home devices. In some cases, the information to be provided to the home network is the same as that obtained from the ISP-side DHCPv6 server. Because this information can be dynamically changed, it cannot be hard-configured in the CPE's configuration. Therefore, the DHCPv6 component on the CPE allows automatic importing of configuration parameters from the DHCPv6 client to the DHCPv6 server pool.

DHCPv6 supports the options for IPv6 on the server described in the following sections.

Information Refresh Server Option

The DHCPv6 information refresh server option can specify a maximum limit for the length of time a client should wait before refreshing the information retrieved from DHCPv6. This option is used with stateless DHCPv6 because there are no addresses or other entities with lifetimes that can tell the client when to contact the DHCPv6 server to refresh its configuration.

NIS- and NIS+-Related Server Options

Users can configure the network information service (NIS) or NIS plus (NIS+) address or domain name of a DHCPv6 server using NIS- and NIS+-related options, and then import that information to the DHCPv6 client.

SIP Server Options

Session Initiation Protocol (SIP) server options contain either a list of domain names or a list of IPv6 addresses that can be mapped to one or more SIP outbound proxy servers. One option carries a list of domain names, and the other option carries a list of 128-bit IPv6 addresses.

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions or calls. A SIP system has several logical components: user agents, proxy servers, redirect servers, and registrars. User agents can contain SIP clients; proxy servers always contain SIP clients.

SNTP Server Option

The Simple Network Time Protocol (SNTP) server option provides a list of one or more IPv6 addresses of SNTP servers available to the client for synchronization. Clients use these SNTP servers to synchronize their system time to that of the standard time servers. The DHCPv6 server can list the SNTP servers in decreasing order of preference, but clients treat the list of SNTP servers as an ordered list.

How to Configure DHCPv6 Server Stateless Autoconfiguration

Configuring the Stateless DHCPv6 Server

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **dns-server** *ipv6-address*
5. **domain-name** *domain*
6. **exit**
7. **interface** *type number*
8. **ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]
9. **ipv6 nd other-config flag**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool dhcp-pool`	Configures a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**dns-server** *ipv6-address* **Example:** `Device(config-dhcp)# dns-server 2001:DB8:3000:3000::42`	Specifies the Domain Name System (DNS) IPv6 servers available to a DHCPv6 client.
Step 5	**domain-name** *domain* **Example:** `Device(config-dhcp)# domain-name example.com`	Configures a domain name for a DHCPv6 client.
Step 6	**exit** **Example:** `Device(config-dhcp)# exit`	Exits DHCPv6 pool configuration mode, and returns the device to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface serial 3`	Specifies an interface type and number, and places the device in interface configuration mode.

	Command or Action	Purpose
Step 8	**ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**] **Example:** `Device(config-if)# ipv6 dhcp server dhcp-pool`	Enables DHCPv6 on an interface.
Step 9	**ipv6 nd other-config flag** **Example:** `Device(config-if)# ipv6 nd other-config flag`	Sets the "other stateful configuration" flag in IPv6 router advertisements (RAs).
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring the Stateless DHCPv6 Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **dns-server** *ipv6-address*
5. **domain-name** *domain*
6. **exit**
7. **interface** *type number*
8. **ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]
9. **ipv6 nd other-config flag**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool dhcp-pool`	Configures a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	**dns-server** *ipv6-address* **Example:** `Device(config-dhcp)# dns-server 2001:DB8:3000:3000::42`	Specifies the Domain Name System (DNS) IPv6 servers available to a DHCPv6 client.
Step 5	**domain-name** *domain* **Example:** `Device(config-dhcp)# domain-name example.com`	Configures a domain name for a DHCPv6 client.
Step 6	**exit** **Example:** `Device(config-dhcp)# exit`	Exits DHCPv6 pool configuration mode, and returns the device to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface serial 3`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 8	**ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**] **Example:** `Device(config-if)# ipv6 dhcp server dhcp-pool`	Enables DHCPv6 on an interface.
Step 9	**ipv6 nd other-config flag** **Example:** `Device(config-if)# ipv6 nd other-config flag`	Sets the “other stateful configuration” flag in IPv6 router advertisements (RAs).
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Enabling Processing of Packets with Source Routing Header Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 source-route**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 source-route** **Example:** `Device(config)# ipv6 source-route`	Enables processing of the IPv6 type 0 routing header.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for DHCPv6 Server Stateless Autoconfiguration

Example: Configuring the Stateless DHCPv6 Function

The following example shows how to use the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) function to configure clients with information about the name lookup system. The server is configured with a DHCP pool, which contains the name lookup information that is to be passed to clients. It does not need to contain a prefix pool. This DHCP pool is attached to the access link to customers (Ethernet 0/0) when you enter the **ipv6 dhcp server** command. The access link also has the **ipv6 nd other-config-flag** command enabled. Router advertisement (RA) messages sent from this interface inform clients that they should use DHCPv6 for “other” (for example, nonaddress) configuration information.

```
ipv6 dhcp pool dhcp-pool
 dns-server 2001:DB8:A:B::1
 dns-server 2001:DB8:3000:3000::42
 domain-name example.com
!
interface Ethernet 0/0
 description Access link down to customers
 ipv6 address 2001:DB8:1234:42::1/64
 ipv6 nd other-config-flag
 ipv6 dhcp server dhcp-pool
```

The client has no obvious DHCPv6 configuration. However, the **ipv6 address autoconfig** command on the uplink to the service provider (Ethernet 0/0) causes the following two events:

- Addresses are autoconfigured on the interface, based on prefixes in RA messages received from the server.
- If received RA messages have the “other configuration” flag set, the interface attempts to acquire the other (for example, nonaddress) configuration from any DHCPv6 servers.

Additional References for DHCP Overview

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
DHCP commands	Cisco IOS IP Addressing Services Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

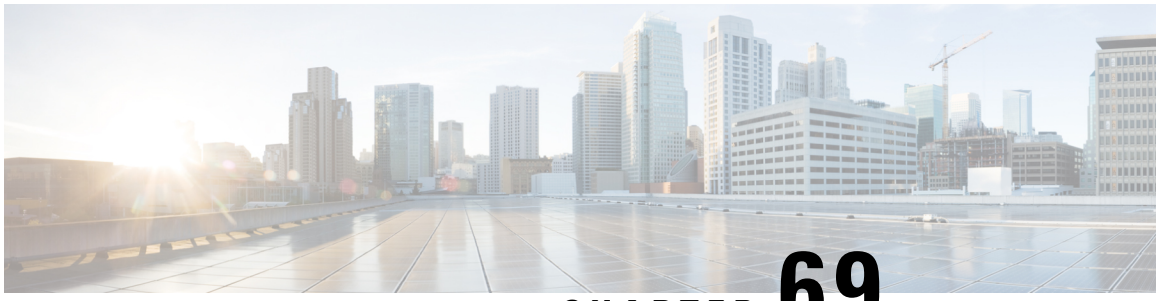
Feature Information for DHCPv6 Server Stateless Autoconfiguration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 101: Feature Information for DHCPv6 Server Stateless Autoconfiguration

Feature Name	Releases	Feature Information
DHCPv6 Server Stateless Autoconfiguration		Hierarchical DHCPv6 for stateless configuration parameters allows a stateless or stateful DHCPv6 client to export configuration parameters (DHCPv6 options) to a local DHCPv6 server pool. The following commands were introduced or modified: **dns-server**, **domain-name**, **ipv6 address autoconfig**, **ipv6 dhcp pool**, **ipv6 dhcp server**, **ipv6 nd other-config-flag**, **ipv6 source-route**.



CHAPTER 69

DHCP Server MIB

The DHCP Server MIB feature provides Simple Network Management Protocol (SNMP) access to and control of Cisco IOS Dynamic Host Configuration Protocol (DHCP) server software on a Cisco router by an external network management device.

Prerequisites for the DHCP Server MIB

SNMP must be enabled on the router before DHCP server trap notifications can be configured.

Information About the DHCP Server MIB

SNMP Overview

SNMP is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language that is used for monitoring and managing devices in a network.

SNMP defines two main types of entities: managers and agents. The SNMP manager is a system that controls and monitors the activities of network hosts using SNMP. The agent is the software component within a remote networking device that maintains the data and reports this data, as needed, to the manager. The manager and agent share a Management Information Base (MIB) that defines the information that the agent can make available to the manager.

An important feature of SNMP is the capability to generate unsolicited notifications from an SNMP agent. These trap notifications are messages alerting the SNMP manager to conditions on the network. Traps are considered an agent-to-manager function and a request for confirmation of receipt from the SNMP manager is not required.

DHCP Server Trap Notifications

DHCP server trap notifications are sent to the SNMP manager for the following events:

- Address utilization for a subnet has risen above or fallen below a configurable threshold.
- Address utilization for an address pool has risen above or fallen below a configurable threshold.
- A lease limit violation is detected. The lease limit configuration allows you to control the number of subscribers per interface.
- The DHCP server has started or stopped.
- A duplicate IP address is detected.

The DHCP Server MIB feature does not send the same type of trap notification back-to-back for the same threshold event. For example, if the low threshold value for available free addresses becomes equal to or less than the configured value, a free address low event trap notification on the subnet or pool is generated. This same trap notification will not be resent until the value for the available free addresses has exceeded the value of the free high threshold and vise versa. This threshold control mechanism applies to all trap notifications concerning thresholds in addition to the trap notifications for the DHCP server start and stop time and the lease limit violation. The duplicate IP address trap notification is not subject to this threshold control mechanism.

Tables and Objects in the DHCP Server MIB

The DHCP Server MIB consists of the following tables and objects. The first character of a row in the table begins with "c" (Cisco) and is mapped to the object defined in the IETF draft RFC, *Dynamic Host Configuration Protocol for IPv4 Server MIB.* If the information is not currently available in Cisco IOS software, the value in the second column is displayed as 0 (zero).

- cDhcpv4SrvSystemsObjects (see Table 7)--System description and object IDs
- cBootpHCCounterObjects (see Table 8)--BOOTP counter information
- cDhcpv4HCCounterObjects (see Table 9)--DHCPv4 counter information
- cDhcpv4ServerSharedNetTable (see Table 10)--DHCP address pool information
- cDhcpv4ServerSubnetTable (see Table 11)--Additional DHCP address pool subnet information including secondary subnet information
- cDhcpv4SrvExtSubnetTable (see Table 12)--Additional DHCP address pool subnet information
- cDhcpv4ServerNotifyObjectsGroup (see Table 13)--This objects group is used by the cDhcpv4ServerNotificationsGroup notifications group.
- cDhcpv4ServerNotificationsGroup (see Table 14)--This notifications group consists of all traps defined in the Cisco IOS DHCP server.
- cDhcpv4SrvExtNotifyGroup (see Table 15)--This notifications group consists of all traps not defined in the draft DHCPv4 Server MIB RFC.

Table 102: cDhcpv4SrvSystemsObjects and Descriptions

Name	Description
cDhcpv4SrvSystemDescr	Contains a textual description of the server (full name and version identification).
cDhcpv4SrvSystemObjectID	Cisco experiment node for the DHCP Server MIB. For example, 1.3.6.1.4.1.9.10.102...

Table 103: cBootpHCCounterObjects and Descriptions

Name	Description
cBootpHCCountRequests	The number of packets received that do contain a BOOTREQUEST message type in the first octet.
cBootpHCCountInvalids	0
cBootpHCCountReplies	The number of packets received that contain a BOOTREPLY message type in the first octet.
cBootpHCCountDroppedUnknown Clients	0
cBootpHCCountDroppedNotServingSubnet	0

Table 104: cDhcpv4HCCounterObjects and Descriptions

Name	Description
cDhcpv4HCCountDiscovers	The number of DHCPDISCOVER packets received.
cDhcpv4HCCountOffers	The number of DHCPOFFER packets sent.
cDhcpv4HCCountRequests	The number of DHCPREQUEST packets sent.
cDhcpv4HCCountDeclines	The number of DHCPDECLINE packets sent.
cDhcpv4HCCountAcks	The number of DHCPACK packets sent.
cDhcpv4HCCountNaks	The number of DHCPNACK packets sent.
cDhcpv4HCCountReleases	The number of DHCPRELEASE packets sent.
cDhcpv4HCCountInforms	The number of DHCPINFORM packets sent.
cDhcpv4HCCountForcedRenews	0
cDhcpv4HCCountInvalids	The number of DHCP packets received whose DHCP message type is not understood or handled by the DHCP server.
cDhcpv4HCCountDropUnknownClient	0
cDhcpv4HCCountDropNotServingSubnet	0

Table 105: cDhcpv4ServerSharedNetTable and Descriptions

Name	Description
cDhcpv4ServerSharedNetName	The DHCP address pool name.
cDhcpv4ServerSharedNetFreeAddr LowThreshold	This entry value corresponds to the **utilization mark high** command in DHCP pool configuration mode multiplied by the total pool addresses then divided by 100.
cDhcpv4ServerSharedNetFreeAddrHighThreshold	This entry value corresponds to the **utilization mark low** command in DHCP pool configuration mode multiplied by the total subnet addresses then divided by 100.
cDhcpv4ServerSharedNetFree Addresses	The number of IPv4 addresses that are available within this shared network.
cDhcpv4ServerSharedNetReserved Addresses	The number of IP addresses that are reserved for the pool (not available for assignment). This entry corresponds to the **ip dhcp excluded-address** global configuration command. The value is zero if no excluded addresses are defined for the pool.
cDhcpv4ServerSharedNetTotal Addresses	The number of IP addresses that are available within this shared network.

Table 106: cDhcpv4ServerSubnetTable and Descriptions

Name	Description
cDhcpv4ServerSubnetAddress	The IP address of the subnet entry in the table.
cDhcpv4ServerSubnetMask	The subnet mask of the subnet.
cDhcpv4ServerSubnetSharedNetworkName	The DHCP address pool name to which the subnet belongs.
cDhcpv4ServerSubnetFreeAddrLowThreshold	This entry value corresponds to the **override utilization high** command in DHCP pool secondary subnet configuration mode multiplied by the total subnet addresses then divided by 100.
cDhcpv4ServerSubnetFreeAddrHighThreshold	This entry value corresponds to the **override utilization low** command in DHCP pool secondary subnet configuration mode multiplied by the total subnet addresses then divided by 100.
cDhcpv4ServerSubnetFree Addresses	The number of free IP addresses that are available in the subnet.

Table 107: cDhcpv4SrvExtSubnetTable and Descriptions

Name	Description
cDhcpv4ServerDefaultRouterAddress	The entry corresponds to the **override default-router** command in DHCP pool secondary subnet configuration mode.
cDhcpv4ServerSubnetStartAddress	The first subnet IP address.
cDhcpv4ServerSubnetEndAddress	The last subnet IP address.

Table 108: cDhcpv4ServerNotifyObjectsGroups and Descriptions

Name	Description
cDhcpv4ServerNotifyDuplicateIpAddr	The IP address is found to be a duplicate. Duplicates are detected by servers who send a PING before offering an IP address lease or by a client sending a gratuitous ARP message reported through a DHCPDECLINE message.
cDhcpv4ServerNotifyDuplicateMac	The offending MAC address that caused a duplicate IPv4 address to be detected, if captured by the server, otherwise set to 00-00-00-00-00-00.
cDhcpv4ServerNotifyClientOrServerDetected	This object is set by the server to client if the client used DHCPDECLINE to mark the offered address as in use, or to server if the server discovered that address was in use by a client before offering it.
cDhcpv4ServerNotifyServerStart	The date and time when the server began operation, which is controlled by the **service dhcp** command.
cDhcpv4ServerNotifyServerStop	The date and time when the server ceased operation, which is controlled by **no service dhcp** command.

Table 109: cDhcpv4ServerNotificationsGroup and Descriptions

Name	Description
cDhcpv4ServerFreeAddressLow	This notification signifies that the number of available IP addresses for a DHCP address pool has fallen below the defined low threshold. This notification corresponds to the **snmp-server enable traps dhcp** global configuration command.
cDhcpv4ServerFreeAddressHigh	This notification signifies that the number of available IP addresses for a DHCP address pool has risen above the defined high threshold. This notification corresponds to the **snmp-server enable traps dhcp** global configuration command.
cDhcpv4ServerStartTime	This notification signifies that the server has started. This notification corresponds to the **service dhcp** and **snmp-server enable traps dhcp time**global configuration commands.

Name	Description
cDhcpv4ServerStopTime	This notification signifies that the server has stopped normally. This notification corresponds to the **no service dhcp** and **snmp-server enable traps dhcp time**global configuration commands.
cDhcpv4ServerDuplicateAddress	This notification signifies that a duplicate IP address has been detected. This notification corresponds to the **snmp-server enable traps dhcp duplicate**global configuration command.

Table 110: cDhcpv4SrvNotifyGroup and Descriptions

Name (not in the RFC draft)	Description
cDhcpv4ServerIfLeaseLimitExceeded	This notification signifies that a per interface lease limit is exceeded. This notification corresponds to the **snmp-server enable traps dhcp interface**global configuration command.
cDhcpv4ServerSubnetFreeAddressLow	This notification signifies that the number of available IP addresses for a subnet has fallen below the defined low threshold. This notification corresponds to the **snmp-server enable traps dhcp subnet**global configuration command.
cDhcpv4ServerSubnetFreeAddressHigh	This notification signifies that the number of available IPv4 addresses for a subnet has risen above the defined high threshold. This notification corresponds to the **snmp-server enable traps dhcp subnet**global configuration command.

How to Enable DHCP Trap Notifications

Configuring the Router to Send SNMP Trap Notifications About DHCP

DHCP trap notifications are disabled by default. The trap notification is disabled if the corresponding trap configuration is not enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps dhcp duplicate**] [**interface**] [**pool**] [**subnet**] [**time**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps dhcp duplicate**] [**interface**] [**pool**] [**subnet**] [**time** **Example:** `Router(config)# snmp-server enable traps dhcp`	Enables the sending of DHCP SNMP trap notifications. • **duplicate** --Sends notification about duplicate IP addresses. • **interface** --Sends notification that a per interface lease limit is exceeded. • **pool** --Sends notification when address utilization for an address pool has risen above or fallen below a configurable threshold. • **subnet** --Sends notification when address utilization for a subnet has risen above or fallen below a configurable threshold. • **time** --Sends notification that the DHCP server has started or stopped. • If you specify the **snmp-server enables traps dhcp** command without any of the optional keywords, all DHCP trap notifications are enabled.
Step 4	**end** **Example:** `Router(config)# end`	Returns the router to privileged EXEC mode.

Troubleshooting Tips

If you are using secondary IP addresses under a single loopback interface and using secondary subnets under a DHCP pool, use one DHCP pool to configure networks for all the secondary subnets instead of using one pool per secondary subnet. The **network** *network-number* [*mask* | */prefix-length*] [**secondary**] command must be configured under a single DHCP address pool rather than multiple DHCP address pools.

The following is the correct configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 network 172.16.2.0 255.255.255.0 secondary
 network 172.16.3.0 255.255.255.0 secondary
 network 172.16.4.0 255.255.255.0 secondary
!
interface Loopback111
```

```
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

The following is the incorrect configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_2
 network 172.16.2.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_3
 network 172.16.3.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_4
 network 172.16.4.0 255.255.255.0
 lease 1 20 30
 accounting default
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

Configuration Examples for the DHCP Server MIB

DHCP Server MIB--Secondary Subnet Trap Example

The following example configures 192.0.2.0/24 as the subnetwork number and mask of the DHCP pool named pool2 and then adds the DHCP pool secondary subnet specified by the subnet number and mask 192.0.4.0/30. The IP addresses in pool2 consist of two disjoint subnets: the addresses from 192.0.2.1 to 192.0.2.254 and the addresses from 192.0.4.1 to 192.0.4.2.

The address pool utilization mark, configured at the global level, will be overridden at the secondary subnet level. A trap is sent to the SNMP manager if the subnet size of the secondary subnet exceeds or goes below the level specified by the **override utilization** commands.

The **utilization mark**{**high**| **low**}**log** command enables a system message to be generated for a DHCP address pool or secondary subnet when the utilization exceeds the configured high utilization threshold or falls below the configured low utilization threshold.

```
!
ip dhcp pool pool2
 utilization mark high 80 log
 utilization mark low 70 log
 network 192.0.2.0 255.255.255.0
 network 192.0.4.0 255.255.255.252 secondary
```

```
 override utilization high 40
 override utilization low 30
!
snmp-server enable traps dhcp subnet
```

DHCP Server MIB--Address Pool Trap Example

In the following example, if the address utilization exceeds the high threshold or drops below the low threshold, an SNMP trap will be sent to the SNMP manager and a system message will be generated.

```
ip dhcp pool pool3
 utilization mark high 80 log
 utilization mark low 70 log
!
snmp-server enable traps dhcp pool
```

DHCP Server MIB--Lease Limit Violation Trap Example

In the following example, four DHCP clients are allowed to receive IP addresses. If a fifth client tries to obtain an IP address, the DHCPDISCOVER messages will not be forwarded to the DHCP server and a trap will be sent to the SNMP manager.

```
ip dhcp limit lease log
interface Serial 0/0
 ip dhcp limit lease 4
 exit
snmp-server enable traps dhcp interface
```

Additional References

The following sections provide references related to the DHCP Server MIB feature.

Related Documents

Related Topic	Document Title
SNMP configuration tasks	"Configuring SNMP Support" module
DHCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
DHCP server configuration tasks including subnet utilization tasks	"Configuring the Cisco IOS DHCP Server" module
DHCP per interface lease limit functionality	"Configuring DHCP Services for Accounting and Security" module
DHCP ODAP tasks including address pool utilization tasks	"Configuring the DHCP Server On-Demand Address Pool Manager" module

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
• CISCO-IETF-DHCP-SERVER-MIB • CISCO-IETF-DHCP-SERVER-EXT-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
Draft RFC: draft-ietf-dhc-server-mib-10.txt	Dynamic Host Configuration Protocol for IPv4 (DHCPv4) Server MIB

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

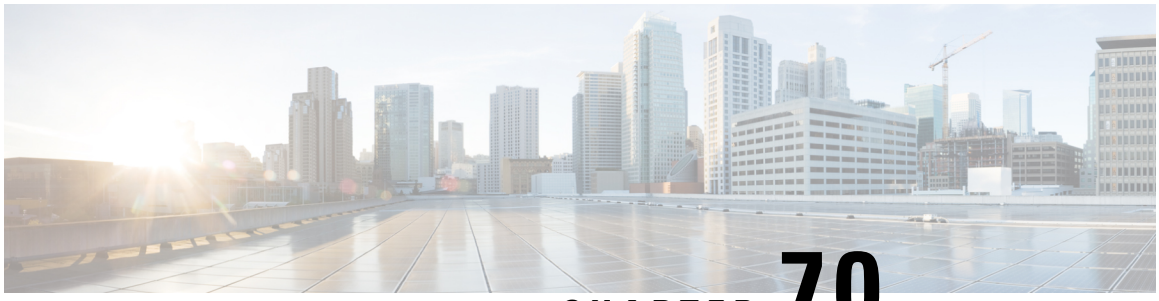
Feature Information for DHCP Server MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 111: Feature Information for DHCP Server MIB

Feature Name	Releases	Feature Information
DHCP Server MIB		The DHCP Server MIB feature provides SNMP access to and control of Cisco IOS DHCP server software on a Cisco router by an external network management device. The following commands were introduced by this feature: **snmp-server enable traps dhcp**and **debug ip dhcp server snmp**.



CHAPTER 70

Asymmetric Lease for DHCPv4 Relay

Restrictions for Asymmetric Lease for DHCPv4 Relay

- Asymmetric lease is supported only for the DHCP relay agent. It is not applicable to DHCP Server.
- When there is a failover from Active to Standby RP on the short lease configured relay agent, the short lease data is lost, and the relay agent only forwards DHCP messages.
- Liveness detection of clients by the relay agent is not supported.
- Notifications to server by relay agent about inactive clients is not supported.
- The server allotted T1 or T2 values should not be zero and the configured short lease value must be less than server allotted T1 value.
- The minimum short lease value on DHCP relay agent must be 60 seconds.
- Clients connected before activating short lease on relay come outside the purview of this feature. Clients connected after activation will be the short lease clients.

Information about Asymmetric Lease for DHCPv4 Relay

Asymmetric lease or short lease is an assigned lease that is shorter than the actual lease granted by the server. You can configure the short lease on a relay agent, which will cause the relay agent to act as the DHCP server proxy for a certain interval. The short lease provides a rebinding or restarting solution for quick failure detection and failover of relay agent. Additionally, short lease is an option to force a lease renewal for clients before the original lease expires. It detects the lease expiry early and helps to keep the clients status live.

DHCPv4 IP Assignment with Asymmetric Lease

On receiving an ACK message, the relay agent does the following:

1. Extracts and stores the server assigned the T1 and T2 values.
2. Validates the configured short lease value with the server that is assigned the T1 and T2 values.
3. Modifies the ACK message and replaces the T1 and T2 values with new values defined as per short lease configuration.

Derivation of Short Lease T1' and T2' values

The T1 also referred to as Renew Timer is the time at which the client renews the lease. T2 or Rebind Timer represents the time at which client tries to rebind. The client sends the renewal message to the server which in turn provides the client with its addresses and configuration parameters.

The rebind message is sent after a client receives no response to a renewal message. Both T1 and T2 are time duration relative to the current time expressed in units of seconds.

On the DHCPv4 Relay Agent, the minimum allowed short lease value is 60 seconds. The T1' can be assigned with the configured short lease value. T2' is derived from T1' as shown below:

T2' = minimum (2 * T1 * 0.8, DHCPv4 Server assigned T2 value)

Renewing and Rebinding Scenarios

The relay agent modifies the ACK received from server, includes the short lease value configured on relay instead of the actual lease value and forwards the modified packet to the client.

When DHCP client requests for a short lease renewal, the relay agent does not pass the request directly to the server. Instead, the relay agent creates and sends an acknowledgement packet by adding a short lease value from the saved information.

The relay agents will continue to reply to the client renew requests until the actual lease renewal time (T1) expires. When the actual lease renewal time expires, the short lease value is no longer valid, and any subsequent renewals are directly forwarded to the server.

If the server acknowledges the request, the lease is extended, and the process starts with the relay agent responding to client's short lease renewals with ACK packets.

When a renewal request fails, the client starts sending rebind request messages and the relay agent acknowledges these messages until actual lease rebind time (T2) expires. This creates an acknowledgement and extends the short lease. If the packet arrives after T2 expires, the short lease value is invalidated, and the packet is forwarded to the server. When the server acknowledges the request, the lease is extended, and the relay agent responds to the client's short lease renewals with ACK packets.

The following sequence diagrams depicts the short lease renewal and rebinding in various scenarios:

Figure 75: Short lease renewal before the server assigned T1 expires

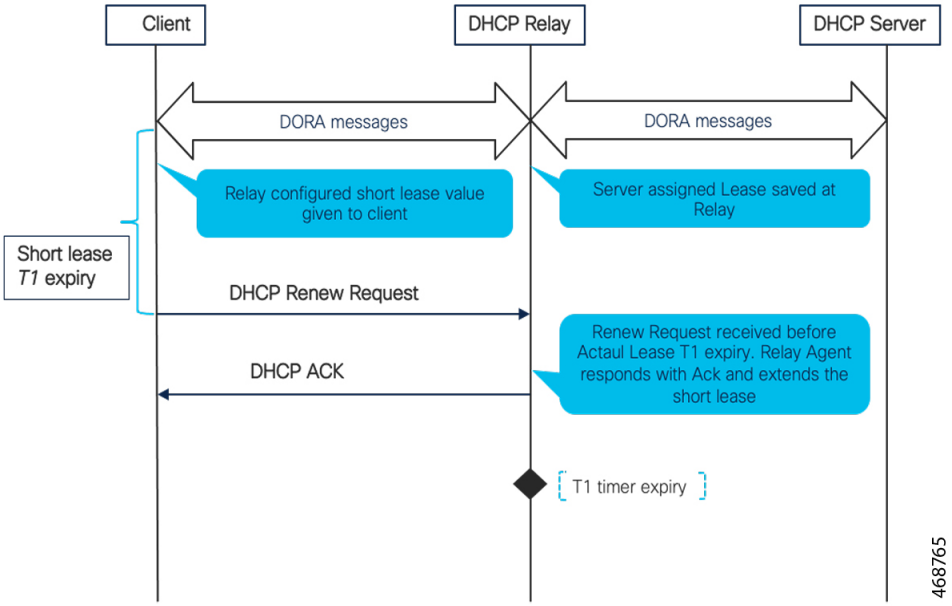


Figure 76: Short lease renewal after the server assigned T1 expiry

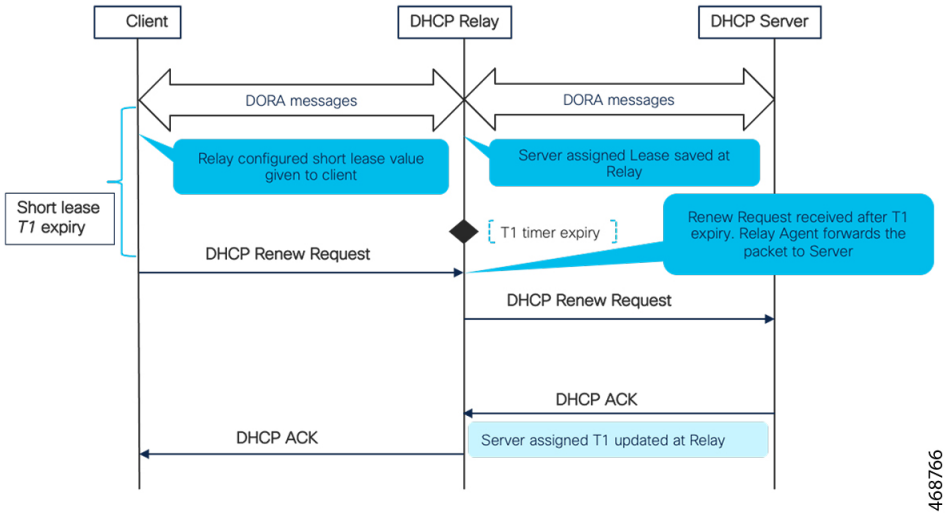


Figure 77: Short lease rebind scenario

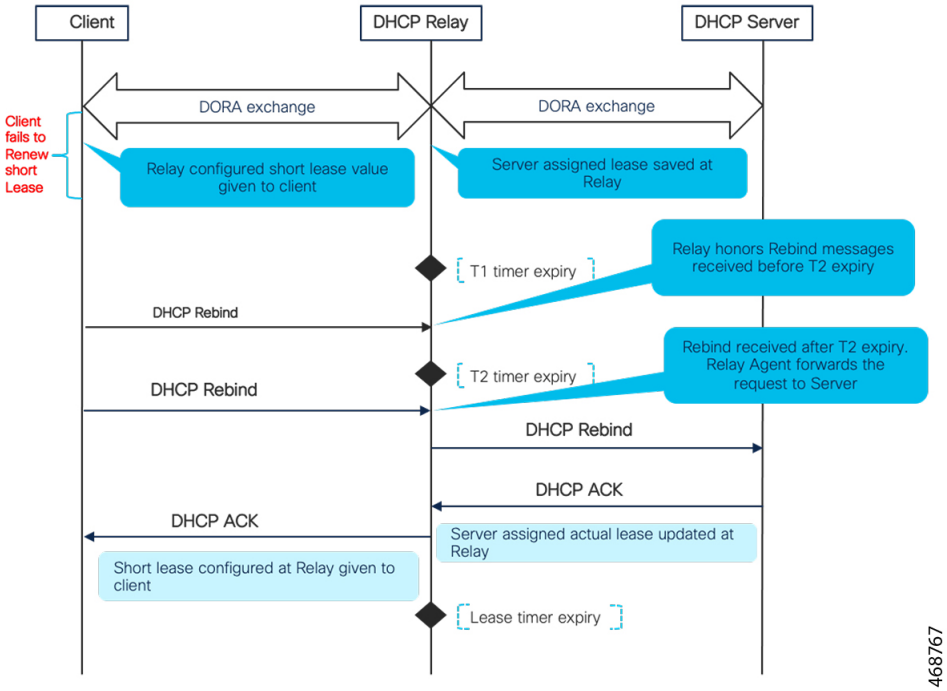


Figure 78: Short lease when the server is down

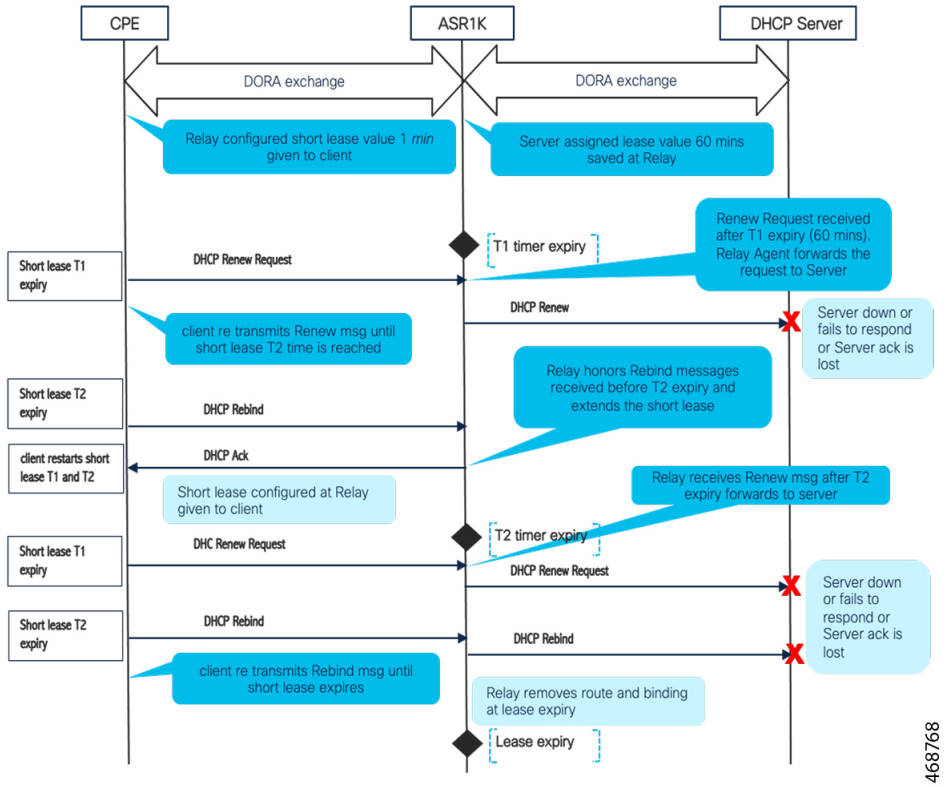
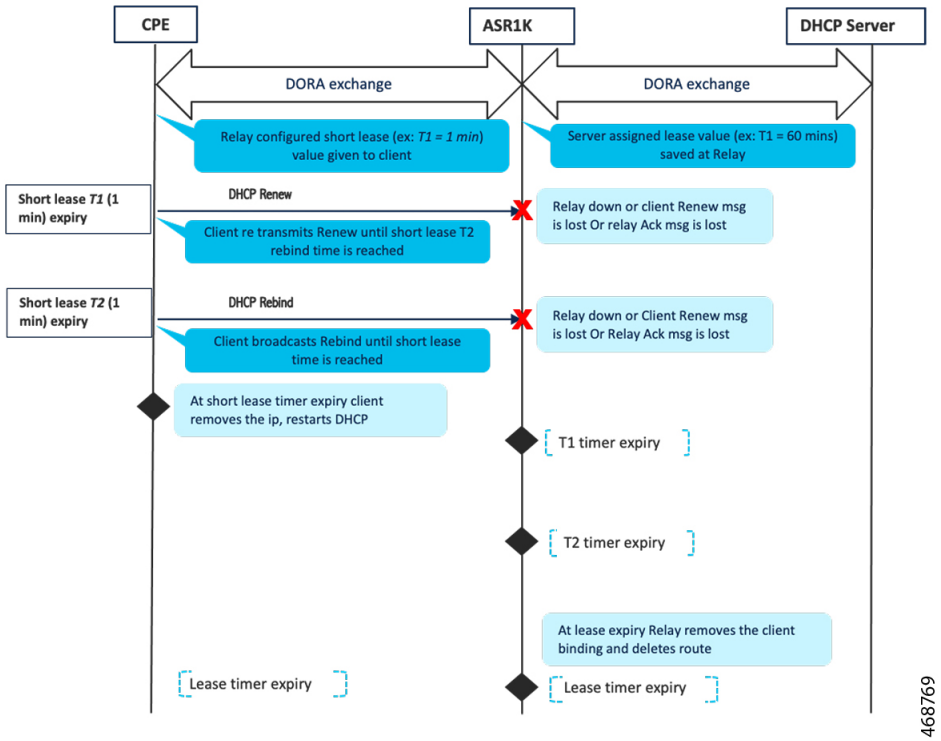


Figure 79: Short lease renew and rebind handling when the Relay Agent is down



SSO and ISSU Support

Note Only short-lease Configuration is synchronized between the active and the standby and short-lease operational data is not synchronized to standby.

The DHCP relay agent detects when the active RP is failing over to the standby RP and keeps the states related to interfaces. Only when configuration sync happens, operational data including the binding is not synchronised. Hence, any subsequent Renew or Rebind request from a client will be forwarded to the Server. Relay (new Active) will create the binding and establish short lease data for each client when server responds with ACK just like new clients.

Configuring Asymmetric Lease for DHCPv4 Relay

You can apply the Asymmetric lease configuration on per interface or globally for all interfaces.

- Configuring Asymmetric Lease on an Interface
- Configuring Asymmetric Lease in Global Configuration Mode

Configuring Asymmetric Lease on an Interface for DHCPv4 Relay

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip dhcp relay shortlease** *time in seconds*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ip dhcp relay shortlease** *time in seconds* **Example:** `Router(config-if)# ip dhcp relay short-lease 500`	Sets and enables the short lease for the client on the interface. You can set the lease time in seconds. The range is from 60 to 3600 seconds.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring Asymmetric Lease in Global Configuration Mode for DHCPv4 Relay

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp-relay short-lease** *time in seconds*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp-relay short-lease** *time in seconds* **Example:** `Router(config-if)# ip dhcp relay short-lease 500`	Sets and enables the short lease for the client globally. You can set the lease time in seconds. The range is from 60 to 3600 seconds.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for the Asymmetric Lease for DHCPv4 Relay

Example: Configuring the Asymmetric Lease on an Interface for DHCPv4 Relay

The following example demonstrates the configuration of the 'ip dhcp relay short-lease' command to allow short lease configuration on an interface level:

```
Router # configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
Router (config)# interface Ethernet0/0.100
Router (config-subif)# ip dhcp relay ?
information DHCP relay information option sourceinterface
Set source interface for relayed messages short-lease
Set and enable short lease for clients
Router(config-if)#ip dhcp relay short-lease ?
<60-3600> Short Lease in Seconds
Router(config-if)#ip dhcp relay short-lease 500 ?
<cr> <cr>
```

Example: Configuring the Asymmetric Lease in Global Configuration Mode for DHCPv4 Relay

The following example demonstrates the configuration of the 'ip dhcp relay short-lease' command to allow short lease configuration on a global level:

```
Router# configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#ip dhcp-relay ? information Relay agent
information option global configuration source-interface
Set source
interface for relayed messages short-lease
Set and enable Short
Lease for Client
Router(config)#ip dhcp-relay short-lease ?
<60-3600> Short Lease in Seconds
Router(config)#ip dhcp-relay short-lease 300 ?
<cr> <cr>
Router# show running-config | sec short-lease ip
dhcp-relay short-lease 300
```

Verifying the Configuration

To verify the short lease configuration, enable both the **debug ip dhcp server events** and **debug ip dhcp server packet details** commands on the relay device.

The following logs will be generated when the DHCP client requests IP address:

```
*Jun 22 03:06:57.686: DHCPD: forwarding BOOTREPLY to client
0063.6973.636f.2d61.6162.622e.6363.3030.2e30.3330.302d.4574.30
2f.30.
*Jun 22 03:06:57.686: DHCPD: Forwarding reply while saving
lease state
*Jun 22 03:06:57.686: DHCPD: Keeping state: Received DHCPACK
*Jun 22 03:06:57.686: DHCPD: lease time = 7200
*Jun 22 03:06:57.686: DHCPD: Server ID saved in Binding =
10.0.0.1
*Jun 22 03:06:57.686: DHCPD: Giaddr Address = 20.0.0.1
*Jun 22 03:06:57.686: DHCPD: Updated short lease data
T1' = 500, T2' = 800 T1 = 3600 T2 = 6300 Lease = 7200
*Jun 22 03:06:57.686: DHCPD: Updated short-lease T1 to 500 and
T2 to 800 in BOOTREPLY
The highlighted logs show that T1 and T2 values were updated to
user configured values (T1' = 500, T2' = 800).
the `show ip dhcp binding` command displays the relay binding created for the short lease.
Router#show ip dhcp binding Bindings from all
pools not associated with VRF:
IP address Client-ID/ Lease expiration
Type State Interface
Hardware address/
User name
20.0.0.2 aabb.cc00.0300 Jun 22 2022 10:36 AM
Relay Active Ethernet0/1 Router#
```

Feature Information for Asymmetric Lease for DHCPv4 Relay

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

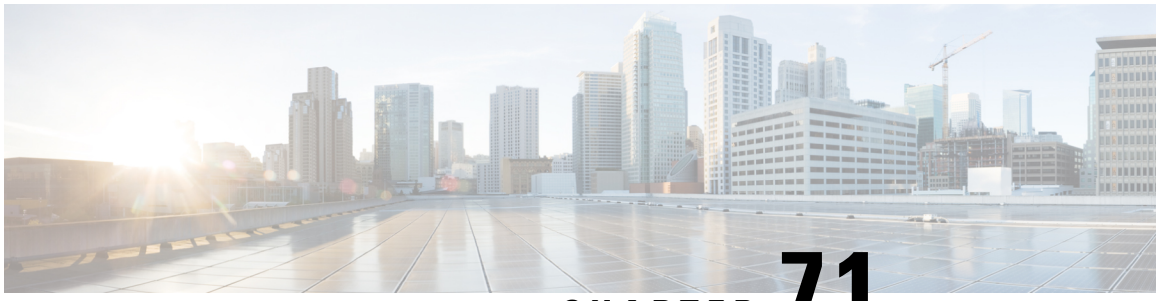
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 112: Feature Information for Asymmetric Lease for DHCPv4 Relay

Feature Name	Releases	Feature Information
Asymmetric Lease for DHCPv4 Relay		This feature allows you to manage or change the lease renewal. It provides options to force renewal of lease and also detects when the lease is nearing the expiry date.

PART VII

DNS



CHAPTER 71

Configuring DNS

The Domain Name System (DNS) is a distributed database in which you can map host names to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated host name. The Cisco IOS XE software maintains a cache of host-name-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. This cache speeds the process of converting names to addresses.

Prerequisites for Configuring DNS

To use DNS, you must have a DNS name server on your network.

Information About DNS

DNS Overview

If your network devices require connectivity with devices in networks for which you do not control name assignment, you can assign device names that uniquely identify your devices within the entire internetwork. The global naming scheme of the Internet, the DNS, accomplishes this task. This service is enabled by default. The following sections summarize DNS concepts and function:

Host Names for Network Devices

Each unique IP address can have an associated host name. DNS uses a hierarchical scheme for establishing host names for network nodes. This allows local control of the segments of the network through a client-server scheme. The DNS system can locate a network device by translating the host name of the device into its associated IP address.

Domains Names for Groups of Networks

IP defines a naming scheme that allows a device to be identified by its location in the IP. This is a hierarchical naming scheme that provides for *domains*. On the Internet, a domain is a portion of the naming hierarchy tree that refers to general groupings of networks based on organization type or geography. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that the IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the File Transfer Protocol (FTP) system, for example, is identified as *ftp.cisco.com*.

Name Servers

To keep track of domain names, IP has defined the concept of a *name server*. Name servers are programs that have complete information about their namespace portion of the domain tree and may also contain pointers to other name servers that can be used to lead to information from any other part of the domain tree. Name servers know the parts of the domain tree for which they have complete information. A name server may also store information about other parts of the domain tree. To map domain names to IP addresses, you must first identify the host names, then specify a name server, and enable the DNS service.

Cache

To speed the process of converting names to addresses, the name server maintains a database, called a *cache*, of host-name-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. The cache stores the results from previous responses. Upon receiving a client-issued DNS query, it will check this local storage to see if the answer is available locally.

Name Resolvers

Name resolvers are programs that extract information from name servers in response to client requests. Resolvers must be able to access at least one name server. The resolver either uses that name server's information to answer a query directly or pursues the query using referrals to other names servers. A resolver will typically be a system routine that is directly accessible to user programs. Therefore, no protocol is necessary between the resolver and the user program.

Zones

The domain namespace is divided into areas called zones that are points of delegation in the DNS tree. A zone contains all domains from a certain point downward, except those for which other zones are authoritative.

Authoritative Name Servers

A name server is said to be an authority for the parts of the domain tree for which it has complete information. A zone usually has an authoritative name server, often more than one. An authoritative name server has been configured with host table information or has acquired host table information though a zone transfer (the action that occurs when a secondary DNS server starts up and updates itself from the primary server).

DNS Operation

Within an organization, you can have many name servers, but Internet clients can query only those that the root name servers know. The other name servers answer internal queries only.

A name server handles client-issued queries to the DNS server for locally defined hosts within a particular zone as follows:

- An authoritative name server responds to DNS user queries for a domain name that is under its zone of authority by using the permanent and cached entries in its own host table. If the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server simply replies that no such information exists..
- A name server that is not configured as the authoritative name server responds to DNS user queries by using information that it has cached from previously received query responses. If no device is configured as the authoritative name server for a zone, queries to the DNS server for locally defined hosts will receive nonauthoritative responses.

Name servers answer DNS queries (forward incoming DNS queries or resolve internally generated DNS queries) according to the forwarding and lookup parameters configured for the specific domain.

DNS Security

An alternating sequence of DNS public key (DNSKEY) RR sets and Delegation Signer (DS) RR sets forms a chain of signed data, with each link in the chain vouching for the next. A DNSKEY RR is used to verify the signature covering a DS RR and allows the DS RR to be authenticated. The DS RR contains a hash of another DNSKEY RR and this new DNSKEY RR is authenticated by matching the hash in the DS RR.

DNS Views

A DNS view is a set of parameters that specify how to handle a DNS query. A DNS view defines the following information:

- Association with a VRF
- Parameters for resolving internally generated DNS queries
- Parameters for forwarding incoming DNS queries
- Internal host table for answering queries or caching DNS responses

Note The maximum number of DNS views and view lists depends on the memory of Cisco device. Configuring a large number of DNS views and view lists uses more device memory, and configuring a large number of views in the view lists uses more device processor time. For optimum performance, configure views and view list members that are required to support your Split DNS query forwarding or query resolution needs.

Restricted View Use Queries from the Associated VRF

A DNS view is always associated with a VRF— the global VRF or a named VRF, so as to limit the view usage in handling DNS queries that arrive on an interface matching a particular VRF:

- A DNS view that is associated with the global VRF can be used only to handle DNS queries that arrive on an interface in the global address space.
- A DNS view that is associated with a named VRF can be used only to handle DNS queries that arrive on an interface that matches the VRF with which the view is associated.

Note Additional restrictions (described in DNS Views) can be placed on a view after it has been defined. Also, a single view can be referenced multiple times, with different restrictions added in each case. However, because the association of a DNS view with a VRF is specified in the DNS view definition, the VRF-specific view-use limitation is a characteristic of the DNS view definition itself and cannot be separated from the view.

Parameters for Resolving Internally Generated DNS Queries

- Domain lookup—Enabling or disabling of DNS lookup to resolve hostnames for internally generated queries.
- Default domain name—Default domain to append to hostnames without a dot.
- Domain search list—List of domain names to try for hostnames without a dot.
- Domain name for multicast lookups—IP address to use for multicast address lookups.
- Domain name servers—List of name servers to use to resolve domain names for internally generated queries.
- Resolver source interface—Source interface to use to resolve domain names for internally generated queries.
- Round-robin rotation of IP addresses—Enabling or disabling of the use of a different IP address associated with the domain name in cache each time hostnames are looked up.

Parameters for Forwarding Incoming DNS Queries

The following parameters define how to forward incoming DNS queries:

- Forwarding of queries—Enabling or disabling of forwarding of incoming DNS queries.
- Forwarder addresses—List of IP addresses to use to forward incoming DNS queries.
- Forwarder source interface—Source interface to use to forward incoming DNS queries.

Sometimes, when a source interface is configured on a device with the split DNS feature to forward DNS queries, the device does not forward the DNS queries through the configured interface. Hence, consider the following points while forwarding the DNS queries using the source interface:

- DNS queries are forwarded to a broadcast address when a forwarding source interface is configured and the DNS forwarder is not configured.
- The source IP address of the forwarded query should be set to the primary IP address of the interface configured, using the **dns forwarding source-interface** *interface* command. If no such configuration exists, then the source IP address of the forwarded DNS query will be the primary IP address of the

outgoing interface. DNS forwarding should be done only when the source interface configured for the DNS forwarding is active.

- The source IP address of the DNS query for the DNS resolver functionality is set using the **domain resolver source-interface** *interface-type number* command. If there is no DNS address configured, then queries will be broadcasted to the defined source interface. DNS resolving should be done only when the source interface configured for the DNS resolving is active. See "Specifying a Source Interface to Forward DNS Queries" for the configuration steps.

DNS View Lists

A DNS view list is an ordered list of DNS views in which additional usage restrictions can be specified for any individual member in the list. The scope of these optional usage restrictions is limited to a specific member of a specific DNS view list. When the device must respond to a DNS query, the Cisco IOS software uses a DNS view list to select the DNS view that will be used to handle a DNS query.

Note The maximum number of DNS views and view lists supported is not specifically limited but is dependent on the amount of memory on the Cisco device. Configuring a larger number of DNS views and view lists uses more device memory, and configuring a larger number of views in the view lists uses more device processor time. For optimum performance, configure no more views and view list members than needed to support your Split DNS query forwarding or query resolution needs.

Order in Which to Check the Members of a DNS View List

When a DNS view list is used to select a DNS view for handling a given DNS query, the Cisco IOS software checks each member of the view list--in the order specified by the list--and selects the first view list member whose restrictions permit the view to be used with the query that needs to be handled.

Usage Restrictions Defined for a DNS View in the View List

A DNS view list member can be configured with usage restrictions defined using access control lists (ACLs) that specify rules for selecting that view list member based on the query hostname or the query source host IP address. The two types of ACLs supported by the Split DNS view list definition are described in "DNS Name Groups".

Note Multiple DNS view lists can be defined so that, for example, a given DNS view can be associated with different restrictions in each list. Also, different DNS view lists can include different DNS views.

Selection of the DNS View List

When the device that is acting as the DNS caching name server needs to respond to a DNS query, the Cisco IOS software uses a DNS view list to determine which DNS view can be used to handle the query:

- If the device is responding to an incoming query that arrives on an interface for which a DNS view list is configured, the interface-specific DNS view list is used.

- If the device is responding to an incoming query that arrives on an interface for which no specific DNS view list is configured, the default DNS view list is used.

If the device is responding to an internally generated query, no DNS view list is used to select a view; the global DNS view is used to handle the query.

The assignment of a DNS view list as the default or to an interface is described in "DNS View Groups".

Selection of a DNS View List Member

The view list members are compared, each in turn, to the characteristics of the DNS query that the device is responding to:

1. If the query is from a different VRF than the view, the view cannot be used to address the query, so the view-selection process moves on to the next member of the view list.
2. The specification of additional view-use restrictions is an optional setting for any view list member.

If the query list does not specify additional restrictions on the view, the view will be used to address the query, so the view-selection process is finished.

If the view list does specify additional restrictions on the view, the query is compared to those restrictions:

- If the query characteristics fail any view-use restriction, the view cannot be used to address the query, so the view-selection process moves on to the next member of the view list.
- If the query characteristics pass all the view-use restrictions, the view will be used to address the query. The view-selection process is finished.
- If the view-selection process reaches the end of the selected DNS view list without finding a view list member that can handle the query, the device discards the query.

The first DNS view list member that is found to have restrictions that match the query characteristics is used to handle the query.

DNS Name Groups

The Split DNS feature supports two types of ACLs that can be used to restrict the use of a DNS view. A DNS name list or a standard IP ACL (or both) can be applied to a DNS view list member to specify view-use restrictions in addition to the VRF-specific restriction that is a part of the view definition itself.

Note In this context, the term “group” is used to refer to the specification of a DNS name list or a standard IP ACL as a usage restriction on a view list member.

DNS View Usage Restrictions Based on the Query Hostname

A DNS name list is a named set of hostname pattern-matching rules, with each rule specifying the type of action to be performed if a query hostname matches the text string pattern in the rule. In order for a query hostname to match a name list, the hostname must match a rule that explicitly permits a matching pattern but the hostname cannot match any rules that explicitly deny a matching pattern.

DNS View Usage Restrictions Based on the Query Source IP Address

A standard IP ACL is a numbered or named set of host IP address-matching rules, with each rule specifying the type of action to be performed if an IP address matches the text string pattern in the rule. The Split DNS feature supports the use of a standard ACL as a view-use restriction based on the query source IP address. In order for a source IP address to match a name list, the IP address must match a rule that explicitly permits a matching pattern but the IP address cannot match any rules that explicitly deny a matching pattern.

DNS View Groups

The Split DNS feature provides two ways to specify the DNS view list that the Cisco IOS software is to use to select the DNS view that will be used to handle an incoming DNS query. For a query that arrives on an interface that is configured to use a particular DNS view list, the interface-specific DNS view list is used. Otherwise, the default DNS view list is used.

Note In this context, the term "group" refers to the specification of a DNS view list as an interface-specific DNS view list or the default view list for the device.

Interface-specific View Lists

A DNS view list can be attached to a device interface. When an incoming DNS query arrives on that interface, the Cisco IOS software uses that view list to select a DNS view to use to handle the query.

Default DNS View List

A DNS view list can be configured as the default DNS view list for the device. When an incoming DNS query arrives on an interface that is not configured to use a specific view list, the Cisco IOS software uses the default view list to select the DNS view to use to handle the query.

How to Configure DNS

Mapping Host Names to IP Addresses

Perform this task to associate host names with IP addresses.

A name server is used to keep track of information associated with domain names. A name server can maintain a database of host name-to-address mappings. Each name can map to one or more IP addresses. In order to use this service to map domain names to IP addresses, you must specify a name server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host** *name* [*tcp-port-number*] *address1* [*address2 ... address8*] [*mx* ns srv]
4. Do one of the following:
    - **ip domain name** *name*

- **ip domain list** *name*

5. **ip name-server** *server-address1* [*server-address2 ... server-address6*]
6. **ip domain lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip host** *name* [*tcp-port-number*] *address1* [*address2 ... address8*] [*mx* ns srv]
	Example:
	`Device(config)# ip host cisco-rtp 192.168.0.148`
	`Device(config)# ip host test mx 1 mx_record`
	`Device(config)# ip host test ns ns_record`
	`Device(config)# ip host test srv 0 0 0 srv_record`
		Defines a static host name-to-address mapping in the host name cache.
		• Typically, it is easier to refer to network devices by symbolic names rather than numerical addresses (services such as Telnet can use host names or addresses). Host names and IP addresses can be associated with one another through static or dynamic means.
		• Manually assigning host names to addresses is useful when dynamic mapping is not available.
		• Mail exchanger (mx) identifies the mail server that is responsible for handling e-mails for a given domain name.
		• Name server (ns) state the authoritative name servers for the given domain.
		• Service (srv) records specifies the location of a service.

Step 4	Do one of the following:
	• **ip domain name** *name*
	• **ip domain list** *name*
	Example:
	`Device(config)# ip domain name cisco.com`
	Example:
	`Device(config)# ip domain list cisco1.com`
		(Optional) Defines a default domain name that the Cisco IOS XE software will use to complete unqualified host names.
		or
		(Optional) Defines a list of default domain names to complete unqualified host names.
		• You can specify a default domain name that the Cisco IOS XE software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any host name that does not contain a complete domain name will have the

	Command or Action	Purpose
		default domain name you specify appended to it before the name is looked up. **Note** If there is no domain list, the domain name that you specified with the **ip domain name** global configuration command is used. If there is a domain list, the default domain name is not used. The **ip domain list** command is similar to the **ip domain name** command, except that with the **ip domain list** command you can define a list of domains, each to be tried in turn until the system finds a match.
Step 5	**ip name-server** *server-address1* [*server-address2 ... server-address6*] **Example:** `Device(config)# ip name-server 172.16.1.111 172.16.1.2`	Specifies one or more hosts that supply name information. • Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS.
Step 6	**ip domain lookup** **Example:** `Device(config)# ip domain lookup`	(Optional) Enables DNS-based address translation. • DNS is enabled by default. Use this command if DNS has been disabled.

What to do next

The name lookup system can be statically configured using the commands described in this task. Some other functions in Cisco IOS XE, such as DHCP can dynamically modify the state of the name lookup system. Use the **show hosts** command to display the cached host names and the DNS configuration.

Disabling DNS Queries for ISO CLNS Addresses

Perform this task to disable DNS queries for ISO CLNS addresses.

If your device has both IP and ISO Connectionless Network Service (ISO CLNS) enabled and you want to use ISO CLNS network service access point (NSAP) addresses, you can use the DNS to query these addresses, as documented in RFC 1348. This feature is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip domain lookup nsap**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip domain lookup nsap** **Example:** `Device(config)# no ip domain lookup nsap`	Disables DNS queries for ISO CLNS addresses.

Verifying DNS

Perform this task to verify your DNS configuration.

SUMMARY STEPS

1. **enable**
2. **ping** *host*
3. **show hosts**
4. **debug ip domain**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** *host* **Example:** `Device# ping cisco-rtp`	Diagnoses basic network connectivity. • After the DNS configuration is set, you can verify the DNS server by using a hostname to ping or telnet to a device.
Step 3	**show hosts** **Example:** `Device# show hosts`	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of host names and addresses. • After a name is resolved using DNS, use the **show hosts** command to view the cached hostnames and the DNS configuration.

	Command or Action	Purpose
Step 4	**debug ip domain** **Example:** `Device# debug ip domain`	Enables DNS debugging and displays DNS debugging information. • To view more DNS debugging options such as DNS server response debugging and so on, use the question mark (?) online help function.

Defining a DNS View

Perform this task to define a DNS view. A DNS view definition can be used to respond to either an incoming DNS query or an internally generated DNS query.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*}
4. [**no**] **dns trust** *name*
5. [**no**] **domain lookup**
6. Do one of the following:
   - **domain name** *domain-name*
   - **domain list** *domain-name*
7. Do one of the following:
   - **domain name-server** [**vrf** *vrf-name*] *name-server-ip-address*
   - **domain name-server interface** *interface*
8. **domain multicast** *domain-name*
9. [**no**] **dns forwarding**
10. **dns forwarder** [**vrf** *vrf-name*] *forwarder-ip-address*
11. **dns forwarding source-interface** *interface*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*} **Example:** `Device(config)# ip dns view vrf vpn101 user3`	Defines a DNS view and enters DNS view configuration mode.
Step 4	[**no**] **dns trust** *name* **Example:** `Device(cfg-dns-view)# dns trust name`	(Optional) Enables or disables storage of trusted keys in a view and enters DNS view configuration mode. The dns trust key enables the DNS security feature.
Step 5	[**no**] **domain lookup** **Example:** `Device(cfg-dns-view)# domain lookup`	(Optional) Enables or disables DNS-based hostname-to-address translation for internally generated DNS queries handled using the DNS view. **Note** The domain lookup capability is enabled by default.

Step 6	Do one of the following:
	• **domain name** *domain-name*
	• **domain list** *domain-name*
	Example:
	`Device(cfg-dns-view)# domain name example.com`
	Example:
	`Device(cfg-dns-view)# domain list example1.com`
		(Optional) Defines a default domain name to be used by this DNS view to complete unqualified hostnames when addressing DNS queries.
		or
		(Optional) Defines a list of domain names to be used by this DNS view to complete unqualified hostnames when addressing DNS queries.
		• The device attempts to respond to the query using the parameters specified by the selected DNS view. First, the Cisco IOS software looks in the hostname cache associated with the view. If the query can be answered from that information, the device responds to the query. Otherwise, because the query cannot be answered using the hostname cache, the device forwards the query using the configured domain name servers.
		• If the device is using this view to handle a DNS query for an unqualified hostname and domain lookup is enabled for the view, the Cisco IOS software appends a domain name (either a domain name from the domain name list or the default domain name) in order to perform any of the following activities:
		  • Looking up the hostname in the name server cache.
		  • Forwarded the query to other name servers (whether to the hosts specified as DNS forwarders in the selected view or to the limited broadcast address).

	Command or Action	Purpose
		• You can specify a single, default domain name, an ordered list of domain names, or both. However, the default domain name is used only if the domain list is empty.

Step 7	Do one of the following:
	• **domain name-server** [**vrf** *vrf-name*] *name-server-ip-address*
	• **domain name-server interface** *interface*
	Example:
	`Device(cfg-dns-view)# domain name-server 192.168.2.124`
	Example:
	`Device(cfg-dns-view)# domain name-server interface FastEthernet0/1`
		(Optional) Defines a list of name servers to be used by this DNS view to resolve internally generated DNS queries. The IP address of the name server can be an IPv4 or IPv6 address, and the IP address can be associated with a Virtual Private Network (VPN) routing and forwarding (VRF) instance.
		or
		(Optional) Defines an interface on which to acquire (through DHCP or PPP interaction on the interface) the IP address of a DNS server to add to the list of DNS name servers to be used by this DNS view to resolve internally generated DNS queries.
		• If both of these commands are configured, DHCP or PPP interaction on the interface causes another IP address to be added to the list.

Step 8	**domain multicast** *domain-name* **Example:** `Device(cfg-dns-view)# domain multicast www.example8.com`	(Optional) Specifies the IP address to use for multicast lookups handled using the DNS view.
Step 9	[**no**] **dns forwarding** **Example:** `Device(cfg-dns-view)# dns forwarding`	(Optional) Enables or disables forwarding of incoming DNS queries handled using the DNS view. **Note** The query forwarding capability is enabled by default.
Step 10	**dns forwarder** [**vrf** *vrf-name*] *forwarder-ip-address* **Example:** `Device(cfg-dns-view)# dns forwarder 192.168.3.240`	Defines a list of name servers to be used by this DNS view to forward incoming DNS queries. • The forwarder IP address can be an IPv4 or IPv6 address. • If no forwarding name servers are defined, then the configured list of domain name servers is used instead. • If no name servers are configured either, then queries are forwarded to the limited broadcast address.
Step 11	**dns forwarding source-interface** *interface* **Example:**	Defines the interface on which to forward queries when this DNS view is used.

	Command or Action	Purpose
	`Device(cfg-dns-view)# dns forwarding source-interface FastEthernet0/0`	
Step 12	**end** **Example:** `Device(cfg-dns-view)# end`	Returns to privileged EXEC mode.

Verifying DNS Views

Perform this task to verify the DNS configuration.

SUMMARY STEPS

1. **enable**
2. **show ip dns view** [**vrf** *vrf-name*] [**default** | *view-name*]
3. **show ip dns server** [**vrf** *vrf-name*] [**default** | *view-name*]
4. **clear ip dns servers**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dns view** [**vrf** *vrf-name*] [**default**	*view-name*] **Example:** `Device# show ip dns view vrf vpn101 user3`	Displays information about a particular DNS view, a group of views (with the same view name or associated with the same VRF), or all configured DNS views.
Step 3	**show ip dns server** [**vrf** *vrf-name*] [**default**	*view-name*] **Example:** `Device# show ip dns server vrf vpn101 user3`	Displays information from name server cache.
Step 4	**clear ip dns servers**	Cleans up server from name server cache.

Defining a DNS View List

Perform this task to define an ordered list of DNS views with optional, additional usage restrictions for each view list member. The device uses a DNS view list to select the DNS view that will be used to handle a DNS query.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns view-list** *view-list-name*
4. **ip dns name-list** [*number*] [*permit/deny*] [*name*]
5. **view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
6. **restrict name-group** *name-list-number*
7. **restrict source access-group** *acl-number*
8. **exit**
9. **end**
10. **show ip dns view-list** *view-list-name*
11. **show ip dns name-list** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dns view-list** *view-list-name* **Example:** `Device(config)# ip dns view-list userlist5`	Defines a DNS view list and enters DNS view list configuration mode.
Step 4	**ip dns name-list** [*number*] [*permit/deny*] [*name*] **Example:** `Device(config)# ip dns name-list 10`	Defines a DNS name list and enters DNS name list configuration mode.
Step 5	**view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# view vrf vpn101 user5 10`	Defines a DNS view list member and enters DNS view list member configuration mode.
Step 6	**restrict name-group** *name-list-number* **Example:** `Device(cfg-dns-view-list-member)# restrict name-group 500`	(Optional) Specifies that this DNS view list member cannot be used to respond to a DNS query unless the query hostname matches a permit clause in the specified DNS name list and none of the deny clauses.

	Command or Action	Purpose
		• To define a DNS name list entry, use the **ip dns name-list** command.
Step 7	**restrict source access-group** *acl-number* **Example:** `Device(cfg-dns-view-list-member)# restrict access-group 99`	(Optional) Specifies that this DNS view list member cannot be used to respond to a DNS query unless the source IP address of the DNS query matches the specified standard ACL. • To define a standard ACL entry, use the **access-list** command.
Step 8	**exit** **Example:** `Device(cfg-dns-view-list-member)# exit`	Exits DNS view list member configuration mode. • To add another view list member to the list, go to Step 4.
Step 9	**end** **Example:** `Device(cfg-dns-view-list)# end`	Returns to privileged EXEC mode.
Step 10	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.
Step 11	**show ip dns name-list** *number* **Example:** `Device# show ip dns name-list 5`	Displays information about a particular DNS name list or all configured DNS name lists.

Modifying a DNS View List

To provide for efficient management of the order of the members in a view list, each view list member definition includes the specification of the position of that member within the list. That is, the order of the members within a view list is defined by explicit specification of position values rather than by the order in which the individual members are added to the list. This enables you to perform either of the following tasks without having to remove all the view list members and then redefine the view list membership in the desired order:

Adding a Member to a DNS View List Already in Use

Perform this optional task if you need to add another member to a DNS view list that is already in use.

For example, suppose the DNS view list named userlist5 is already defined and in use as a default view list or as an interface-specific view list. Assume that the list consists of the following members:

- DNS view user1 with position number 10

- DNS view user2 with position number 20
- DNS view user3 with position number 30

If you need to add DNS view user4 as the second member of the list, add that view to the list with a position number value from 11 to 19. You do not need to remove the three existing members and then add all four members to the list in the desired order.

SUMMARY STEPS

1. **enable**
2. **show ip dns view-list** *view-list-name*
3. **configure terminal**
4. **ip dns view-list** *view-list-name*
5. **view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
6. **end**
7. **show ip dns view-list** *view-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip dns view-list** *view-list-name* **Example:** `Device(config)# ip dns view-list userlist5`	Defines a DNS view list and enters DNS view list configuration mode.
Step 5	**view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# view user4 15`	Defines a DNS view list member and enters DNS view list member configuration mode.
Step 6	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(cfg-dns-view-list-member)# end`	
Step 7	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.

Changing the Order of the Members of a DNS View List Already in Use

Perform this optional task if you need to change the order of the members of a DNS view list that is already in use.

For example, suppose the DNS view list named userlist5 is already defined and in use as a default view list or as an interface-specific view list. Assume that the list consists of the following members:

- DNS view user1 with position number 10
- DNS view user2 with position number 20
- DNS view user3 with position number 30

If you want to move DNS view user1 to the end of the list, remove that view from the list and then add it back to the list with a position number value greater than 30. You do not need to remove the three existing members and then add the members back to the list in the desired order.

SUMMARY STEPS

1. **enable**
2. **show ip dns view-list** *view-list-name*
3. **configure terminal**
4. **ip dns view-list** *view-list-name*
5. **no view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
6. **view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
7. **end**
8. **show ip dns view-list** *view-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.

	Command or Action	Purpose
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip dns view-list** *view-list-name* **Example:** `Device(config)# ip dns view-list userlist5`	Defines a DNS view list and enters DNS view list configuration mode.
Step 5	**no view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# no view user1 10`	Removes a DNS view list member from the list.
Step 6	**view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# view user1 40`	Defines a DNS view list member and enters DNS view list member configuration mode.
Step 7	**end** **Example:** `Device(cfg-dns-view-list-member)# end`	Returns to privileged EXEC mode.
Step 8	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.

Specifying the Default DNS View List for the DNS Server of the Device

Perform this task to specify the default DNS view list for the device's DNS server. The device uses the default DNS view list to select a DNS view to use to handle an incoming DNS query that arrives on an interface for which no interface-specific DNS view list has been defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server view-group** *name-list-number*
4. **exit**
5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dns server view-group** *name-list-number* **Example:** `Device(config)# ip dns server view-group 500`	Configures the default DNS view list for the device's DNS server.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Displays information about how DNS view lists are applied. The default DNS view list, if configured, is listed in the default DNS view information as the argument for the **ip dns server view-group** command.

Specifying a DNS View List for a Device Interface

Perform this optional task if you need to specify a DNS view list for a particular device interface. The device uses that view list to select a DNS view to use to handle a DNS query that arrives on that interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface*
4. **ip dns view-group** *view-list-name*
5. **end**
6. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface* **Example:** `Device(config)# interface ATM2/0`	Configures an interface type and enter interface configuration mode so that the specific interface can be configured.
Step 4	**ip dns view-group** *view-list-name* **Example:** `Device(config-if)# ip dns view-group userlist5`	Configures the DNS view list for this interface on the device.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Displays information about how DNS view lists are applied. Any DNS view lists attached to interfaces are listed in the information for each individual interface, as the argument for the **ip dns view-group** command.

Specifying a Source Interface to Forward DNS Queries

Perform this optional task if you need to specify a source interface to forward the DNS queries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*}
4. **domain resolver source-interface** *interface-type number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*} **Example:** `Device(config)# ip dns view vrf vpn32 user3`	Creates the DNS view of the specified name associated with the specified VRF instance and then enters DNS view configuration mode.
Step 4	**domain resolver source-interface** *interface-type number* **Example:** `Device(cfg-dns-view)# domain resolver source-interface fastethernet 0/0`	Sets the source IP address of the DNS queries for the DNS resolver functionality.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuration Examples for DNS

Example: Creating a Domain List with Alternate Domain Names

The following example establishes a domain list with several alternate domain names:

```
ip domain list csi.com
ip domain list telecomprog.edu
ip domain list merit.edu
```

Example: Mapping Host Names to IP Addresses

The following example configures the host-name-to-address mapping process. IP DNS-based translation is specified, the addresses of the name servers are specified, and the default domain name is given.

```
! IP DNS-based host name-to-address translation is enabled
ip domain lookup
! Specifies hosts 192.168.1.111 and 192.168.1.2 as name servers
ip name-server 192.168.1.111 192.168.1.2
! Defines cisco.com as the default domain name the device uses to complete
! Set the name for unqualified host names
ip domain name cisco.com
```

Example: Customizing DNS

The following example shows the ip dns servers.

```
show ip dns server

   IP             VRF      TTL(s)    RTT(ms)  RTO(ms)  EDNS  DNSSEC  RECURSION
-------------------------------------------------------------------------------
2::1              red      628       1451     1451      Yes     Yes     Yes
172.168.10.1               875       1787     1787      Yes     Yes     Yes
2.2.2.1           red      606       1447     1447      Yes     Yes     Yes
1::1                       207       300      300       Yes     Yes     Yes
1.1.1.1                    179       242      242       Yes     Yes     Yes
```

Example: Split DNS View Lists Configured with Different View-use Restrictions

The following example shows how to define two DNS view lists, userlist1 and userlist2. Both view lists comprise the same three DNS views:

- DNS view user1 that is associated with the usergroup10 VRF
- DNS view user2 that is associated with the usergroup20 VRF
- DNS view user3 that is associated with the usergroup30 VRF

Both view lists contain the same DNS views, specified in the same order:

```
ip dns view-list userlist15
 view vrf usergroup100 user1 10
  restrict name-group 121
  exit
view vrf usergroup200 user2 20
  restrict name-group 122
  exit
view vrf usergroup300 user3 30
  restrict name-group 123
  exit
!
exit
ip dns view-list userlist16
 view vrf usergroup100 user1 10
  restrict name-group 121
  restrict source access-group 71
  exit
 view vrf usergroup200 user2 20
  restrict name-group 122
  restrict source access-group 72
  exit
 view vrf usergroup300 user3 30
  restrict name-group 123
  restrict source access-group 73
  exit
exit
```

The two DNS view lists differ, though, in the usage restrictions placed on their respective view list members. DNS view list userlist15 places only query hostname restrictions on its members while view list userlist16 restricts each of its members on the basis of the query hostname and the query source IP address:

- Because the members of userlist15 are restricted only based on the VRF from which the query originates, userlist15 is typical of a view list that can be used to select a DNS view for handling DNS requests from internal clients.
- Because the members of userlist16 are restricted not only by the query VRF and query hostname but also by the query source IP address, userlist16 is typical of a view list that can be used to select a DNS view for handling DNS requests from external clients.

Additional References for Configuring DNS

Related Documents

Related Topic	Document Title
Master Command List	Cisco IOS Master Command List
IP Addressing Services Command Reference	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1348	DNS NSAP Resource Records

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

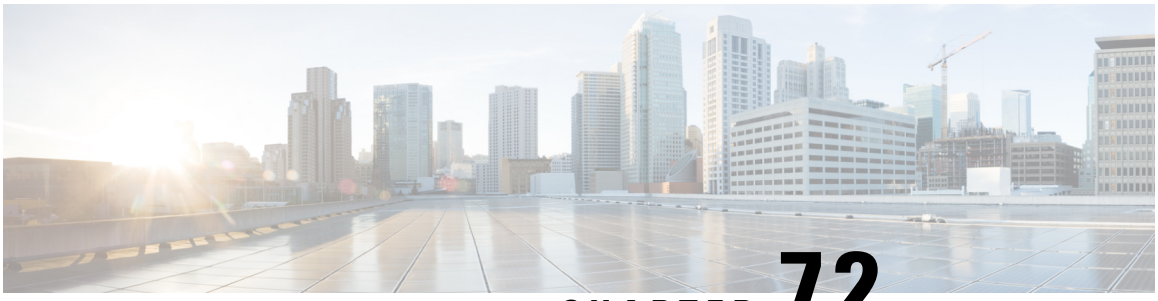
Feature Information for Configuring DNS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 113: Feature Information for Configuring DNS

Feature Name	Releases	Feature Configuration Information
Configuring DNS	Cisco IOS XE Release 2.1	The Domain Name System (DNS) is a distributed database in which you can map host names to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated host name. The Cisco IOS XE software maintains a cache of host name-to-address mappings. This cache speeds the process of converting names to addresses.
	Cisco IOS XE Release 3.13S	The following commands were introduced or modified: **debug ip domain**, **debug ip domain replies**.
	Cisco IOS XE Release 3.16S	The following commands were introduced or modified: **dns trust**, **clear ip dns servers**.



CHAPTER 72

Dynamic DNS Support for Cisco IOS Software

The Dynamic DNS Support for Cisco IOS Software feature enables Cisco IOS software devices to perform Dynamic Domain Name System (DDNS) updates to ensure that an IP host DNS name is correctly associated with its IP address.

It provides two mechanisms to generate or perform DDNS: the IETF standard as defined by RFC 2136 and a generic HTTP using various DNS services. With this feature, you can define a list of hostnames and IP addresses that will receive updates, specify an update method, and specify a configuration for Dynamic Host Configuration Protocol (DHCP) triggered updates.

Restrictions for Dynamic DNS Support for Cisco IOS Software

The performance of the DHCP client can be impacted when the Dynamic DNS Support for Cisco IOS Software feature is enabled, because of sending DDNS update packets and waiting for responses from the server (before sending the ACK to the client REQUEST) and the client (immediately after receiving the ACK and assigning the address to the interface). The default for the client is two attempts with a 5-second wait time between attempts.

The DHCP server continues to process DHCP client DISCOVER and REQUEST packets while waiting for the DDNS updates to complete. Even if the update is done before sending the ACK to the client, it does not delay processing of other DHCP requests. The DHCP server could be impacted minimally because of the time and memory needed in order to set up the DDNS update and get things started.

Reloading the system may take a little longer in some cases, such as, if there are outstanding DDNS updates that need to complete.

Information About Dynamic DNS Support for Cisco IOS Software

Domain Name System and Dynamic Updates

The DNS was designed to support queries of a statically configured database. The data was expected to change, but minimally. All updates were made as external edits to a zone master file. The domain name identifies a node within the domain name space tree structure. Each node has a set (possibly empty) of Resource Records (RRs). All RRs having the same NAME, CLASS, and TYPE are called a Resource Record Set (RRset).

There are address (A) or forward RRs and pointer (PTR) or reverse RRs. The DDNS update can specify additions or deletions of hostnames and IP addresses. The two mechanisms to update this information are by using HTTP-based protocols such as DynDNS.org or by using the IETF standard.

DDNS Updates for HTTP-Based Protocols

The Dynamic DNS Support for Cisco IOS Software feature provides the capability of a proprietary HTTP-based protocol to generate or perform DDNS updates. The most notable HTTP-based protocol is DynDNS.org, but there are many others.

Since most of these protocols consist of a simple HTTP command that specifies parameters such as hostname and IP address in the URL portion of the command, this feature takes the same generic approach. You can specify the hostname and IP address in a URL. Configuration of a maximum interval between updates is also allowed.

DHCP Support for DDNS Updates

Before the Dynamic DNS Support for Cisco IOS Software feature, a DHCP server assigned IP addresses to DHCP clients and any DNS information was static. In a network that uses a DHCP server, there are many cases in which DNS hostnames should be associated with the IP addresses that are being assigned. There is an existing method for dynamically updating DNS for DHCP by using information in the fully qualified domain name (FQDN) DHCP option (if it is supplied by the client).

The Dynamic DNS Support for Cisco IOS Software feature enables the DHCP server to support a new FQDN DHCP option. In addition, when the address on an interface is configured, the client can pass the new FQDN option to the server so that name-to-address and address-to-name translations can be updated for the DHCP client as well.

Feature Design of Dynamic DNS Support for Cisco IOS Software

The Dynamic DNS Support for Cisco IOS Software feature enables the tracking of the FQDN DHCP option. If dynamic updates are enabled for the DHCP server, the server updates the PTR RR. The PTR RRs are used for reverse mapping (translation of addresses to names). PTRs use official names not aliases. The name in a PTR record is the local IP address portion of the reverse name.

If the client requests the server to update A RRs as well, the server will attempt to do it. The A RR provides the name-to-address mapping for a DNS zone. The server may be configured to override the client suggestion and always update PTR and A RRs.

The DHCP client can specify whether or not it wants to allow dynamic updates (include the FQDN option), instruct the server to allow the client to update both A and PTR RRs (normally only the A RR is updated by the client), and optionally instruct the server not to update any DNS information (either because the client will be updating both or simply because the client does not want the server to do any updates at all).

There are three basic components of the Dynamic DNS Support for Cisco IOS Software feature that are as follows:

- Definition of the hostname list and IP addresses that will receive updates using a new command that specifies a group of hostnames. Each configured list can consist of any number of IPv4 addresses or hostnames. If a hostname is configured, the name is translated to an IPv4 address at the time at which it is used.
- Specification of an update method. The two options for the update method are HTTP and DDNS. If the HTTP option is specified, the configuration will include a URL. The username and password must be explicitly written into the URL string and the entire "GET" operation must be specified on one line. The specification will be stored in a linked list. If the update method is DDNS, the configuration will include the update of the IP address.

Events that trigger updates can be as follows:

- IP address that is assigned by a DHCP server for an IP device
- IP address assigned to a router using a DHCP client
- Forwarding of the fully qualified domain name (FQDN) of a user or router hostname from the DHCP client to the server
- Point-to-Point Protocol (PPP)/IP Control Protocol (IPCP) obtaining an IP address for a router interface
- Forced update using a timer to verify a router IP address

Associated with each update method is a value specifying the maximum number of seconds between updates. If left unspecified, then the update is performed only when the address is changed. If specified, the update is performed automatically if the specified number of seconds have passed since the last update.

How to Configure Dynamic DNS Support for Cisco IOS Software

Configuring a Host List

Perform this task to configure a host list if you are going to use a host list in your configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host-list** *host-list-name*
4. **host** [**vrf** *vrf-name*] {*host-ip-address* | *hostname*}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip host-list** *host-list-name* **Example:** `Router(config)# ip host-list abc`	Specifies a list of hosts and enters host-list configuration mode. The *host-list-name argument*assigns a name to the list of hosts.
Step 4	**host** [**vrf** *vrf-name*] {*host-ip-address* | *hostname*} **Example:** `Router(host-list)# host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com host 10.6.6.6 f.com host vrf abc a.com b.com c.com host vrf def 10.1.1.1 10.2.2.2 10.3.3.3`	Configures one or more hosts. The arguments and keyword are as follows: • **vrf** *vrf-name* --Associates a hostname with a virtual private network (VPN) routing and forwarding instance (VRF) name. **Note** All hostnames or IP addresses specified after the **vrf** keyword are associated with that VRF. • *host-ip-address* --Specifies an IP address for a host in the host list. You can specify more than one host using this argument by listing the hostname and IP addresses on the same line. • *hostname* --Specifies a hostname.
Step 5	**exit** **Example:** `Router(host-list)# exit`	Exits to global configuration mode.

Examples

The following example shows how to configure several hosts with VRF:

```
ip host-list abc
 host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com
 host 10.6.6.6 f.com
 host vrf abc a.com b.com c.com
 host vrf def 10.1.1.1 10.2.2.2 10.3.3.3
```

Verifying the Host-List Configuration

To verify the host-list configuration, perform the following steps.

SUMMARY STEPS

1. **show ip host-list**
2. **show running-config | inc host-list**
3. **show running-config | inc host**
4. **debug ip ddns update**

DETAILED STEPS

Step 1 show ip host-list

Use this command to verify that the IP addresses and hostnames have been assigned to a host list, for example:

Example:

```
Router# show ip host-list abc
Host list: abc
 ddns.abc
 10.2.3.4
 ddns2.abc
 10.3.4.5
 ddns3.com
 10.3.3.3
 d.org
 e.org
 1.org.2.org
 3.com
 10.2.2.2 (VRF: test)
 10.5.5.5 (VRF: test)
 a.net (VRF: test)
 b.net (VRF: test)
```

Step 2 show running-config | inc host-list

Use this command to verify the configuration of a host list, for example:

Example:

```
Router# show running-config | inc host-list
ip host-list a
ip host-list b
ip host-list c
ip host-list abc
```

Step 3 show running-config | inc host

Use this command to verify the configuration of a hostname, for example:

Example:

```
Router# show running-config | inc host
hostname who
ip host who 10.0.0.2
ip host-list a
```

```
 host 10.1.1.1 a.com b.com 10.2.2.3 10.2.2.2 c.com. 10.3.3.3 10.4.4.4
 host d.com
 host vrf abc 10.10.10.4 10.10.10.8
 host vrf def 10.2.3.4 10.6.7.8
ip host-list b
 host a.com b.com c.com 10.1.1.1 10.2.2.2 10.3.3.3
 host vrf ppp 10.2.1.0
ip host-list c
 host 10.1.1.1 10.2.2.2 10.3.3.3 a.com b.com 10.4.4.4 10.5.5.5 d.com
 host 10.6.6.6 f.com
 host vrf zero a.com b.com c.om
 host vrf one 10.1.1.1 10.2.2.2 10.3.3.3
ip host-list unit-test
 host ddns.unit.test 10.2.3.4 ddns2.unit.test 10.3.4.5 ddns3.com 10.3.3.3 d.org e.org
 host 1.org.2.org 3.com
 host vrf ZERO 10.2.2.2 10.5.5.5 a.net b.net
 ip ddns update hostname use-this.host.name
 ip ddns update this-method host 10.2.3.4
 ip ddns update this-method host this-host
 ip ddns update this-method host-group this-list
 ip ddns update this-method host 10.3.4.5
 ip ddns update test host 10.19.192.32
 ip ddns update test host 10.19.192.32
 ip ddns update a host-group a
 ip ddns update a host-group ab
 ip ddns update aa host-group ab
 ip ddns update method host 10.33.44.55
```

Step 4 debug ip ddns update

Use the **debug ip ddns update** command for the following configuration to verify the configuration of the hosts. Two servers are configured in the host list. A DHCP client is configured for IETF DDNS updating of both A and DNS RRs and requesting the DHCP server to update neither. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server not to update either A or PTR Resource Records. This is configured using the interface version of the command. The DHCP server is configured to allow the DHCP client to update whatever RRs it chooses.

Example:

```
!Configure the DHCP Client
ip host-list servers
 host 10.19.192.32 10.0.0.1
ip ddns update method testing
 ddns
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing host-group servers
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging
debug ip ddns update
!The update to the server 10.0.0.1 fails in this example
00:18:58:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.8, mask 255.0.0.0,
 hostname canada_reserved
00:18:58: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server 10.19.192.32
00:18:58: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration to settle
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server 10.19.192.32
00:19:01: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server 10.0.0.1
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server 10.0.0.1
00:19:01: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.8 server 10.0.0.1
```

```
00:19:01: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.8 server 10.0.0.1
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.19.192.32
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.0.0.1
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:19:01: DDNS: Using server 10.0.0.1
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 6 (YXDOMAIN)
00:19:01: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = 10.in-addr.arpa
00:19:01: DDNS: Update: delete 10.0.0.11.in-addr.arpa. all PTR RRs
00:19:01: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:19:01: DDNS: Dynamic DNS Update 2 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:01: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:01: DDNS: Using server 10.19.192.32
00:19:01: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:19:01: DDNS: Zone = hacks
00:19:01: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:01: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
00:19:01: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:01: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 finished
00:19:01: DYNDNSUPD: Another update completed (total outstanding=2)
00:19:11: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:11: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:11: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:11: DDNS: Using server 10.0.0.1
00:19:11: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:11: DDNS: Zone = hacks
00:19:11: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:11: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
00:19:11: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:19:11: DDNS: Using server 10.0.0.1
00:19:11: DDNS: Dynamic Update 1: (sending to server 10.0.0.1)
00:19:11: DDNS: Zone = hacks
00:19:11: DDNS: Prerequisite: canada_reserved.hacks not in use
00:19:11: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.8
00:19:21: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:21: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 failed
00:19:21: DYNDNSUPD: Another update completed (total outstanding=1)
00:19:21: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:19:21: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.8 failed
00:19:21: DYNDNSUPD: Another update completed (total outstanding=0)
```

Configuring DHCP Support of DDNS Updates

DDNS updates contain information about A or forward RRs for a particular IP address. The IP address is in dotted decimal form, and there must be at least one A record for each host address. The name specified is the

hostname expressed as an FQDN (ns.example.com). The PTR or reverse RRs map a domain name to another domain name and is used for reverse mapping (IP address to domain name).

The updates are performed using messages. In general, you will probably want DDNS updates done by the server after the server has sent the ACK response to the DHCP client. Performing the DDNS updates before sending the ACK response will delay the response to the client. Both methods are supported. The default is to do the updates after sending the response.

When looking for a client hostname to use in the update, the server will take the hostname from the FQDN option, if such exists, first. If there is no FQDN option, the server will look for a HOSTNAME option and take the name from there.

If the FQDN or HOSTNAME option is included in subsequent RENEWAL messages, the server will attempt to perform the DDNS update each time the lease is renewed. This process gives the opportunity for the client to change the name specified after the lease has been granted and have the server do the appropriate updates. Although the server has this capability, the DHCP client will continue to use the same hostname throughout the duration of a lease.

The IP address of the server to update is discovered by sending a DNS query for records associated with the hostname to update. If such a record exists, the hostname of the master DNS server is extracted from this information. If no such record exists, the record, which should be included in the response, is used as the authoritative record for the zone where the hostname exists. In either case, once the master DNS server hostname is found, another query for A RRs is sent in order to discover the IP address of this server. The resulting IP address is used for sending updates.

Perform this task to configure the DDNS updates.

Before you begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

Note DHCP server-pool configuration commands and interface configurations have precedence over global configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp update dns** [**both**] [**override**] [**before**]
4. **ip dhcp-client update dns** [**server** {**both** | **none**}]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp update dns** [**both**] [**override**] [**before**] **Example:** `Router(config)# ip dhcp update dns both override`	Enables DDNS updates of PTR RRs for all address pools except those configured with the per-pool **update dns** command, which overrides global configuration. The keywords are as follows: • **both** --(Optional) Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client has specified in the FQDN option that the server should not perform the updates. • **override** --(Optional) Enables the DHCP server to perform DDNS updates for PTR RRs even if the DHCP client has specified in the FQDN option that the server should not perform the updates. **Note** If you specify the **both** and **override** keywords together, this enables the DHCP server to perform DDNS updates for A and PTR RRs overriding anything the DHCP client specified in the FQDN option to the contrary. • **before** --(Optional) Enables the DHCP server to perform DDNS updates before sending the DHCP ACK back to the client. The default is to perform updates after sending the DHCP ACK.
Step 4	**ip dhcp-client update dns** [**server** {**both** | **none**}] **Example:** `Router(config)# ip dhcp-client update dns server both`	Enables DDNS updates of PTR RRs. The optional **server** keyword enables the server to perform DDNS updates for A and PTR RRs. The keywords are as follows: • **both** --Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client specifies in the FQDN option that the server should not perform the updates. • **none** --Enables the DHCP client to perform DDNS updates and the server will not perform any updates. The server can override this action.

	Command or Action	Purpose
		Note The **ip dhcp-client update dns server none** command instructs the server not to perform any updates. If configured to do so, the server can override the client. **Note** The **ip dhcp-client update dns server both** command instructs the server to update both the A and PTR RRs.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Examples

The following example shows how to configure A and PTR RR updates that are performed by the server only:

```
ip dhcp-client update dns server both

ip dhcp update dns both override
```

Configuring DDNS Update Support on Interfaces

Perform this task to configure your interfaces for DDNS update capability.

Note The interface configuration overrides the global configuration.

Before you begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

Note The changes will not take effect until any current lease on the interface is released and a new lease is requested that uses a new DHCP DISCOVER packet. This means configuring the **ip address dhcp** command or using the **release dhcp** EXEC command followed by the **renew dhcp** EXEC command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type number*
4. **ip dhcp client update dns** [**server** {**both** | **none**}]
5. **ip address dhcp**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type number* **Example:** `Router(config)# interface ethernet1`	Specifies an interface type and number and enters interface configuration mode.
Step 4	**ip dhcp client update dns** [**server** {**both** | **none**}] **Example:** `Router(config-if)# ip dhcp client update dns server both`	Configures the DHCP client to include an FQDN option when sending packets to the DHCP server. The keywords are as follows: • **both** --(Optional) Enables the DHCP server to perform DDNS updates for A and PTR RRs, unless the DHCP client specifies in the FQDN option that the server should not perform the updates. • **none** --(Optional) Enables the DHCP client to perform DDNS updates and the server will not perform any updates. The server can override this action. **Note** The **ip dhcp client update dns server none** command instructs the server not to perform any updates. If configured to do so, the server can override the client. **Note** The **ip dhcp client update dns server both** command instructs the server to update both the A and PTR RRs.

	Command or Action	Purpose
Step 5	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Releases any current lease on the interface and enables the configuration. **Note** You can also release any lease by using the **release dhcp** EXEC command followed by the **renew dhcp** EXEC command.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.

Configuring a Pool of DHCP Servers to Support DDNS Updates

There are two parts to the DDNS update configuration on the client side. First, if the **ip ddns update method** command is configured on the client, which specifies the DDNS-style updates, then the client will be trying to generate or perform A updates. If the **ip ddns update method ddns both** command is configured, then the client will be trying to update both A and PTR RRs.

Second, the only way for the client to communicate with the server, with reference to what updates it is generating or expecting the server to generate, is to include an FQDN option when communicating with the server. Whether or not this option is included is controlled on the client side by the **ip dhcp-client update dns** command in global configuration mode or the **ip dhcp client update dns** command in interface configuration mode.

If the FQDN option is included in the DHCP interaction, then the client may instruct the server to update "reverse" (the default), "both", or "none." Obviously, if the **ip ddns update method** command is configured with the **ddns** and **both**keywords, then the FQDN option configuration should reflect an IP DHCP client update DNS server none, but you have to configure the system correctly.

Finally, even if the client instructs the server to update both or update none, the server can override the client request and do whatever it was configured to do anyway. If there is an FQDN option in the DHCP interaction as above, then server can communicate to the client that it was overridden, in which case the client will not perform the updates because it knows that the server has done the updates. Even if the server is configured to perform the updates after sending the ACK (the default), it can still use the FQDN option to instruct the client what updates it will be performing and thus the client will not do the same types of updates.

If the server is configured with the **update dns** command with or without any keywords, and if the server does not see an FQDN option in the DHCP interaction, then it will assume that the client does not understand DDNS and will automatically act as though it were configured to update both A and PTR RRs on behalf of the client.

Perform this task to configure a pool of DHCP servers to support DDNS updates.

Before you begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **update dns** [**both** | **never**] [**override**] [**before**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Router(config)# ip dhcp pool test`	Assigns a name to a DHCP pool and enters DHCP configuration mode.
Step 4	**update dns** [**both** | **never**] [**override**] [**before**] **Example:** `Router(dhcp-config)# update dns never`	Enables DDNS update capability for a pool of DHCP servers for any addresses assigned from this address pool. If the server is configured using this command with or without any of the other keywords, and if the server does not see an FQDN option in the DHCP interaction, then it will assume that the client does not understand DDNS and act as though it were configured to update both A and PTR records on behalf of the client. The keywords are as follows: • **both** --(Optional) Perform forward and reverse updates. If the **before** optional keyword is specified along with the **both** keyword, the server can perform DDNS updates before sending the ACK back to the client. If the **override** optional keyword is specified with the **both** keyword, the server can override the client and update forward and reverse RRs. If the **override** and **before** optional keywords are specified with the **both** keyword, the server can override the client (forward and reverse updates) and perform the updates before sending the ACK.

	Command or Action	Purpose
		• **never** --(Optional) Never perform updates for this pool. • **override** --(Optional) Override the client FQDN flags. If the **before** optional keyword is specified, the updates will be performed before sending the ACK. • **before** --(Optional) Perform updates before sending the ACK.
Step 5	**exit** **Example:** `Router(dhcp-config)# exit`	Exits to global configuration mode.

Examples

The following example shows how to configure a pool of DHCP servers to perform updates for A and PTR RRs before the ACK is sent:

```
ip dhcp pool test
 update dns both before
```

Configuring the Update Method and Interval

Perform this task to specify the update method and interval maximum.

Before you begin

In order for DDNS updates to discover the DNS server, in cases in which the user did not configure the server, the **ip name-server** command should be configured. This name server should be reachable by the system, and the **ip domain lookup** command should be configured (which is the default anyway). In cases in which the configured hostname does not include a period (is not a fully qualified domain name [FQDN]), an IP domain name should be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ddns update method** *method-name*
4. **interval minimum** *days hours minutes seconds*
5. **interval maximum** *days hours minutes seconds*
6. **ddns** [**both**]
7. **http**
8. **add** *url*
9. **remove** *url*
10. exit

11. **exit**
12. **interface** *interface-type number*
13. **ip ddns update hosthame** *hostname*
14. ip ddns update *name*
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ddns update method** *method-name* **Example:** `Router(config)# ip ddns update method myupdate`	Specifies the update method name and enters DDNS update method configuration mode.
Step 4	**interval minimum** *days hours minutes seconds* **Example:** `Router(DDNS-update-method)# interval minimum 1 0 0 0`	Configures a minimum update interval. The arguments are as follows: • *days* --Range is from 0 to 365. • *hours* --Range is from 0 to 23. • *minutes* --Range is from 0 to 59. • *seconds* --Range is from 0 to 59.
Step 5	**interval maximum** *days hours minutes seconds* **Example:** `Router(DDNS-update-method)# interval maximum 1 0 0 0`	Configures a maximum update interval. The arguments are as follows: • *days* --Range is from 0 to 365. • *hours* --Range is from 0 to 24. • *minutes* --Range is from 0 to 60. • *seconds* --Range is from 0 to 60.
Step 6	**ddns** [**both**] **Example:** `Router(DDNS-update-method)# ddns`	Configures DDNS as the update method. The **both** keyword specifies that both A and PTR RRs will be updated.

	Command or Action	Purpose
		Note You can specify DDNS or HTTP but not both in one step. If you have specified DDNS, you must disable it by using the **no ddns** command before you can configure HTTP. For the HTTP configuration, see Steps 7,8, and 9.
Step 7	**http** **Example:** `Router(DDNS-update-method)# http`	Configures HTTP as the update method and enters DDNS-HTTP configuration mode.
Step 8	**add** *url* **Example:** `Router(DDNS-HTTP)# add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>`	Configures a URL that should be invoked in order to add or change a mapping between a hostname and an IP address. The following example configures the URL to be invoked to add or change the mapping information using DynDNS.org: • http://userid:password@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>. You have to enter the URL string above. Userid is your userid and password is your password at the DynDNS.org website. The special character strings **< h >** and **< a >** will be substituted with the hostname to update and the IP address with which that hostname should be associated, respectively. **Note** Before entering the question mark (?) character, press the control (Ctrl) key and the v key together on your keyboard. This will allow you to enter the ? without the software interpreting the ? as a help query.
Step 9	**remove** *url* **Example:** `Router(DDNS-HTTP)# remove http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>`	Configures a URL that should be invoked in order to remove a mapping between a hostname and an IP address. The URL takes the same form as the **add** keyword in Step 8.
Step 10	exit **Example:** `Router(DDNS-HTTP)# exit`	Exits to update-method configuration mode.
Step 11	**exit** **Example:** `Router(DDNS-update-method)# exit`	Exits to global configuration mode.

	Command or Action	Purpose
Step 12	**interface** *interface-type number* **Example:** `Router(config)# interface ether1`	Enters interface configuration mode.
Step 13	**ip ddns update hosthame** *hostname* **Example:** `Router(config-if)# ip ddns update hostname abc.dyndns.org`	Specifies a host to be used for the updates. The update will associate this hostname with the configured IP address of the interface. The *hostname* argument specifies the hostname that will receive the updates (for example, DynDNS.org).
Step 14	ip ddns update *name* **Example:** `Router(config-if) ip ddns update myupdate`	Specifies the name of the update method to use for sending Dynamic DNS updates associated with address changes on this interface.
Step 15	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Examples

The following example shows how to configure the update method, the maximum interval of the updates (globally), and configure the hostname on the interface:

```
ip ddns update method mytest
ddns
 http
!Before entering the question mark (?) character in the add http CLI, press the control
(Ctrl) key and the v key together on your keyboard. This will allow you to enter the ?
without the software interpreting the ? as a help query.

 add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>

 interval maximum 1 0 0 0
 exit
interface ether1

 ip ddns update hostname abc.dyndns.org

 ip ddns update mytest
```

Verifying DDNS Updates

Use the **debug ip ddns update** command to verify that DDNS updates are being performed. There are several sample configurations and the debug output that would display for that scenario.

Sample Configuration #1

The following scenario has a client configured for IETF DDNS updating of A DNS RRs during which a DHCP server is expected to update the PTR DNS RR. The DHCP client discovers the DNS server to update using an SOA RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an FQDN DHCP option and notifies the DHCP server that it will be updating the A RRs.

```
!Configure the DHCP Client
ip ddns update method testing
 ddns
interface Ethernet1
 ip dhcp client update dns
 ip ddns update testing
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging
Router# debug ip ddns update
00:14:39:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.4, mask
255.0.0.0, hostname canada_reserved
00:14:39: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.4
00:14:39: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:14:42: DHCPC: Server performed PTR update
00:14:42: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.4
00:14:42: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:14:42: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:14:42: DDNS:   Zone = hacks
00:14:42: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:14:42: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.4
00:14:42: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:14:42: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.4 finished
00:14:42: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #2

The following scenario has the client configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client discovers the DNS server to update using an SOA RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command.

```
!Configure the DHCP Client
ip dhcp-client update dns server none
ip ddns update method testing
 ddns both
interface Ethernet1
 ip ddns update testing
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging
Router# debug ip ddns update
00:15:33:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.5, mask
```

```
255.0.0.0, hostname canada_reserved
00:15:33: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.5
00:15:33: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:15:36: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.5
00:15:36: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:15:36: DDNS:   Zone = 10.in-addr.arpa
00:15:36: DDNS:   Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:15:36: DDNS:   Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:15:36: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:15:36: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:15:36: DDNS:   Zone = hacks
00:15:36: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:15:36: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.5
00:15:36: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:15:36: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.5 finished
00:15:36: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #3

The following scenario the client is configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client explicitly specifies the server to update. The DHCP client is configured to include an FQDN DHCP option which instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command. The DHCP server is configured to override the client request and update both A and PTR RR anyway.

```
!Configure the DHCP Client
ip dhcp client update dns server non
ip ddns update method testing
 ddns both
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns both override
!Enable Debugging on the DHCP Client
Router# debug ip ddns update
00:16:30:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.6, mask
255.0.0.0, hostname canada_reserved
00:16:30: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:16:30: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:16:33: DHCPC: Server performed both updates
```

Sample Configuration #4

In the following scenario the client is configured for IETF DDNS updating of both A and DNS RRs and requesting the DHCP server to update neither. The DHCP client explicitly specifies the server to update. The DHCP client is configured to include an FQDN DHCP option which instructs the DHCP server not to update either A or PTR RRs. This is configured using the global version of the command. The DHCP server is configured to allow the client to update whatever RR it chooses.

```
!Configure the DHCP Client
ip dhcp client update dns server non
```

```
ip ddns update method testing
 ddns both
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing host 172.19.192.32
 ip address dhcp
end
!Configure the DHCP Server
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Enable Debugging on the DHCP Client
Router# debug ip ddns update
00:17:52:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.7, mask
255.0.0.0, hostname canada_reserved
00:17:52: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:17:52: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7
00:17:55: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '11.in-addr.arpa'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 6
(YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = 10.in-addr.arpa
00:17:55: DDNS: Update: delete 10.0.0.11.in-addr.arpa. all PTR RRs
00:17:55: DDNS: Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 2 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=1)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 6 (YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS: Zone = hacks
00:17:55: DDNS: Update: delete canada_reserved.hacks all A RRs
00:17:55: DDNS: Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 2 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=0)
```

Sample Configuration #5

In the following scenario, the debug output shows the HTTP-style DDNS updates. The sample configuration defines a new IP DDNS update method named dyndns that configures a URL to use when adding or changing an address. No URL has been defined for use when removing an address since DynDNS.org does not use such a URL for free accounts. A maximum update interval of 28 days has been configured, so specifying that updates should be sent at least every 28 days. Configuring the new dyndns update method should be used for Ethernet interface .

Note Before entering the question mark (?) character in the "add http" configuration after the **update** keyword, press the control (Ctrl) key and the "v" key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

```
!Configure the DHCP Client
ip ddns update method dyndns
 http
    add http://test:test@<s>/nic/update?system=dyndns&hostname=<h>&myip=<a>
    interval max 28 0 0 0
interface ethernet1
 ip ddns update hostname test.dyndns.org
 ip ddns update dyndns host members.dyndns.org
 ip addr dhcp
!Enable Debugging
Router# debug ip ddns update
00:04:35:%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.32.254.187,
mask 255.255.255.240, hostname test.dyndns.org
00:04:35: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
10.208.196.94
00:04:35: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:04:38: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: init
00:04:38: HTTPDNSUPD: Session ID = 0x7
00:04:38: HTTPDNSUPD: URL =
'http://test:test@10.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.32.254.187'
00:04:38: HTTPDNSUPD: Sending request
00:04:40: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:04:40: HTTPDNSUPD: DATA START
good 10.32.254.187
00:04:40: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:04:40: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
00:04:40: HTTPDNSUPD: Freeing response
00:04:40: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:04:40: HTTPDNSUPD: Clearing all session 7 info
!28 days later, the automatic update happens.
00:05:39: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
10.208.196.94
00:05:39: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: init
00:05:39: HTTPDNSUPD: Session ID = 0x8
00:05:39: HTTPDNSUPD: URL =
'http://test:test@10.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.32.254.187'
00:05:39: HTTPDNSUPD: Sending request
00:05:39: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: DATA START
nochg 10.32.254.187
```

```
00:05:39: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:05:39: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: Freeing response
00:05:39: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:05:39: HTTPDNSUPD: Clearing all session 8 info
```

Configuration Examples for Dynamic DNS Support for Cisco IOS Software

Configuration of the DHCP Client Example

The following example shows that no DDNS updates will be performed for addresses assigned from the address pool “abc.” Addresses allocated from the address pool “def” will have both forward (A) and reverse (PTR) updates performed. This configuration has precedence over the global server configurations.

```
ip dhcp update dns both override
 ip dhcp pool abc
  network 10.1.0.0 255.255.0.0
!
 update dns never
!
 ip dhcp pool def
  network 10.10.0.0 255.255.0.0
```

Configuration of the DHCP Server Example

The following example shows how to configure A and PTR RR updates that are performed by the server only:

```
ip dhcp-client update dns server both
```

```
ip dhcp update dns both override
```

Configuration of the HTTP Updates Example

The following example shows how to configure a PPPoE server for HTTP DDNS:

```
!Username and Password for PPP Authentication Configuration
!
username user1 password 0 cisco
!
!DHCP Pool Configuration
ip dhcp pool mypool
 network 10.10.10.0 255.255.255.0
 default-router 10.10.10.1
!
!VPDN configuration for PPPoE
vpdn enable
!
vpdn-group pppoe
accept-dialin
protocol pppoe
```

```
virtual-template 1
!
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
!Port used to connect to the Internet, it can be the same port that is under test, but to
make the test clear and simple these two are separated.
!
interface FastEthernet0/0
 ip address 10.0.58.71 255.255.255.0
!
!Port under test.
!
interface FastEthernet0/1
 no ip address
 pppoe enable
!
!Virtual template and address pool config for PPPoE.
interface Virtual-Template1
 ip unnumbered Loopback0
 ip mtu 1492
 peer default ip address dhcp-pool mypool
 ppp authentication chap
```

The following example shows how to configure a DHCP client for IETF DDNS:

```
!Default hostname of the router.
hostname mytest
!
!Default domain name on the router.
ip domain name test.com
!
!Port under test.
!
interface FastEthernet0/1
 no ip address (configured to "ip address dhcp")
```

The following example shows how to configure the method of update and the maximum interval of the updates (globally) and configure the hostname on the interface:

Note Before entering the question mark (?) character in the "add http" configuration after the **update** keyword, press the control (Ctrl) key and the "v" key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

```
ip ddns update method mytest
ddns
 http

 add http://test:test@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>

 interval maximum 1 0 0 0
 exit
interface ether1

 ip ddns update hostname abc.dyndns.org

 ip ddns update mytest
```

The following are examples of URLs that can be used to update some HTTP DNS update services. These URLs are correct to the best of the knowledge of Cisco but have not been tested in all cases. Where the word "USERNAME:" appears in the URL, the customer account username at the HTTP site should be used.

Where the word "PASSWORD" appears in the URL, the customer password for that account should be used:

Note Before entering the question mark (?) character in the "add http" configuration after the **update** keyword, press the control (Ctrl) key and the "v" key together on your keyboard. This will allow you to enter the ? without the software interpreting it as a help query.

DDNS

```
http://USERNAME:PASSWORD@members.dyndns.org/nic/update?system=dyndns&hostname=<h>&myip=<a>
!Requires "interval max 28 0 0 0" in the update method definition.
```

TZO

```
http://cgi.tzo.com/webclient/signedon.html?TZOName=<h>&Email=USERNAME&TZOKey=PASSWORD&IPAddress=<a>
```

EASYDNS

```
http://USERNAME:PASSWORD@members.easydns.com/dyn/ez-ipupdate.php?action=edit&myip=<a>&host_id=<h>
```

JUSTLINUX

```
http://USERNAME:PASSWORD@www.justlinux.com/bin/controlpanel/dyndns/jlc.pl?direct=1&username=USERNAME&password=PASSWORD&host=<h>&ip=<a>
```

DYNS

```
http://USERNAME:PASSWORD@www.dyns.cx/postscript.php?username=USERNAME&password=PASSWORD&host=<h>&ip=<a>
```

HN

```
http://USERNAME:PASSWORD@dup.hn.org/vanity/update?ver=1&IP=<a>
```

ZONEEDIT

```
http://USERNAME:PASSWORD@www.zoneedit.com/auth/dynamic.html?host=<h>&dnsto=<a>
```

Note Because these services are provided by the respective companies, the URLs may be subject to change or the service could be discontinued at any time. Cisco takes no responsibility for the accuracy or use of any of this information. The URLs were obtained using an application called "ez-ipupdate," which is available for free on the Internet.

Additional References

The following sections provide references related to the Dynamic DNS Support for Cisco IOS Software feature.

Related Documents

Related Topic	Document Title
DNS Configuration Tasks	"Configuring DNS" module
DNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2136	*Dynamic Updates in the Domain Name System (DNS Update)*
RFC 3007	*Secure Domain Name System (DNS) Dynamic Update*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

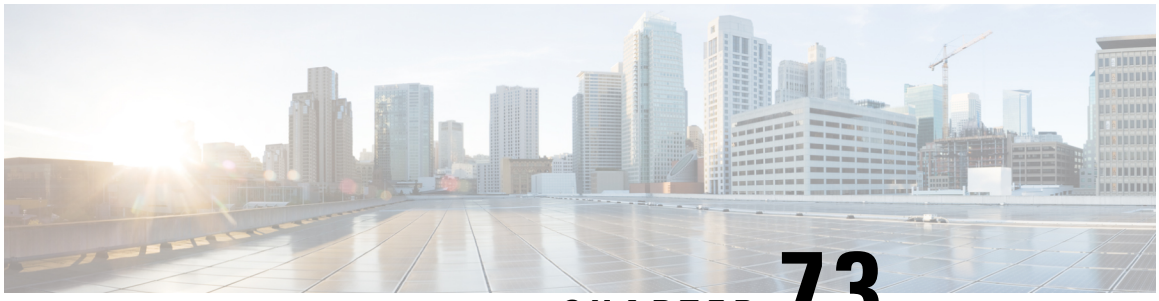
Feature Information for Dynamic DNS Support for Cisco IOS Software

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 114: Feature Information for Dynamic DNS Support for Cisco IOS Software

Feature Name	Releases	Feature Information
Dynamic DNS Support for Cisco IOS Software	12.3(8)YA 12.3(14)T	The Dynamic DNS Support for Cisco IOS Software feature enables Cisco IOS software devices to perform Dynamic Domain Name System (DDNS) updates to ensure that an IP host DNS name is correctly associated with its IP address.



CHAPTER 73

VRF-Aware DNS

The VRF-Aware DNS feature enables the configuration of a Virtual Private Network (VPN) routing and forwarding instance (VRF) table so that the domain name system (DNS) can forward queries to name servers using the VRF table rather than the named DNS server in the global IP address space. This feature allows DNS requests to be resolved within the appropriate Multiprotocol Label Switching (MPLS) VPN.

Information About VRF-Aware DNS

Domain Name System

Domain Name System (DNS) is a standard that defines a domain naming procedure used in TCP/IP. A domain is a hierarchical separation of the network into groups and subgroups with domain names identifying the structure. The named groups consist of named objects, usually devices like IP hosts, and the subgroups are domains. DNS has three basic functions:

- Name space: This function is a hierarchical space organized from a single root into domains. Each domain can contain device names or more specific information. A special syntax defines valid names and identifies the domain names.
- Name registration: This function is used to enter names into the DNS database. Policies are outlined to resolve conflicts and other issues.
- Name resolution: This function is a distributed client and server name resolution standard. The name servers are software applications that run on a server and contain the resource records (RRs) that describe the names and addresses of those entities in the DNS name space. A name resolver is the interface between the client and the server. The name resolver requests information from the server about a name. A cache can be used by the name resolver to store learned names and addresses.

A DNS server can be a dedicated device or a software process running on a device. The server stores and manages data about domains and responds to requests for name conflict resolutions. In a large DNS implementation, there can be a distributed database over many devices. A server can be a dedicated cache.

VRF Mapping and VRF-Aware DNS

To keep track of domain names, IP has defined the concept of a name server, whose job is to hold a cache (or database) of names appended to IP addresses. The cached information is important because the requesting DNS will not need to query for that information again, which is why DNS works well. If a server had to query each time for the same address because it had not saved any data, the queried servers would be flooded and would crash.

A gateway for multiple enterprise customers can be secured by mapping the remote users to a VRF domain. Mapping means obtaining the IP address of the VRF domain for the remote users. By using VRF domain mapping, a remote user can be authenticated by a VRF domain-specific AAA server so that the remote-access traffic can be forwarded within the VRF domain to the servers on the corporate network.

To support traffic for multiple VRF domains, the DNS and the servers used to resolve conflicts must be VRF aware. VRF aware means that a DNS subsystem will query the VRF name cache first, then the VRF domain, and store the returned RRs in a specific VRF name cache. Users are able to configure separate DNS name servers per VRF.

VRF-aware DNS forwards queries to name servers using the VRF table. Because the same IP address can be associated with different DNS servers in different VRF domains, a separate list of name caches for each VRF is maintained. The DNS looks up the specific VRF name cache first, if a table has been specified, before sending a query to the VRF name server. All IP addresses obtained from a VRF-specific name cache are routed using the VRF table.

How to Configure VRF-Aware DNS

Defining a VRF Table and Assigning a Name Server to Enable VRF-Aware DNS

Perform this task to define a VRF table and assign a name server.

A VRF-specific name cache is dynamically created if one does not exist whenever a VRF-specific name server is configured by using the **ip name-server vrf**command option or a permanent name entry is configured by using the **ip host vrf** command option. The VRF name cache is removed whenever all name server and permanent entries in the VRF are disabled.

It is possible that multiple name servers are configured with the same VRF name. The system will send queries to those servers in turn until any of them responds, starting with the server that sent a response the last time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*]
7. **ip domain lookup** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Defines a VRF table and enters VRF configuration mode. • The *vrf-name* argument can be up to 32 characters.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:21`	Creates routing and forwarding tables for a VRF.
Step 5	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode.
Step 6	**ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*] **Example:** `Router(config)# ip name-server vrf vpn1 172.16.1.111 172.16.1.2`	Assigns the address of one or more name servers to a VRF table to use for name and address resolution. • The **vrf** keyword is optional but must be specified if the name server is used with VRF. The *vrf-name* argument assigns a name to the VRF.
Step 7	**ip domain lookup** [**vrf** *vrf-name*] **Example:** `Router(config)# ip domain lookup vrf`	(Optional) Enables DNS-based address translation. • DNS is enabled by default. You only need to use this command if DNS has been disabled.

Mapping VRF-Specific Hostnames to IP Addresses

Perform this task to map VRF-specific hostnames to IP addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. Do one of the following:
   - **ip domain name** [**vrf** *vrf-name*] *name*
   - **ip domain list** [**vrf** *vrf-name*] *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ip domain name** [**vrf** *vrf-name*] *name*
	• **ip domain list** [**vrf** *vrf-name*] *name*
	Example:
	`Device(config)# ip domain name vrf vpn1 cisco.com`
	Example:
	`Device(config)# ip domain list vrf vpn1 cisco.com`
		Defines a default domain name that the Cisco IOS XE software will use to complete unqualified hostnames.
		or
		Defines a list of default domain names to complete unqualified hostnames.
		• You can specify a default domain name that the Cisco IOS XE software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any hostname that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up.
		• The **vrf** keyword and *vrf-name* argument specify a default VRF domain name.
		• The **ip domain list** command can be entered multiple times to specify more than one domain name to append when doing a DNS query. The system will append each in turn until it finds a match.

Configuring a Static Entry in a VRF-Specific Name Cache

Perform this task to configure a static entry in a VRF-specific name cache.

A VRF-specific name cache is dynamically created if one does not exist whenever a name server is configured for the VRF by using the **ip name-server vrf**command option or a permanent name entry is configured by using the **ip host vrf** command option. The VRF name cache is removed whenever all name server and permanent entries in the VRF are disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host vrf** [*vrf-name*] *name*[*tcp-port*] *address1*[*address2 ... address8*] [*mx ns srv*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip host vrf** [*vrf-name*] *name*[*tcp-port*] *address1*[*address2 ... address8*] [*mx ns srv*]
	Example:
	`Device(config)# ip host vrf vpn3 company1.com 172.16.2.1`
	`Device(config)# ip host test mx 1 mx_record`
	`Device(config)# ip host test ns ns_record`
	`Device(config)# ip host test srv 0 0 0 srv_record`
		Defines a static hostname-to-address mapping in the host cache.
		• The IP address of the host can be an IPv4 or IPv6 address, and the IP address can be associated with a Virtual Private Network (VPN) routing and forwarding (VRF) instance.
		• If the **vrf** keyword and *vrf-name* arguments are specified, then a permanent entry is created only in the VRF-specific name cache.
		• Mail exchanger (mx) identifies the mail server that is responsible for handling e-mails for a given domain name.
		• Name server (ns) state the authoritative name servers for the given domain.
		• Service (srv) records specifies the location of a service.

Verifying the Name Cache Entries in the VRF Table

Perform this task to verify the name cache entries in the VRF table.

SUMMARY STEPS

1. **enable**
2. **show hosts** [**vrf** *vrf-name*] {**all**| *hostname*} [**summary**]
3. **clear host** [**vrf** *vrf-name*] {**all**| *hostname*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show hosts** [**vrf** *vrf-name*] {**all**| *hostname*} [**summary**] **Example:** `Device# show hosts vrf vpn2`	• Displays the default domain name, the style of name lookup service, a list of name server hosts, the cached list of hostnames and addresses, and the cached list of hostnames and addresses specific to a particular Virtual Private Network (VPN). • The **vrf** keyword and *vrf-name* argument only display the entries if a VRF name has been configured. • If you enter the **show hosts** command without specifying any VRF, only the entries in the global name cache will display.
Step 3	**clear host** [**vrf** *vrf-name*] {**all**| *hostname*} **Example:** `Device# clear host vrf vpn2`	(Optional) Deletes entries from the hostname-to-address global address cache or VRF name cache.

Configuration Examples for VRF-Aware DNS

Example: VRF-Specific Name Server Configuration

The following example shows how to specify a VPN named vpn1 with the IP addresses of 172.16.1.111 and 172.16.1.2 as the name servers:

```
ip name-server vrf vpn1 172.16.1.111 172.16.1.2
```

Example: VRF-Specific Domain Name List Configuration

The following example shows how to add several domain names to a list in vpn1 and vpn2. The domain name is only used for name queries in the specified VRF.

```
ip domain list vrf vpn1 company.com
ip domain list vrf vpn2 school.edu
```

If there is no domain list, the domain name that you specified with the **ip domain name** global configuration command is used. If there is a domain list, the default domain name is not used. The **ip domain list** command is similar to the **ip domain name** command, except that with the **ip domain list**command you can define a list of domains, each to be tried in turn until a match is found.

VRF-Specific Domain Name Configuration Example

The following example shows how to define cisco.com as the default domain name for a VPN named vpn1. The domain name is only used for name queries in the specified VRF.

```
ip domain name vrf vpn1 cisco.com
```

Any IP hostname that does not contain a domain name (that is, any name without a dot) will have the dot and cisco.com appended to it before being looked up.

VRF-Specific IP Host Configuration Example

The following example shows how to define two static hostname-to-address mappings in the host cache for vpn2 and vpn3:

```
ip host vrf vpn2 host2 10.168.7.18
ip host vrf vpn3 host3 10.12.0.2
```

Additional References

Related Documents

Related Topic	Document Title
VRF-aware DNS configuration tasks: Enabling VRF-aware DNS, mapping VRF-specific hostnames to IP addresses, configuring a static entry in a VRF-specific hostname cache, and verifying the hostname cache entries in the VRF table	"VRF-Aware DNS" module
DNS configuration tasks	"Configuring DNS" module
DNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

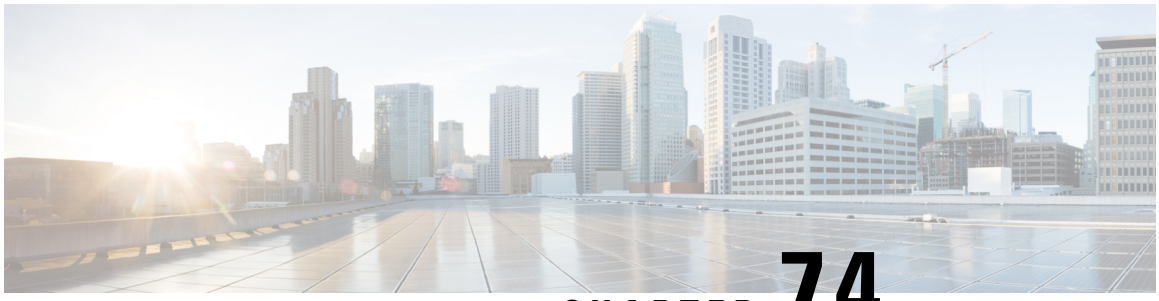
Feature Information for VRF-Aware DNS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 115: Feature Information for DNS

Feature Name	Releases	Feature Configuration Information
VRF-Aware DNS	Cisco IOS XE Release 2.1	The VRF-Aware DNS feature enables the configuration of a Virtual Private Network (VPN) routing and forwarding instance (VRF) table so that the domain name system (DNS) can forward queries to name servers using the VRF table rather than the named DNS server in the global IP address space. This feature allows DNS requests to be resolved within the appropriate Multiprotocol Label Switching (MPLS) VPN.



CHAPTER 74

Local Area Service Discovery Gateway

The Service Discovery Gateway feature enables multicast Domain Name System (mDNS) to operate across Layer 3 (L3) boundaries. An mDNS gateway will be able to provide transport for service discovery across L3 boundaries by filtering, caching and extending services from one subnet to another. Prior to implementation of this feature, mDNS was limited in scope to within a subnet due to the use of link-local scoped multicast addresses. This feature enhances Bring Your Own Device (BYOD).

Caution Extension of services should be done with proper care. Generally, only specific services should be extended. Service names should be unique in the network to avoid duplicate name conflicts.

See Feature Information for Service Discovery Gateway section to check feature availability for your platform release version.

Information About Service Discovery Gateway

Service Announcement Redistribution and Service Extension

Redistribution of announcements is the actual forwarding of announcements and query responses while service extension is the capability of proxying services between subnets. The actual replication of the service announcement can help to speed up the visibility of newly announced services and also a service's withdrawal if a service or device is turned off.

Note Extension of services such as printers or Apple TV works fine without actual replication of service announcements. The Service Discovery Gateway will cache announcements, queries and their responses in the cache. If another device queries for a service, the Service Discovery Gateway will be able to provide an answer from its cache.

Enable the **redistribution mdns-sd** command only on a per-interface basis, and only if it is actually required. You must ensure that there are no loops in the network topology corresponding to the interface for which service announcement redistribution is being enabled. A loop can lead to a broadcast storm.

Redistribution of service announcement information cannot be done globally. You can enable redistribution of service information only at the interface level.

Extending Services Across Subnets—An Overview

You need to enable a multicast Domain Name System (mDNS) gateway to extend services across subnet boundaries. You can enable an mDNS gateway for a device or for an interface. You must enable routing of services for the device before enabling it at the interface level. After the mDNS gateway is enabled on a device or interface, you can extend services across subnet boundaries.

To extend services across subnets, you must do the following:

1. Set Filter Options to Extend Services Across Subnets—You can allow services such as printer services to be accessed across subnets. If printer x is available on interface 1, users on interface 2 can use printer x without configuring the printer on their local systems.
2. Extend Services Across Subnets—The filter created in Step 1 should be applied on the interfaces 1 and 2. Only then can users on other interfaces access the printer service.

For the sample scenario where a printer service is accessible by clients on other interfaces, you must apply these filters:

- On the interface where the printer service is available (IN filter) —You want to allow the printer service *into* the mDNS cache, so that it can be accessed by users on other subnets.
- On the interface where the printer service is available (OUT filter)—Since clients on other interfaces will access the service (printer x, for example), you should allow queries coming from the device (OUT filter, from the device's point of view).
- On each interface where clients reside (IN filter)—For clients on other interfaces (subnets) wanting to access the printer service, you must allow queries from users into the mDNS cache (IN filter).

Remember Applying the IN filter means that you are allowing the printer service into the device mDNS cache, and other interfaces can access it. Applying the OUT filter means that you are allowing the queries out of the cache so that queries from clients on other interfaces can reach the printer interface. On other client-facing interfaces, the IN filter is applied to allow queries in.

Note

- Filters can be applied at the global level and at the interface level. Filters applied at the interface level takes precedence over the filters applied at the global level.
- The term 'service discovery information' refers to services (printer services, etc), queries (queries for printer services, etc, from one interface to the other), announcements (printer service is removed, etc), and service-instances (a specific service—printer x, Apple TV 3, etc) that you want to extend across subnets.

Set Filter Options to Extend Services Across Subnets

You can set filter options to allow services such as printer services into or out of a device or interface. You can also permit or prohibit queries, announcements, services learnt from an interface, specific service–instances, and locations. Use the **service-list mdns-sd** command to create a service-list and set filter options.

You need to create a service-list and use filter options within it. While creating a service-list, use one of the following options:

- The **permit** option permits specific services, announcements and service-instances across subnets.
- The **deny** option restricts services, announcements and service-instances from being transported across subnets.
- The **query** option is provided to browse services. For example, if you want to browse printer services periodically, then you can create a service-list with the **query** option, and add the printer service to the query. When you set a period for the query, the service entries are refreshed in the cache memory.

You must mention a sequence number when using the **permit** or **deny** option. The filtering is done sequentially, in the ascending order. The same service-list can be associated with multiple sequence numbers. Within a sequence, match statements (commands) must be used to specify what needs to be filtered. Generally, match statements are used to filter queries (for example, queries from clients to find printer and fax services), announcements (new service is added, and so on), specific service–instances, types of service such as printer services (so that the service is allowed into the cache for use), services available for a specific interface (printers and Apple TVs associated with a VLAN), and locations.

Note A service-list by itself does not contain any services. You must specify a service type in the match statement when setting filter options to allow or prohibit services. (For example, '_ipp._tcp' is the service type for an IPP printing service running over TCP).

Sample scenario - Consider a device is in a client segment. The goal is to allow the following on the device:

- All queries from clients to the device.
- Printer services to clients on other subnets.

The following example explains how to achieve the goal:

```
!
service-list mdns-sd mixed permit 10
  match message-type query
!
service-list mdns-sd mixed permit 20
  match message-type announcement
  match service-type _ipps._tcp.local
!
```

In the above example, a service-list called 'mixed' is created and the **permit** option is used twice—to filter queries and to filter printer services and announcements. The filtering is done in the sequence given below:

- Sequence 10 - A match statement is used to filter queries.
- Sequence 20 - Match statements are used to filter announcements and printer services.

The match statement in Sequence 10 sets a filter for queries on the device, but does not specify that queries be allowed *into* the device. To allow queries from clients, the filter needs to be applied on the interface in the IN direction. The example is displayed in the Extend Services Across Subnets section.

Similarly, the match statements in Sequence 20 sets a filter for announcements and printer services on the device, but does not specify that they be allowed *into* the device. To allow announcements and printer services into the device, the filter needs to be applied on the required interfaces in the IN direction. The example is displayed in the Extend Services Across Subnets section.

If neither the **permit** option nor the **deny** option is used, the default action is to disallow services from being transported to other subnets.

Browsing services periodically—Service-lists of the type **query** can be used to browse services. Such queries are called active queries. Active queries periodically send out requests for the services specified within the query on all interfaces. As services have a specific Time to Live (TTL) duration, active queries can help to keep services fresh in the cache memory.

In the following example, a service-list named 'active-query' is created and the service-list is of the type **query**. Services such as printer services are specified within the query, and these are the services that we want to extend. Typically, these services would match the services that have been configured as 'permitted' services in the IN filter.

```
!
service-list mdns-sd active-query query
  service-type _universal._sub._ipp._tcp
  service-type _ipp._tcp.local
  service-type _ipps._tcp.local
  service-type _raop._tcp.local
!
```

The purpose of an active query and a query associated with a match statement is different. When you enable an active query, services are browsed periodically. A query is used in a match statement to permit or prohibit queries (not active queries) on the interface.

Note

- Service-list creation can only be used globally and cannot be used at the interface level.
- You can create a new service-instance of a specific service-type using the **service-instance mdns-sd** command.
- A service end–point (such as a printer, fax, and so on) sends unsolicited announcements when a service starts up. After that, it sends unsolicited announcements whenever a network change event occurs (such as, an interface coming up or going down, and so on). The device always responds to queries.

Remember Filtering only sets filter options and specifies that certain services need to be filtered. You must *apply* the filters on an interface for the services, queries, or announcements to actually be permitted or prohibited on the interface. To know about applying filters and the other available service discovery configuration options, refer the Extend Services Across Subnets section.

Extend Services Across Subnets

You must have set filter options for the device before extending services across subnets. If you have set filter options for specific services and other service discovery information to be allowed, prohibited or queried periodically, you can apply the filters for an interface.

Before applying filters, note the following:

- You must enable multicast Domain Name System (mDNS) on a device to apply filter options. You can enable mDNS using the command **service-routing mdns-sd**
- Since you might want to allow services into the device or prohibit services from being learnt on an interface, you must apply the filter in the needed direction. The options **IN** and **OUT** perform the desired actions on the interface.
- Typically, a service-policy is applied on an interface. Global service-policies are optional and affect all L3 interfaces.

Sample scenario - A device is in a client segment and the goal is to allow the following between the device interfaces:

- All queries from clients to the device.
- Printer services.

A note about filter options - Filter options have been set for the above scenario by creating a service-list called 'mixed' and adding filter options to it. (see Set Filter Options to Extend Services Across Subnets for more details). The following example explains how to apply the filters:

```
!
interface Ethernet0/0
 description *** (wireless) Clients here plus some printers
 ip address 172.16.33.7 255.255.255.0
 service-routing mdns-sd
  service-policy mixed IN
!
interface Ethernet0/3
 description *** (wireless) Clients here plus some printers
 ip address 172.16.57.1 255.255.255.0
 service-routing mdns-sd
  service-policy mixed IN
!
```

In the above example, service-routing is enabled on the interface and the filter options in the service-policy 'mixed' are applied in the **IN** direction. In other words, all queries and printer services will be allowed into the device, from the interfaces Ethernet 0/0 and Ethernet 0/3.

Sample scenario for browsing specific services - A service-list of the type **query** (called active query) has been created. It contains services that we want to browse periodically, such as printer services (see Set Filter Options to Extend Services Across Subnets for more details about creating an active query). To enable browsing of the services in the query, you must apply the active query for the device.

```
!
service-routing mdns-sd
 service-policy-query active-query 90
!
```

In the above example, the period is set to 90 seconds. The services within the active query are queried on all interfaces of the device after an interval of 90 seconds.

Note

- You can enable browsing of services for specific interfaces. If browsing of services is enabled globally, you can disable browsing of services on specific interfaces.
- Services are browsed specific to a device or interface by the mDNS process. So, the IN or OUT option is not relevant for browsing of services.

You can use the following options after enabling mDNS on a device or interface.

Purpose	**Use this Command** **Note** The complete syntax is provided in the corresponding task.	**Global and Interface Configuration Options**
For a service-list, apply a filter to allow or prohibit services.	**service-policy**	Global and interface levels.
Set some part of the system memory for cache.	**cache-memory-max**	Global level.
Configure an active query and the query period so that specified services are queried periodically.	**service-policy-query**	
Designate a specific device or interface in a domain for routing mDNS announcement and query information.	**designated-gateway**	Global and interface levels.
Access services in the proximity of the device. **Note** Service policy proximity filtering functionality is only available on wireless devices and their interfaces.	**service-policy-proximity**	Global and interface levels.
Configure service-type enumeration period for the device.	**service-type-enumeration period**	Global level.
Specify an alternate source interface for outgoing mDNS packets on a device.	**source-interface**	Global level.
Configure the maximum rate limit of incoming mDNS packets for a device.	**rate-limit**	Global level.

Speed up visibility of newly announced services and withdrawal of services when a service or device is turned off.	**redistribute**	Interface level.

How to Configure Service Discovery Gateway

Setting Filter Options for Service Discovery

Before you begin

Ensure that you permit a query or announcement when you set filter options. If you do not use a **permit** option and only use **deny** options, you will not be able to apply the filter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**}
4. **match message-type** {**announcement** | **any** | **query**}
5. **match service-instance** {*instance-name* | **any** | **query**}
6. **match service-type** *mDNS-service-type-string*
7. **match location civic** *civic-location-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**} **Example:** `Device(config)# service-list mdns-sd sl1 permit 3`	Enters mdns service discovery service-list mode. • Creates a service-list and applies a filter on the service-list according to the **permit** or **deny** option applied to the sequence number. Or

	Command or Action	Purpose
	Or `Device(config)# service-list mdns-sd sl4 query`	• Creates a service-list and associates a query for the service-list name if the **query** option is used. **Remember** When you set filter options, ensure that you permit a query or announcement for a service-list. If you do not use a **permit** option and only use **deny** options, you will not be able to apply the filter.
Step 4	**match message-type** {**announcement** | **any** | **query**} **Example:** `Device(config-mdns-sd-sl)# match message-type announcement`	Configures parameters for a service-list based on a service announcement or query. **Note** You cannot use the **match** command if you have used the **query** option. The **match** command can be used only for the **permit** or **deny** option.
Step 5	**match service-instance** {*instance-name* | **any** | **query**} **Example:** `Device(config-mdns-sd-sl)# match service-instance printer-3`	Configures parameters for a service-list based on a service-instance or query.
Step 6	**match service-type** *mDNS-service-type-string* **Example:** `Device(config-mdns-sd-sl)# match service-type _ipp._tcp.local`	Configures parameters for a service-list based on a service-type.
Step 7	**match location civic** *civic-location-name* **Example:** `Device(config-mdns-sd-sl)# match location civic location3`	Configures parameters for a service-list based on a civic location.
Step 8	**exit** **Example:** `Device(config-mdns-sd-sl)# exit`	Exits mdns service discovery service-list mode, and returns to global configuration mode.

What to do next

Apply filters on an interface for the services, queries, or announcements to actually be permitted or prohibited on the interface.

Applying Service Discovery Filters and Configuring Service Discovery Parameters

After enabling multicast Domain Name System (mDNS) gateway for a device, you can apply filters (IN-bound filtering or OUT-bound filtering) and active queries by using **service-policy** and **service-policy-query** commands, respectively.

Note Steps 5 to 11 are mDNS Service Discovery configuration options. The steps are optional and not meant to be used in any specific order.

Before you begin

You must set filter options for the device before applying filters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-routing mdns-sd**
4. **service-policy** *service-policy-name* {**IN** | **OUT**}
5. **cache-memory-max** *cache-config-percentage*
6. **service-policy-query** *service-list-name query-period*
7. **designated-gateway enable** [**ttl** *duration*]
8. **service-policy-proximity** *service-list-name* [**limit** *number-of-services*]
9. **service-type-enumeration period** *period-value*
10. **source-interface** *type number*
11. **rate-limit in** *maximum-rate-limit*
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-routing mdns-sd** **Example:**	Enables mDNS gateway functionality for a device and enters multicast DNS configuration (config-mdns) mode.

	Command or Action	Purpose
	`Device(config)# service-routing mdns-sd`	
Step 4	**service-policy** *service-policy-name* {**IN** | **OUT**} **Example:** `Device(config-mdns)# service-policy sl1 IN`	For a service-list, applies a filter on incoming service discovery information (IN-bound filtering) or outgoing service discovery information (OUT-bound filtering). **Note** Global service-policies are optional and effect all L3 interfaces. Typically, a service-policy is applied on an interface.
Step 5	**cache-memory-max** *cache-config-percentage* **Example:** `Device(config-mdns)# cache-memory-max 20`	Sets some part of the system memory (in percentage) for cache. **Note** By default, 10% of the system memory is set aside for cache. You can override the default value by using this command.
Step 6	**service-policy-query** *service-list-name query-period* **Example:** `Device(config-mdns)# service-policy-query sl4 100`	Creates an active query and configures the service-list-query period.
Step 7	**designated-gateway enable** [**ttl** *duration*] **Example:** `Device(config-mdns)# designated-gateway enable`	Designates the device to route mDNS announcement and query information for the domain.
Step 8	**service-policy-proximity** *service-list-name* [**limit** *number-of-services*] **Example:** `Device(config-mdns)# service-policy-proximity sl1 limit 10`	Configures service policy proximity filtering on the device. • Service policy proximity filtering is only available for wireless clients and is based on Radio Resource Management (RRM). Wired clients and services are not affected by the limit. • The default value for the maximum number of services that can be returned is 50.
Step 9	**service-type-enumeration period** *period-value* **Example:** `Device(config-mdns)# service-type-enumeration period 45`	Configures service-type enumeration period for the device.
Step 10	**source-interface** *type number* **Example:**	Specifies an alternate source interface for outgoing mDNS packets on a device.
Step 11	**rate-limit in** *maximum-rate-limit* **Example:** `Device(config-mdns)# rate-limit in 80`	Configures the maximum rate limit of incoming mDNS packets for a device.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-mdns)# exit`	Exits multicast DNS configuration mode, and returns to global configuration mode.

Applying Service Discovery Filters for an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-routing mdns-sd**
5. **service-policy** *service-policy-name* {**IN** | **OUT**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters Interface multicast DNS configuration mode, and enables interface configuration.
Step 4	**service-routing mdns-sd** **Example:** `Device(config-if)# service-routing mdns-sd`	Enables mDNS gateway functionality for an interface and enters multicast DNS configuration (config-mdns) mode.
Step 5	**service-policy** *service-policy-name* {**IN**	**OUT**} **Example:** `Device(config-if-mdns-sd)# service-policy sl1 IN`	For a service-list, applies a filter on incoming service discovery information (IN-bound filtering) or outgoing service discovery information (OUT-bound filtering).

	Command or Action	Purpose
		Remember When you set filter options, ensure that you permit a query or announcement for a service-list. If you have not permitted a service, query, or announcement while setting filter options, then you will see this warning when you apply the filter: **Warning**: Please enable explicit service-list rule with the permit action to allow queries and responses.
Step 6	**exit** **Example:** `Device(config-if-mdns-sd)# exit`	Exits Interface multicast DNS configuration mode, and returns to interface configuration mode.

Creating a Service Instance

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-instance mdns-sd service** *instance-name* **regtype** *service-type* **domain** *name*
4. {**ipv4addr** | **ipv6addr**} *IP-address*
5. **port** *number*
6. **target-hostname** *host-name*
7. **txt** *text-record-name*
8. **priority** *value*
9. **weight** *value*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**service-instance mdns-sd service** *instance-name* **regtype** *service-type* **domain** *name* **Example:** `Device(config)# service-instance mdns-sd service printer-3 regtype _ipp._tcp.local domain tcp4`	Creates a service-instance of a specific service type and enters multicast Domain Name System (mDNS) service discovery service-instance (config-mdns-sd-si) mode. **Note** In this mode, you can configure various parameters for the service-instance. The subsequent steps show how to configure service-instance parameters.
Step 4	{**ipv4addr** | **ipv6addr**} *IP-address* **Example:** `Device(config-mdns-sd-si)# ipv4addr 209.165.200.230 255.255.255.0`	Specifies the IPv4 or IPv6 address of the port on which the service is available.
Step 5	**port** *number* **Example:** `Device(config-mdns-sd-si)# port 9100`	Specifies the port on which the service is available.
Step 6	**target-hostname** *host-name* **Example:** `Device(config-mdns-sd-si)# target-hostname fqdn-of-printer.example.com.`	Specifies the fully qualified domain name (FQDN) of the target host.
Step 7	**txt** *text-record-name* **Example:** `Device(config-mdns-sd-si)# txt _ipp._tcp.local=printer3`	Specifies the text record associated with the service instance. **Note** A TXT record is a type of DNS record that provides text information to sources outside your domain. Specify the text record in the format 'service-type=service-name'. To specify multiple records, use a semicolon (;) as a separator.
Step 8	**priority** *value* **Example:** `Device(config-mdns-sd-si)# priority 3`	(Optional) Specifies the priority value for the service-instance. The default priority value is zero.
Step 9	**weight** *value* **Example:** `Device(config-mdns-sd-si)# weight 20`	(Optional) Specifies the weight value for the service-instance. The default weight value is zero.
Step 10	**exit** **Example:** `Device(config-mdns-sd-si)# exit`	Exits multicast Domain Name System (mDNS) service discovery service-instance (config-mdns-sd-si) mode and enters global configuration mode.

Verifying and troubleshooting Service Discovery Gateway

Note The show and debug commands mentioned below are not in any specific order.

SUMMARY STEPS

1. **show mdns requests** [**detail** | [**type** *record-type*] [**name** *record-name*]]
2. **show mdns cache** [**interface** *type number* [**detail**] | [**name** *record-name*] [**type** *record-type*] [**detail**]]
3. **show mdns statistics** {**all** | **interface** *type number* | **service-list** *list-name* | [**cache** | **service-policy**] {**all** | **interface** *type number*} | **services orderby providers**}
4. **show mdns service-types** [**all** | **interface** *type number*]
5. **debug mdns** {**all** | **error** | **event** | **packet** | **verbose**}

DETAILED STEPS

Step 1 **show mdns requests** [**detail** | [**type** *record-type*] [**name** *record-name*]]

Example:

```
Device# show mdns requests detail

MDNS Outstanding Requests
==================================================
Request name  :   _ipp._tcp.local
Request type  :   PTR
Request class :   IN
```

This command displays information for outstanding multicast Domain Name System (mDNS) requests, including record name and record type information.

Step 2 **show mdns cache** [**interface** *type number* [**detail**] | [**name** *record-name*] [**type** *record-type*] [**detail**]]

Example:

Note You can use the **detail** keyword for a specific interface, record or type. You cannot use it independently with the **show mdns cache** command.

```
Device# show mdns cache

mDNS CACHE
==========================================================================================
[<NAME>]                                                [<TYPE>][<CLASS>] [<TTL>/Remaining] [Accessed]
 [If-index] [<RR Record Data>]


_services._dns-sd._udp.local                              PTR     IN      4500/4496              0
     3       _ipp._tcp.local

_ipp._tcp.local                                           PTR     IN      4500/4496              1
     3       printer1._ipp._tcp.local
```

```
printer1._ipp._tcp.local                                SRV     IN      120/116         1       3
     0       0       5678    much-WS.local

printer1._ipp._tcp.local                                TXT     IN      4500/4496               1
     3       (1)''

music-WS.local                                        A       IN      120/116         1       3
  192.168.183.1
```

This command displays mDNS cache information.

Step 3 **show mdns statistics** {**all** | **interface** *type number* | **service-list** *list-name* | [**cache** | **service-policy**] {**all** | **interface** *type number*} | **services orderby providers**}

Example:

```
Device# show mdns statistics all

mDNS Statistics
 mDNS packets sent       : 0
 mDNS packets received   : 31
 mDNS packets dropped    : 8
 mDNS cache memory in use: 64264(bytes)
```

This command displays mDNS statistics.

Step 4 **show mdns service-types** [**all** | **interface** *type number*]

Example:

```
Device# show mdns service-types

mDNS SERVICES
================================
[<NAME>]          [<TTL>/Remaining] [If-name]
_ipp._tcp.local     4500/4496
```

This command displays mDNS statistics.

Step 5 **debug mdns** {**all** | **error** | **event** | **packet** | **verbose**}

Example:

```
Device# debug mdns all
```

This command enables all mDNS debugging flows.

Configuration Examples for Service Discovery Gateway

Example: Setting Filter Options for Service Discovery

The following example shows creation of a service-list sl1. The permit option is being applied on sequence number 3 and all services with message-type announcement are filtered and available for transport across various subnets associated with the device.

```
Device> enable
Device# configure terminal
Device(config)# service-list mdns-sd sl1 permit 3
Device(config-mdns-sd-sl)# match message-type announcement
Device(config-mdns-sd-sl)# exit
```

Example: Applying Service Discovery Filters and Configuring Service Discovery Parameters

```
Device> enable
Device# configure terminal
Device(config)# service-routing mdns-sd
Device(config-mdns)# service-policy serv-pol1 IN
Device(config-mdns)# cache-memory-max 20
Device(config-mdns)# service-policy-query sl-query1 100
Device(config-mdns)# designated-gateway enable
Device(config-mdns)# rate-limit in 80
Device(config-mdns)# exit
```

Example: Applying Service Discovery Filters for an Interface

Example: Setting Multiple Service Discovery Filter Options

The following example shows creation of filters using service-lists mixed, permit-most, permit-all, and deny-all. Then, the filters are applied at various interfaces, as required.

```
!
service-list mdns-sd mixed permit 10
 match message-type query
!
service-list mdns-sd mixed permit 20
 match message-type announcement
 match service-type _ipps._tcp.local
!
service-list mdns-sd mixed permit 30
 match message-type announcement
 match service-type _ipp._tcp.local
 match service-type _universal._sub._ipp._tcp
!
service-list mdns-sd mixed permit 40
 match message-type announcement
!
service-list mdns-sd mixed deny 50
!
service-list mdns-sd permit-most deny 10
 match service-type _sleep-proxy._udp.local
!
service-list mdns-sd permit-most permit 20
!
service-list mdns-sd permit-all permit 10
```

```
!
service-list mdns-sd deny-all permit 10
 match message-type query
!
service-list mdns-sd deny-all deny 20
!
service-list mdns-sd active-query query
 service-type _universal._sub._ipp._tcp.local
 service-type _ipp._tcp.local
 service-type _ipps._tcp.local
 service-type _raop._tcp.local
!
service-routing mdns-sd
 service-policy-query active-query 900
!
!
interface Ethernet0/0
 description *** (wireless) Clients here plus some printers or aTVs
 ip address 172.16.33.7 255.255.255.0
 service-routing mdns-sd
  service-policy mixed IN
  service-policy permit-all OUT
!
interface Ethernet0/1
 description *** AppleTVs, Print Servers here
 ip address 172.16.57.1 255.255.255.0
 service-routing mdns-sd
  service-policy permit-most IN
  service-policy permit-all OUT
!
interface Ethernet0/2
 description *** Clients only, we don't want to learn anything here
 ip address 172.16.58.1 255.255.255.0
 service-routing mdns-sd
  service-policy deny-all IN
  service-policy permit-all OUT
!
interface Ethernet0/3
 no ip address
 shutdown
!
```

In the above example, the service-lists are:

- permit-all - As the name suggests, this service-list permits all resource records, and should be used with care. This is typically applied in the OUT direction; allows the cache to respond to all requests regardless of query content or query type.
- permit-most - This allows anything in, except for sleep-proxy services. This is because extending sleep-proxy services causes an issue with devices that register with a sleep proxy across the Service Discovery Gateway. Due to split horizon, the real (sleeping) device won't be able to re-register its services when waking up again when its pointer (PTR) record is pointing to the sleep-proxy.
- deny-all - This prevents the cache from learning anything. Again incoming on a segment where only clients live. As a result, clients will be able to query for services from the cache (hence the permit 10 match query), but there is no need to learn anything from the clients.
- mixed - This is created to be used in client segments. In addition to clients (such as iPads, PCs, and so on), the occasional printer or a TV will also connect. The purpose here is to learn about those specific services but not about services the clients provide. The filter applied is IN. As a result, the following actions are applicable:

- Allow every query IN.
- Allow specific services in (such as printer services [IPP]).
- Deny everything else.

In addition, to keep the service PTRs fresh in the cache an active query is configured. The active query queries for those services that we want to extend. Typically, this would match the services that have been configured as 'permitted' services in the IN filter. The value is set to 900 seconds. The duration is enough to refresh the PTRs as they typically have a TTL of 4500 seconds.

Example: Creating a Service Instance

```
Device> enable
Device# configure terminal
Device(config)# service-instance mdns-sd service printer-3 regtype _ipp._tcp.local domain
tcp4
Device(config-mdns-sd-si)# ipv4addr 209.165.200.230 255.255.255.0
Device(config-mdns-sd-si)# port 9100
Device(config-mdns-sd-si)# target-hostname fqdn-of-printer.example.com.
Device(config-mdns-sd-si)# txt _ipp._tcp.local=printer3
Device(config-mdns-sd-si)# priority 3
Device(config-mdns-sd-si)# weight 20
Device(config-mdns-sd-si)# exit
```

Note When you create a service-instance, a text record is created even if you do not configure service-instance parameters.

Additional References for Service Discovery Gateway

Related Documents

Related Topic	Document Title
Master Command List	Cisco IOS Master Command List
IP Addressing Services Command Reference	Cisco IOS IP Addressing Services Command Reference
Configuring DNS	IP Addressing: DNS Configuration Guide
DNS conceptual information	“Information About DNS” section in IP Addressing: DNS Configuration Guide

Standards and RFCs

Standard/RFC	Title
RFC 6762	Multicast DNS

Standard/RFC	Title
RFC 6763	DNS-Based Service Discovery
Multicast DNS Internet-Draft	Multicast DNS Internet draft

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Service Discovery Gateway

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

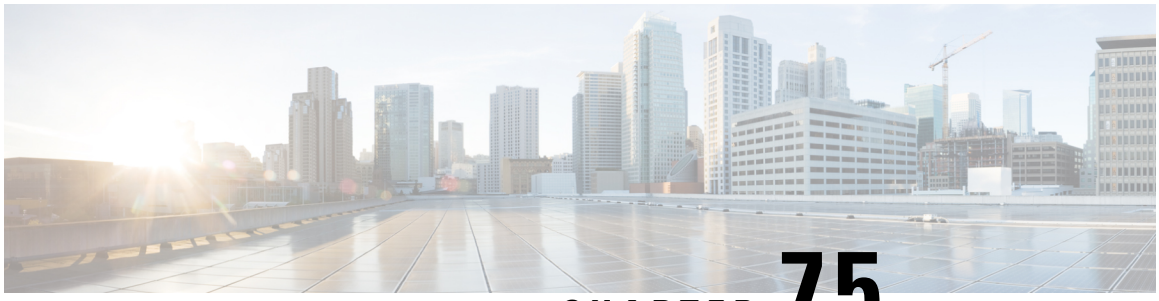
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 116: Feature Information for Service Discovery Gateway

Feature Name	Releases	Feature Information
Service Discovery Gateway		The Service Discovery Gateway feature enables multicast Domain Name System (mDNS) to operate across L3 boundaries (different subnets). The following commands were introduced or modified: **cache-memory-max, clear mdns cache, clear mdns statistics, debug mdns, match message-type, match service-instance, match service-type, redistribute mdns-sd, service-list mdns-sd, service-policy, service-policy-query, service-routing mdns-sd, show mdns cache, show mdns requests, show mdns statistics**
Service Discovery Gateway—Phase 2		The Service Discovery Gateway feature was enhanced with additional filter and configuration options. The following commands were introduced or modified: **clear mdns cache, clear mdns service-types, clear mdns statistics, designated-gateway, match location, rate-limit, service-instance mdns-sd, service-policy-proximity, service-routing mdns-sd, service-type-enumeration, show mdns cache, show mdns statistics, source-interface**
Service Discovery Gateway—Phase 3		The Service Discovery Gateway feature was enhanced with the following features: • **De-congestion of incoming mDNS traffic using the rate limiting mechanism**—The rate-limit value range was reset to1-100 p/s. • **Redistribution of service-withdrawal announcements across subnets when services are withdrawn, to improve mDNS cache efficiency and to avoid message loops**—The **withdraw-only** option was added to the **redistribute mdns-sd** command. • **A filter criterion for services available and learnt on a specific interface**—The **match learnt-interface** command was added to filter services. • **Enabling and disabling of periodic browsing of services on specific interfaces**—The **service-policy-query (interface)** command was added. For existing, globally configured active queries, the **disable** option was added to disable browsing of services on an interface, retaining the configurations on other interfaces. The following commands were introduced or modified: **match learnt-interface, rate-limit, redistribute mdns-sd, service-policy-query (interface)**

PART VIII

NAT



CHAPTER 75

Configuring NAT for IP Address Conservation

This module describes how to configure Network Address Translation (NAT) for IP address conservation and how to configure inside and outside source addresses. This module also provides information about the benefits of configuring NAT for IP address conservation.

NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks. Before packets are forwarded onto another network, NAT translates the private (not globally unique) addresses in the internal network into legal addresses. NAT can be configured to advertise to the outside world only one address for the entire network. This ability provides more security by effectively hiding the entire internal network behind that one address.

NAT is also used at the enterprise edge to allow internal users access to the Internet. It allows Internet access to internal devices such as mail servers.

Prerequisites for Configuring NAT for IP Address Conservation

Access Lists

All access lists that are required for use with the configuration tasks that are described in this module must be configured before initiating a configuration task. For information about how to configure an access list, see the *IP Access List EntrySequence Numbering* document.

Note If you specify an access list with a NAT command, NAT will not support the **permit ip any any** command. This command is commonly used in an access list.

NAT Requirements

Before configuring NAT in your network, ensure that you know the interfaces on which NAT is configured and for what purposes. The following requirements help you decide how to configure and use NAT:

- Define the NAT inside and outside interfaces if:
  - Users exist off multiple interfaces.
  - Multiple interfaces connect to the internet.
- Define what you need NAT to accomplish:
  - Allow internal users to access the internet.
  - Allow the internet to access internal devices such as a mail server.
  - Allow overlapping networks to communicate.
  - Allow networks with different address schemes to communicate.
  - Allow networks with different address schemes to communicate.
  - Redirect TCP traffic to another TCP port or address.
  - Use NAT during a network transition.

From Cisco IOS XE Denali 16.3 release, NAT support is introduced on Bridge Domain Interface (BDI) for enabling NAT configuration on the BDI interface.

Restrictions for Configuring NAT for IP Address Conservation

- When you configure Network Address Translation (NAT) on an interface, that interface becomes optimized for NAT packet flow. Any nontranslated packet that flows through the NAT interface goes through a series of checks to determine whether the packet must be translated or not. These checks result in increased latency for nontranslated packet flows and thus negatively impact the packet processing latency of all packet flows through the NAT interface. We highly recommend that a NAT interface must be used only for NAT-only traffic. Any non-NAT packets must be separated and these packets must go through an interface that does not have NAT configured on it. You can use Policy-Based Routing (PBR) for separating non-NAT traffic.
- NAT Virtual Interfaces (NVIs) are not supported in the Cisco IOS XE software.
- In Cisco IOS XE software, NAT outside interfaces show up in the translations tables, by default. This view of NAT outside interfaces causes the connection that originates from the outside interface of the device to fail. To restore connectivity, you must explicitly deny the outside Interface within the NAT ACL using the **deny** command. After using the **deny** command, no translation is observed for the outside interface.
- NAT is not practical if large numbers of hosts in the stub domain communicate outside of the domain.
- Some applications use embedded IP addresses in such a way that translation by a NAT device is impractical. These applications may not work transparently or at all through a NAT device.

- In a NAT configuration, addresses configured for any inside mapping must not be configured for any outside mapping.
- Do not configure the interface IP address as part of the IP address NAT pool.
- By default, support for the Session Initiation Protocol (SIP) is enabled on port 5060. Therefore, NAT-enabled devices interpret all packets on this port as SIP call messages. If other applications in the system use port 5060 to send packets, the NAT service may corrupt the packet. This packet corruption is due to its attempt to interpret the packet as a SIP call message.
- NAT hides the identity of hosts, which may be an advantage or a disadvantage depending on the needed result.
- Devices that are configured with NAT must not advertise the local networks to outside the network. However, routing information that NAT receives from the outside can be advertised in the stub domain as usual.
- NAT outside interface is not supported on a VRF. However, NAT outside interface is supported in iWAN and is part of the Cisco Validated Design.
- For VRF-aware NAT, remove the NAT configuration before you remove the VRF configuration.
- If you specify an access list to use with a NAT command, NAT does not support the **permit ip any any** command. This NAT command is commonly used in the access list.
- This platform does not support an access list with a port range.
- NAT configuration is not supported on the access side of the Intelligent Services Gateway (ISG).
- Using any IP address that is configured of a device as an address pool or in a NAT static rule is not supported. NAT can share the physical interface address (not any other IP address) of a device only by using the NAT interface overload configuration. A device uses the ports of its physical interface and NAT must receive communication about the ports that it can safely use for translation. This communication happens only when the NAT interface overload is configured.
- The output of the **show ip nat statistics** command displays information about all IP address pools and NAT mappings that you have configured. If your NAT configuration has a high number of IP address pools and NAT mappings, the update rate of the pool and mapping statistics in **show ip nat statistics** is slow. For example, NAT configuration output with 1000 to 4000 NAT mappings.
- Static and dynamic NAT with generic routing encapsulation (generic GRE) and dynamic NAT with Layer 2 do not work when used along with hardware-based Cisco AppNav appliances such as, Wide Area Application Services (WAAS). In the context of WAAS, generic GRE is an out of path deployment mechanism. It helps to return packets from the WAAS Wide-Area Application Engine (WAE) through the GRE tunnel to the same device from which they were originally redirected after completing optimization.
- Port Address Translation (also called NAT overload) only supports protocols whose port numbers are known; these protocols are Internet Control Message Protocol (ICMP), TCP, and UDP. Other protocols do not work with PAT because they consume the entire address in an address pool. Configure your access control list to only permit ICMP, TCP, and UDP protocols, so that all other protocol traffic is prevented from entering the network.
- NAT, Zone-Based Policy Firewall, and Web Cache Communication Protocol (WCCP) cannot coexist in a network.

- Non-Pattable traffic, is traffic for a protocol where there are no ports. PAT/Overload can only be done on protocols where the ports are known, that is, UDP, TCP, and ICMP.

  When NAT overload (PAT) is configured and Non-Pattable traffic hits the router, Non-Pattable BIND entry gets created for this traffic. Following is a bind entry in the NAT table:

  ```
  ---  213.252.7.132         172.16.254.242        ---
  ```

  This bind entry consumes an entire address from the pool. In this example, 213.252.7.132 is an address from an overloaded pool.

  That means an inside local IP Address gets bound to the outside global IP which is similar to static NAT. Because of this binding action, new inside local IP Addresses cannot use this global IP Address until the current entry gets timed out. All the translation that is created off this BIND is 1-to-1 translations instead of overload.

  To avoid consumption of an entire address from the pool, make sure that there are not any entries for the Non-Pattable traffic across the router.

- When configuring NAT with ACLs or route maps, the ACLs or route maps must not overlap. If the ACLs or route maps overlap, NAT cannot map to the required transition.

Information About Configuring NAT for IP Address Conservation

Benefits of Configuring NAT for IP Address Conservation

NAT allows organizations to resolve the problem of IP address depletion when they have existing networks and must access the Internet. Sites that do not yet possess Network Information Center (NIC)-registered IP addresses must acquire them. If more than 254 clients are present or planned, the scarcity of Class B addresses becomes a serious issue. Cisco IOS XE NAT addresses these issues by mapping thousands of hidden internal addresses to a range of easy-to-get Class C addresses.

Sites that already have registered IP addresses for clients on an internal network may want to hide those addresses from the Internet. This action disable hacker to directly attack the clients. With clients addresses hidden, an extent of security is established. Cisco IOS XE NAT gives LAN administrators complete freedom to expand Class A addressing. The Class A addressing expansion is drawn from the reserve pool of the Internet Assigned Numbers Authority (RFC 1597). This expansion occurs within the organization without concern for addressing changes at the LAN/Internet interface.

The Cisco IOS XE software can selectively or dynamically perform NAT. This flexibility allows the network administrator to use a mix of RFC 1597 and RFC 1918 addresses or registered addresses. NAT is designed for use on various devices for IP address simplification and conservation. In addition, Cisco IOS XE NAT allows the selection of internal hosts that are available for NAT.

A significant advantage of NAT is that it can be configured without requiring changes to hosts or devices in the network. However, changes are required on few other devices where NAT is configured.

In Cisco IOS XE Denali 16.3 release, Multi-Tenant support for NAT feature was introduced. With Multi-Tenant support, the configuration changes of a Virtual Routing and Forwarding (VRF) instance does not interrupt the traffic flow of other VRFs in the network.

NAT is a feature that allows the IP network of an organization to appear, from the outside, to be using a different IP address space than the one that it is actually using. Thus, NAT allows an organization with nonglobally routable addresses to connect to the Internet by translating those addresses into a globally routable

address space. NAT also allows a graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into classless interdomain routing (CIDR) blocks. NAT is described in RFC 1631.

How NAT Works

A device that is configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit device between a stub domain and the backbone. When a packet exits the domain, NAT translates the locally significant source address into a globally unique address. When a packet enters the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. If NAT cannot allocate an address because it has run out of addresses, it drops the packet. Then, NAT sends an Internet Control Message Protocol (ICMP) host unreachable packet to the destination.

Uses of NAT

NAT can be used for the following scenarios:

- Connect to the internet when all your hosts do not have globally unique IP addresses. Network Address Translation (NAT) enables private IP networks that use nonregistered IP addresses to connect to the Internet. NAT is configured on a device at the border of a stub domain (mentioned as the *inside network*) and a public network such as the Internet (mentioned as the *outside network*). NAT translates internal local addresses to globally unique IP addresses before sending packets to the outside network. As a solution to the connectivity problem, NAT is practical only when relatively few hosts in a stub domain communicate simultaneously outside the domain. When outside communication is necessary, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses. Also, these addresses can be reused when they are no longer in use.
- Change your internal addresses. Instead of changing the internal addresses, which can be a considerable amount of work, you can translate them by using NAT.
- For basic load-sharing of TCP traffic. You can map a single global IP address with many local IP addresses by using the TCP Load Distribution feature.

Types of NAT

NAT operates on a router—generally connecting only two networks. Before any packets are forwarded to another network, NAT translates the private (inside local) addresses within the internal network into public (inside global) addresses. This functionality gives you the option to configure NAT so that it advertises only a single address for your entire network to the outside world. Doing this translation, NAT effectively hides the internal network from the world, giving you some additional security.

The types of NAT include:

- Static address translation (static NAT)—Allows one-to-one mapping between local and global addresses.
- Dynamic address translation (dynamic NAT)—Maps unregistered IP addresses to registered IP addresses from a pool of registered IP addresses.
- Overloading—Maps multiple unregistered IP addresses to a single registered IP address (many to one) by using different ports. This method is also known as Port Address Translation (PAT). Thousands of users can be connected to the Internet by using only one real global IP address through overloading.

NAT Inside and Outside Addresses

The term *inside* in a Network Address Translation (NAT) context refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network have addresses in one space (known as the *local* address space). These hosts appear to those users outside the network as being in another space (known as the *global* address space).

Similarly, the term *outside* refers to those networks to which the stub network connects, and which are not under the control of an organization. Also, hosts in outside networks can be subject to translation, and can thus have local and global addresses. NAT uses the following definitions:

- Inside local address—An IP address that is assigned to a host on the inside network. The address that the Network Information Center (NIC) or service provider assigns is probably not a legitimate IP address.
- Inside global address—A legitimate IP address assigned by the NIC or service provider that represents one or more inside local IP addresses to the outside world.
- Outside local address—The IP address of an outside host as it appears to the inside network. Not necessarily a legitimate address, it is allocated from the address space that is routable on the inside.
- Outside global address—The IP address that is assigned to a host on the outside network by the owner of the host. The address is allocated from a globally routable address or network space.

NAT supports the following VRFs:

Table 117: VRF NAT Support

NAT Inside Interface	NAT Outside Interface	Condition
Global VRF (also referred to as a non-VRF interface)	Global VRF (also referred to as a non-VRF interface)	Normal
VRF X	Global VRF (also referred to as a non-VRF interface)	When NAT is not configured for Match-in-VRF support. For more details, see the *Match-in-VRF Support for NAT* chapter.
VRF X	VRF X	When both inside and outside interfaces are in the same VRF, and NAT is configured with Match-in-VRF support.

This section describes the following topics:

- Inside Source Address Translation, on page 1008
- Overloading of Inside Global Addresses, on page 1010

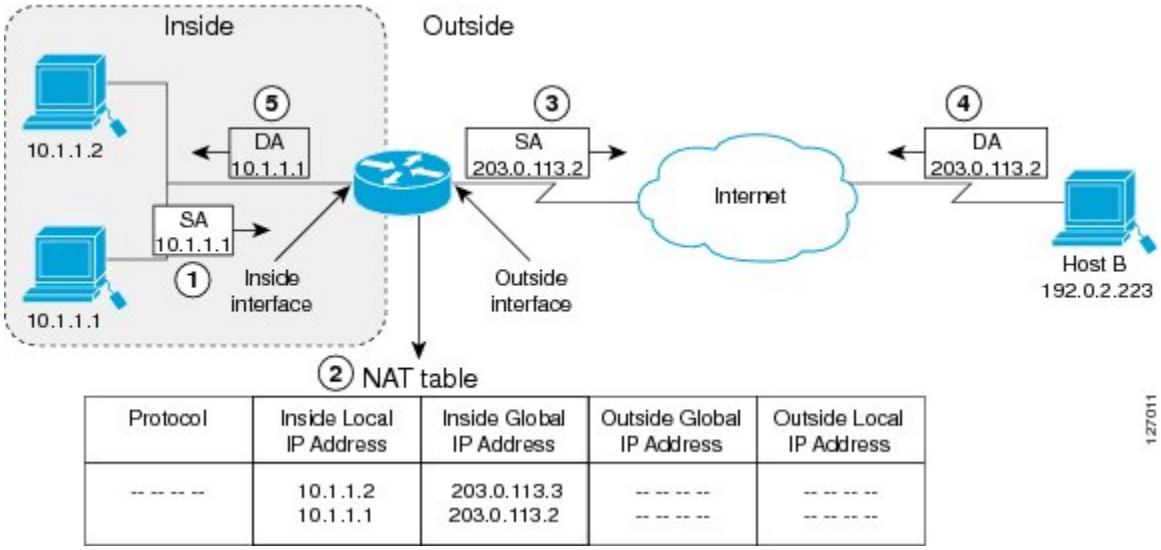
Inside Source Address Translation

You can translate IP addresses into globally unique IP addresses when communicating outside of your network. You can configure inside source address translation of static or dynamic NAT as follows:

- *Static translation* establishes a one-to-one mapping between the inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.
- *Dynamic translation* establishes a mapping between an inside local address and a pool of global addresses.

The following figure illustrates a device that is translating a source address inside a network to a source address outside the network.

Figure 80: NAT Inside Source Translation



The following process describes the inside source address translation, as shown in the preceding figure:

1. The user at host 10.1.1.1 opens a connection to Host B in the outside network.
2. The first packet that the device receives from host 10.1.1.1 causes the device to check its Network Address Translation (NAT) table. Based on the NAT configuration, the following scenarios are possible:
   - If a static translation entry is configured, the device goes to Step 3.
   - If no translation entry exists, the device determines that the source address (SA) 10.1.1.1 must be translated dynamically. The device selects a legal, global address from the dynamic address pool, and creates a translation entry in the NAT table. This kind of translation entry is called a *simple entry*.
3. The device replaces the inside local source address of host 10.1.1.1 with the global address of the translation entry and forwards the packet.
4. Host B receives the packet and responds to host 10.1.1.1 by using the inside global IP destination address (DA) 203.0.113.2.
5. When the device receives the packet with the inside global IP address, it performs a NAT table lookup by using the inside global address as a key. It then translates the address to the inside local address of host 10.1.1.1 and forwards the packet to host 10.1.1.1.

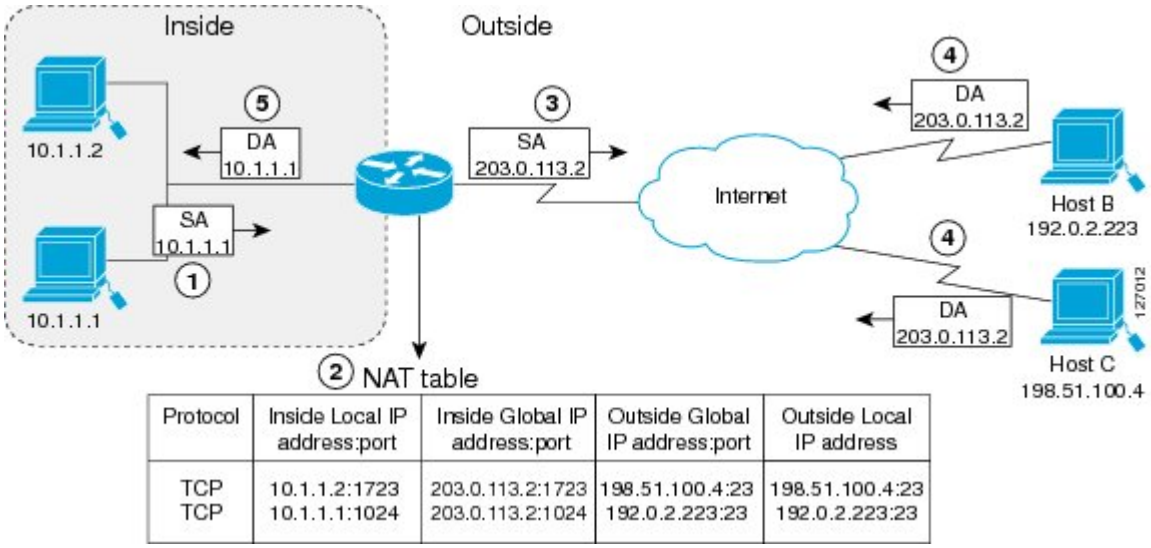
Host 10.1.1.1 receives the packet and continues the conversation. The device performs Steps 2 to 5 for each packet that it receives.

Overloading of Inside Global Addresses

You can conserve addresses in the inside global address pool by allowing a device to use one global address for many local addresses. This type of Network Address Translation (NAT) configuration is called overloading. When overloading is configured, the device maintains enough information from higher-level protocols (for example, TCP or UDP port numbers). This action translates the global address back to the correct local address. When multiple local addresses map to one global address, the TCP or UDP port numbers of each inside host distinguish between local addresses.

The following figure illustrates a NAT operation when an inside global address represents multiple inside local addresses. The TCP port numbers act as differentiators.

Figure 81: NAT Overloading Inside Global Addresses



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	10.1.1.2:1723	203.0.113.2:1723	198.51.100.4:23	198.51.100.4:23
TCP	10.1.1.1:1024	203.0.113.2:1024	192.0.2.223:23	192.0.2.223:23

The device performs the following process in the overloading of inside global addresses, as shown in the preceding figure. Both Host B and Host C believe that they are communicating with a single host at address 203.0.113.2. Whereas, they are actually communicating with different hosts; the port number is the differentiator. In fact, many inside hosts can share the inside global IP address by using many port numbers.

1. The user at host 10.1.1.1 opens a connection to Host B.
2. The first packet that the device receives from host 10.1.1.1 causes the device to check its NAT table. Based on your NAT configuration the following scenarios are possible:
   - If no translation entry exists, the device determines that IP address 10.1.1.1 must be translated, and translates inside local address 10.1.1.1 to a legal global address.
   - If overloading is enabled and another translation is active, the device reuses the global address from that translation and saves enough information. This saved information can be used to translate the global address back, as an entry in the NAT table. This type of translation entry is called an *extended entry*.
3. The device replaces inside local source address 10.1.1.1 with the selected global address and forwards the packet.
4. Host B receives the packet and responds to host 10.1.1.1 by using the inside global IP address 203.0.113.2.

5. When the device receives the packet with the inside global IP address, it performs a NAT table lookup by using a protocol, the inside global address and port, and the outside address and port as keys. It translates the address to the inside local address 10.1.1.1 and forwards the packet to host 10.1.1.1.

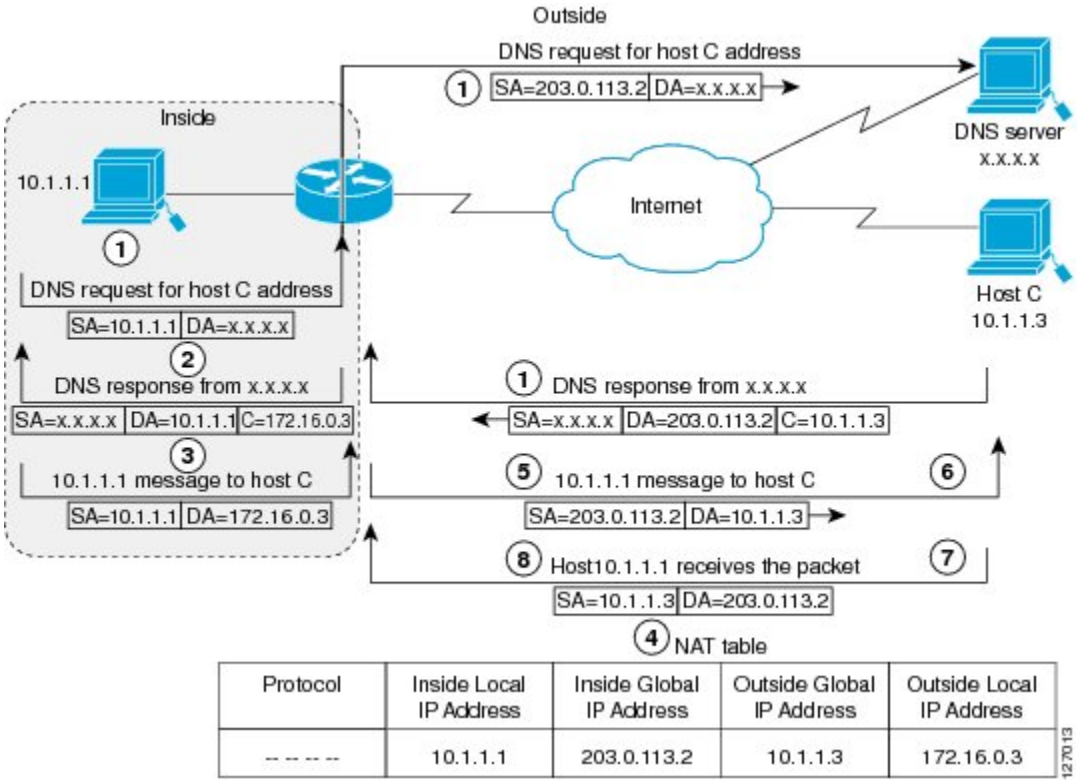
Host 10.1.1.1 receives the packet and continues the conversation. The device performs Steps 2 to 5 for each packet it receives.

Address Translation of Overlapping Networks

Use Network Address Translation (NAT) to translate IP addresses if the IP addresses that you use are not legal or officially assigned. Overlapping networks result when you assign an IP address to a device on your network. This device is already legally owned and assigned to a different device on the Internet or outside the network.

The following figure shows how NAT translates overlapping networks.

Figure 82: NAT Translating Overlapping Addresses



The following steps describe how a device translates overlapping addresses:

1. Host 10.1.1.1 opens a connection to Host C using a name, requesting a name-to-address lookup from a Domain Name System (DNS) server.

2. The device intercepts the DNS reply, and translates the returned address if there is an overlap. That is, the resulting legal address resides illegally in the inside network. To translate the return address, the device creates a simple translation entry. This entry maps the overlapping address, 10.1.1.3 to an address from a separately configured, outside the local address pool.

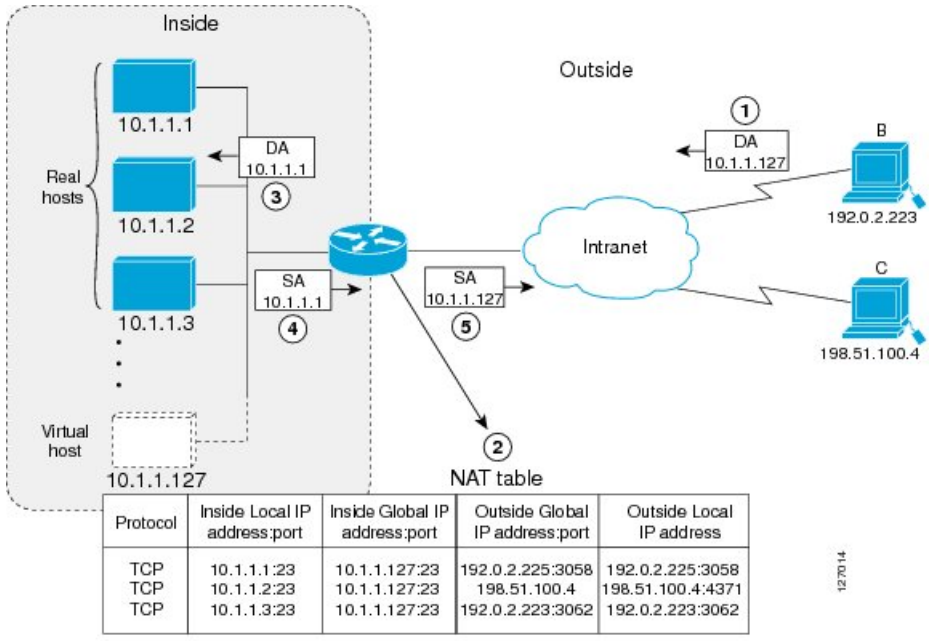
The device examines every DNS reply to ensure that the IP address is not in a stub network. If it is, the device translates the address as described in the following steps:

1. Host 10.1.1.1 opens a connection to 172.16.0.3.
2. The device sets up the translation mapping of the inside local and global addresses to each other. It also sets up the translation mapping of the outside global and local addresses to each other.
3. The device replaces the SA with the inside global address and replaces the DA with the outside global address.
4. Host C receives the packet and continues the conversation.
5. The device does a lookup, replaces the DA with the inside local address, and replaces the SA with the outside local address.
6. Host 10.1.1.1 receives the packet and the conversation continues using this translation process.

TCP Load Distribution for NAT

Your organization may have multiple hosts that must communicate with a heavily used host. By using Network Address Translation (NAT), you can establish a virtual host on the inside network that coordinates load sharing among real hosts. Destination addresses that match an access list are replaced with addresses from a rotary pool. Allocation is done on a round-robin basis and only when a new connection is opened from the outside to inside the network. Non-TCP traffic is passed untranslated (unless other translations are configured). The following figure illustrates how TCP load distribution works.

Figure 83: NAT TCP Load Distribution



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	10.1.1.1:23	10.1.1.127:23	192.0.2.225:3058	192.0.2.225:3058
TCP	10.1.1.2:23	10.1.1.127:23	198.51.100.4	198.51.100.4:4371
TCP	10.1.1.3:23	10.1.1.127:23	192.0.2.223:3062	192.0.2.223:3062

A device performs the following process when translating rotary addresses:

1. Host B (192.0.2.223) opens a connection to a virtual host at 10.1.1.127.

2. The device receives the connection request and creates a new translation, allocating the next real host (10.1.1.1) for the inside local IP address.
3. The device replaces the destination address with the selected real host address and forwards the packet.
4. Host 10.1.1.1 receives the packet and responds.
5. The device receives the packet and performs a NAT table lookup by using the inside local address and port number. It also does a NAT table lookup by using the outside address and port number as keys. The device then translates the source address to the address of the virtual host and forwards the packet.
6. The device will allocate IP address 10.1.1.2 as the inside local address for the next connection request.

Static IP Address Support

A public wireless LAN provides users of mobile computing devices with wireless connections to a public network, such as the Internet.

To support users who are configured with a static IP address, the NAT Static IP Address Support feature extends the capabilities of public wireless LAN providers. By configuring a device to support users with a static IP address, public wireless LAN providers extend their services to a greater number of users.

Users with static IP addresses can use services of the public wireless LAN provider without changing their IP address. NAT entries are created for static IP clients and a routable address is provided.

RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. Communication between a network access server (NAS) and a RADIUS server is based on UDP. Generally, the RADIUS protocol is considered a connectionless service. RADIUS-enabled devices handle issues that are related to a server availability, retransmission, and timeouts rather than the transmission protocol.

The RADIUS client is typically a NAS, and the RADIUS server is usually a daemon process running on a UNIX or Windows NT machine. The client passes user information to designated RADIUS servers and acts on the response that is returned. To deliver service to the user, RADIUS servers receive a user connection request, authenticate the user, and then return the configuration information necessary for the client. A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

Denial-of-Service Attacks

A denial-of-service (DoS) attack typically involves misuse of standard protocols or connection processes. The intent of DoS attack is to overload and disable a target, such as a device or web server. DoS attacks can come from a malicious user or from a computer that is infected with a virus or worm. Distributed DoS attack is an attack that comes from many different sources at once. This attack can be when a virus or worm has infected many computers. Such distributed DoS attacks can spread rapidly and involve thousands of systems.

Viruses and Worms That Target NAT

Viruses and worms are malicious programs that are designed to attack computers and networking equipment. Although viruses are typically embedded in discrete applications and run only when executed, worms self-propagate and can quickly spread by their own. Although a specific virus or worm may not expressly

target NAT, it may use NAT resources to propagate itself. The Rate Limiting NAT Translation feature can be used to limit the impact of viruses and worms. These viruses and worms originate from specific hosts, access control lists, and VPN routing and forwarding (VRF) instances.

How to Configure NAT for IP Address Conservation

The tasks that are described in this section configure NAT for IP address conservation. Ensure that you configure at least one of the tasks that are described in this section. Based on your configuration, you may need to configure more than one task.

Configuring Inside Source Addresses

Inside source addresses, can be configured for static or dynamic translations. Based on your requirements, you can configure either static or dynamic translations.

Configuring Static Translation of Inside Source Addresses

Configure static translation of the inside source addresses to allow one-to-one mapping between an inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.

Note Configure different IP addresses for an interface on which NAT is configured and for inside addresses that are configured by using the **ip nat inside source static** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask* [**secondary**]
10. **ip nat outside**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 10.10.10.1 172.16.131.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode. **Note** Conditional translation is not supported with **ip nat outside source route-map** configuration.

Configuring Dynamic Translation of Inside Source Addresses

Dynamic translation establishes a mapping between an inside local address and a pool of global addresses. Dynamic translation is useful when multiple users on a private network must access the Internet. The dynamically configured pool IP address may be used as needed. It is released for use by other users when access to the Internet is no longer required.

Note When inside global or outside local addresses belong to a directly connected subnet on a NAT device, the device adds IP aliases for them. This action enables it to answer Address Resolution Protocol (ARP) requests. However, a situation can arise where the device answers packets that are not destined for it, possibly causing a security issue. This security issue can happen when an incoming Internet Control Message Protocol (ICMP) packet or a UDP packet that is destined for one of the aliased addresses does not have a corresponding NAT translation in the NAT table. Also, the device itself runs a corresponding service, for example, Network Time Protocol (NTP). Such a situation can cause minor security risks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:**	Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
	`Device(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28`	
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Establishes dynamic source translation, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters an interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters an interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the Same Global Address for Static NAT and PAT

You can configure the same global address for the static NAT and PAT. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.

Note This is not supported with **ip nat inside source static** configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat outside source static** *outside global-ip outside local-ip*
4. **ip nat outside source static** {**tcp | udp**} *outside global-ip global-port outside local-ip local-port* **extendable**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat outside source static** *outside global-ip outside local-ip* **Example:** `Device(config)# ip nat outside source static 10.21.0.202 12.182.174.202`	Establishes static translation between an outside local address and an outside global address.
Step 4	**ip nat outside source static** {**tcp	udp**} *outside global-ip global-port outside local-ip local-port* **extendable** **Example:** `Router(config)# ip nat outside source static tcp 10.21.14.49 22512 12.182.174.202 5009 extendable`	• Establishes static translation between an outside global address and inside local address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Using NAT to Allow Internal Users Access to the Internet

Perform this task to allow your internal users access to the Internet and conserve addresses in the inside global address pool using overloading of global addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *name* **overload**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.224`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • The access list must permit only those addresses that are to be translated. (Remember that there is an implicit "deny all" at the end of each access list.) Use of an access list that is too permissive can lead to unpredictable results.

	Command or Action	Purpose
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **overload** **Example:** `Device(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.29 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Address Translation Timeouts

You can configure address translation timeouts that is based on your NAT configuration.

By default, dynamic address translations time out after a period of remaining idle. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours. Use the **ip nat translation timeout** command to change the timeout value for dynamic address translations.

You can use the **ip nat translation max-entries** command to change the default global NAT translation limit.

Note On Catalyst 6500 Series Switches, when the NAT translation is done in the hardware, timers are reset every 100 seconds or once the set timeout value is reached.

Changing the Translation Timeout

By default, dynamic address translations time out after some period of remaining idle. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours. Configure the **ip nat translation timeout** *seconds* command to change the timeout value for dynamic address translations that do not use overloading.

Changing the Timeouts When Overloading Is Configured

If you have configured overloading, you can control the translation entry timeout, because each translation entry contains more context about the traffic using it.

Based on your configuration, you can change the timeouts that are described in this section. If you must quickly free your global IP address for a dynamic configuration, configure a shorter timeout than the default timeout. You can do it by using the **ip nat translation timeout** command. However, the configured timeout is longer than the other timeouts configured using commands specified in the following task. If a finish (FIN) packet does not close a TCP session properly from both sides or during a reset, change the default TCP timeout. You can do it by using the **ip nat translation tcp-timeout** command.

When you change the default timeout using the **ip nat translation** *timeout* command, the timeout that you configure overrides the default TCP and UDP timeout values, unless you explicitly configure the TCP timeout value (using the **ip nat translation tcp-timeout** *seconds* command) or the UDP timeout value (using the **ip nat translation udp-timeout** *seconds* command).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat translation** *seconds*
4. **ip nat translation udp-timeout** *seconds*
5. **ip nat translation dns-timeout** *seconds*
6. **ip nat translation tcp-timeout** *seconds*
7. **ip nat translation finrst-timeout** *seconds*
8. **ip nat translation icmp-timeout** *seconds*
9. **ip nat translation syn-timeout** *seconds*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat translation** *seconds* **Example:** `Device(config)# ip nat translation 300`	(Optional) Changes the amount of time after which NAT translations time out. • The default timeout is 24 hours, and it applies to the aging time for half-entries. • The timeout configured using this command overrides the default TCP and UDP timeout values, unless explicitly configured.
Step 4	**ip nat translation udp-timeout** *seconds* **Example:** `Device(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value. • The default is 300 seconds. This default value only applies if the general IP NAT translation timeout value (using the **ip nat translation** *seconds* command) is not configured.
Step 5	**ip nat translation dns-timeout** *seconds* **Example:** `Device(config)# ip nat translation dns-timeout 45`	(Optional) Changes the Domain Name System (DNS) timeout value.
Step 6	**ip nat translation tcp-timeout** *seconds* **Example:** `Device(config)# ip nat translation tcp-timeout 2500`	(Optional) Changes the TCP timeout value. • The default is 7440 seconds. This default value only applies if the general IP NAT translation timeout value (using the **ip nat translation** *seconds* command) is not configured.
Step 7	**ip nat translation finrst-timeout** *seconds* **Example:** `Device(config)# ip nat translation finrst-timeout 45`	(Optional) Changes the finish and reset timeout value. • **finrst-timeout**—The aging time after a TCP session receives both finish-in (FIN-IN) and finish-out (FIN-OUT) requests or after the reset of a TCP session.
Step 8	**ip nat translation icmp-timeout** *seconds* **Example:** `Device(config)# ip nat translation icmp-timeout 45`	(Optional) Changes the ICMP timeout value.

	Command or Action	Purpose
Step 9	**ip nat translation syn-timeout** *seconds* **Example:** `Device(config)# ip nat translation syn-timeout 45`	(Optional) Changes the synchronous (SYN) timeout value. • The synchronous timeout or the aging time is used only when a SYN request is received on a TCP session. When a synchronous acknowledgment (SYNACK) request is received, the timeout changes to TCP timeout.
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Allowing Overlapping Networks to Communicate Using NAT

Tasks in this section are grouped because they perform the same action. However, the tasks are executed differently depending on the type of translation that is implemented—static or dynamic. Perform the task that applies to the translation type that you have implemented.

This section contains the following tasks:

- Configuring Static Translation of Overlapping Networks
- Configuring Dynamic Translation of Overlapping Networks
- What to Do Next

Configuring Static Translation of Overlapping Networks

Configure static translation of overlapping networks that are based on the following requirements:

- If your IP addresses in the stub network are legitimate IP addresses belonging to another network.
- If you want to communicate with those hosts or routers by using static translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 192.168.121.33 10.2.2.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.

	Command or Action	Purpose
Step 11	**end** **Example:** `Device(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

What to Do Next

When you have completed the required configuration, go to the "Monitoring and Maintaining NAT" module.

Configuring Server TCP Load Balancing

Perform this task to configure a server TCP load balancing by way of destination address rotary translation. The commands that are specified in the task allow you to map one virtual host with many real hosts. Each new TCP session opened with the virtual host is translated into a session with a different real host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary**
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside destination-list** *access-list-number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary** **Example:**	Defines a pool of addresses containing the addresses of the real hosts.

	Command or Action	Purpose
	`Device(config)# ip nat pool real-hosts 192.168.201.2 192.168.201.5 prefix-length 28 type rotary`	
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines an access list permitting the address of the virtual host.
Step 5	**ip nat inside destination-list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside destination-list 2 pool real-hosts`	Establishes dynamic inside destination translation, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface serial 0`	Specifies a different interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.15.129 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.

	Command or Action	Purpose
Step 13	**end** **Example:** `Device(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Enabling Route Maps on Inside Interfaces

Before you begin

All route maps required for use with this task must be configured before you begin the configuration task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**]| **static** *local-ip global-ip* [**route-map** *map-name*]}
4. **exit**
5. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**]| **static** *local-ip global-ip* [**route-map** *map-name*]} **Example:** `Device(config)# ip nat inside source static 192.168.201.6 192.168.201.21 route-map isp2`	Enables route mapping with static NAT configured on the NAT inside interface.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT.

Enabling NAT Route Maps Outside-to-Inside Support

The NAT Route Maps Outside-to-Inside Support feature enables you to configure a Network Address Translation (NAT) route map configuration. It allows IP sessions to be initiated from the outside to the inside. Perform this task to enable the NAT Route Maps Outside-to-Inside Support feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
4. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
5. **ip nat inside source route-map** *name* **pool** *name* [**reversible**]
6. **ip nat inside source route-map** *name* **pool** *name* [**reversible**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 4	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool POOL-B 192.168.201.7 192.168.201.9 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 5	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Device(config)# ip nat inside source route-map MAP-A pool POOL-A reversible`	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.
Step 6	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Device(config)# ip nat inside source route-map MAP-B pool POOL-B reversible`	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring NAT of External IP Addresses Only

When you configure NAT of external IP addresses, NAT can be configured to ignore all embedded IP addresses for any application and traffic type. Traffic between a host and the traffic outside an enterprise's network flows through the internal network. A device that is configured for NAT translates the packet to an address that can be routed inside the internal network. If the intended destination is outside an enterprise's network, the packet gets translated back to an external address and is sent out.

Note When you configure the **ip nat outside source static** command to add static routes for outside local addresses, there is a delay in the translation of packets and packets are dropped. Packets are dropped because a shortcut is not created for the initial synchronization (SYN) packet when NAT is configured for static translation. To avoid dropped packets, configure either the **ip nat outside source static add-route** command or the **ip route** command.

Benefits of configuring NAT of external IP addresses only are:

- Allows an enterprise to use the Internet as its enterprise backbone network.
- Allows the use of network architecture that requires only the header translation.
- Gives the end client a usable IP address at the starting point. This address is the address that is used for IPsec connections and for traffic flows.
- Supports public and private network architecture with no specific route updates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* [**no-payload**]}
4. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]}
5. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]}
6. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** *local-ip global-ip* [**no-payload**]}
7. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]}
8. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]}
9. **exit**

10. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static network 10.1.1.1 192.168.251.0/24 no-payload`	Disables the network packet translation on the inside host device.
Step 4	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static tcp 10.1.1.1 2000 192.168.1.1 2000 no-payload`	Disables port packet translation on the inside host device.
Step 5	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static 10.1.1.1 192.168.1.1 no-payload`	Disables packet translation on the inside host device.
Step 6	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** *local-ip global-ip* [**no-payload**]} **Example:** `Device(config)# ip nat outside source static 10.1.1.1 192.168.1.1 no-payload`	Disables packet translation on the outside host device.
Step 7	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:**	Disables port packet translation on the outside host device.

	Command or Action	Purpose
	`Device(config)# ip nat outside source static tcp 10.1.1.1 20000 192.168.1.1 20000 no-payload`	
Step 8	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Device(config)# ip nat outside source static network 10.1.1.1 192.168.251.0/24 no-payload`	Disables network packet translation on the outside host device.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	Displays active NAT.

Configuring the NAT Default Inside Server Feature

The NAT Default Inside Server feature helps forward packets from the outside to a specified inside local address. Traffic that does not match any existing dynamic translations or static port translations are redirected, and packets are not dropped.

Dynamic mapping and interface overload can be configured for gaming devices. For online games, outside traffic comes on a different UDP port. If a packet is destined for an interface from outside an enterprise's network, and there is no match in the NAT table for fully extended entry or static port entry, the packet is forwarded to the gaming device using a simple static entry.

Note

- You can use this feature to configure gaming devices with an IP address different from the IP address of the PC. To avoid unwanted traffic or DoS attacks, use access lists.
- For traffic going from the PC to the outside, it is better to use a route map so that extended entries are created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip* **interface** *type number*
4. **ip nat inside source static tcp** *local-ip local-port* **interface** *global-port*
5. **exit**
6. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip* **interface** *type number* **Example:** `Device(config)# ip nat inside source static 10.1.1.1 interface Ethernet 1/1`	Enables static NAT on the interface.
Step 4	**ip nat inside source static tcp** *local-ip local-port* **interface** *global-port* **Example:** `Device(config)# ip nat inside source static tcp 10.1.1.1 23 interface 23`	(Optional) Enables the use of telnet to the device from the outside.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT.

Reenabling RTSP on a NAT Router

The Real Time Streaming Protocol (RTSP) is a client/server multimedia presentation control protocol that supports multimedia application delivery. Some of the applications that use RTSP include Windows Media Services (WMS) by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks.

When the RTSP protocol passes through a NAT router, the embedded address and port must be translated for the connection to be successful. NAT uses Network Based Application Recognition (NBAR) architecture to parse the payload and translate the embedded information in the RTSP payload.

RTSP is enabled by default. Use the **ip nat service rtsp port** *port-number* command to reenable RTSP on a NAT router if this configuration has been disabled.

Configuring Support for Users with Static IP Addresses

Configuring support for users with static IP addresses enables those users to establish an IP session in a public wireless LAN environment.

Before you begin

Before configuring support for users with static IP addresses, you must first enable NAT on your router and configure a RADIUS server host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **ip nat allow-static-host**
7. **ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name*
8. **ip nat inside source list** *access-list-number* **pool** *name*
9. **access-list** *access-list-number* **deny ip** *source*
10. **end**
11. **show ip nat translations verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Configures an interface and enters an interface configuration mode.
Step 4	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**ip nat allow-static-host** **Example:** `Device(config)# ip nat allow-static-host`	Enables static IP address support. • Dynamic Address Resolution Protocol (ARP) learning will be disabled on this interface, and NAT will control the creation and deletion of ARP entries for the static IP host.

	Command or Action	Purpose
Step 7	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name* **Example:** `Device(config)# ip nat pool pool1 172.16.0.0 172.16.0.254 netmask 255.255.255.0 accounting WLAN-ACCT`	Specifies an existing RADIUS profile name to be used for authentication of the static IP host.
Step 8	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Specifies the access list and pool to be used for static IP support. • The specified access list must permit all traffic.
Step 9	**access-list** *access-list-number* **deny ip** *source* **Example:** `Device(config)# access-list 1 deny ip 192.168.196.51`	Removes the traffic of the device from NAT. • The *source* argument is the IP address of the device that supports the NAT Static IP Support feature.
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**show ip nat translations verbose** **Example:** `Device# show ip nat translations verbose`	(Optional) Displays active NAT translations and additional information for each translation table entry, including how long ago the entry was created and used.

Examples

The following is sample output from the **show ip nat translations verbose** command:

```
Device# show ip nat translations verbose

--- 172.16.0.0 10.1.1.1           ---                 ---
create 00:05:59, use 00:03:39, left 23:56:20, Map-Id(In): 1, flags: none wlan-flags: Secure
 ARP added, Accounting Start sent Mac-Address:0010.7bc2.9ff6 Input-IDB:Ethernet1/2, use_count:
 0, entry-id:7, lc_entries: 0
```

Configuring the Rate Limiting NAT Translation Feature

SUMMARY STEPS

1. **enable**
2. **show ip nat translations**
3. **configure terminal**
4. **ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*}
5. **end**

6. **show ip nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip nat translations** **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT. • A specific host, access control list, or VRF instance generating an unexpectedly high number of NAT requests may be the source of a malicious virus or worm attack.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*} **Example:** `Device(config)# ip nat translation max-entries 300`	Configures the maximum number of NAT entries that are allowed from the specified source. • The maximum number of allowed NAT entries is 2147483647, although a typical range for a NAT rate limit is 100 to 300 entries. • When you configure a NAT rate limit for all VRF instances, each VRF instance is limited to the maximum number of NAT entries that you specify. • When you configure a NAT rate limit for a specific VRF instance, you can specify a maximum number of NAT entries for the named VRF instance that is greater than or less than that allowed for all VRF instances.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat statistics** **Example:** `Device# show ip nat statistics`	(Optional) Displays current NAT usage information, including NAT rate limit settings. • After setting a NAT rate limit, use the **show ip nat statistics** command to verify the current NAT rate limit settings.

	Command or Action	Purpose
		Note The CEF counters associated with the output of the **show ip nat statistics** command signify the number of packets that are translated and forwarded in the SW plane. Packets that require translation are punted to the SW plane in the absence of the corresponding NF shortcuts in the HW plane. This enables SW plane to carry out the translation and program the corresponding NF shortcuts in the HW in order to facilitate the HW translation for subsequent packets that match the given flow. A route-map based NAT rule does not maintain Half Entry mappings and this implies that every new packet flow that matches the given rule is directed to the SW plane for translation and forwarding. Such packets undergo translation in the SW plane. This in turn results in the increment of the afore mentioned CEF counters. This is an expected behavior when you employ a route-map-based NAT configuration. However, note that these packets that undergo translation in the SW result in the corresponding full flow NF shortcuts to be programmed in the HW. This is to facilitate the HW translation of subsequent packets that match the given flow.

Configuring Bypass NAT Functionality

The Bypass NAT functionality feature reduces the TCAM size by resolving the deny jump issue. To enable the Bypass NAT functionality feature, you must:

- Create a NAT bypass pool by using a reserved loopback address (127.0.0.1).
- Create a new NAT mapping containing a new ACL with all existing deny statements that are converted to permit statements.

You can enable the Bypass NAT functionality by creating new NAT mapping with new ACL mapped to a bypass pool.

To configure the bypass-pool with 127.0.0.1 as reserved loopback address:

```
enable
configure terminal
access-list 60 permit 25.33.0.0 0.0.255.255
ip nat pool bypass-pool 127.0.0.1 127.0.0.1 prefix-length 24
ip nat inside source list 60 pool bypass-pool
end
```

To convert existing configuration with deny statements:

```
enable
configure terminal
ip access list extended nat-acl
```

```
deny ip host 10.10.10.10 host 10.77.64.17
permit ip any 10.77.64.0 0.0.15.255
ip nat inside source list nat-acl pool nat-pool
end
```

New converted configuration using bypass pool with permit statements:

```
enable
configure terminal
ip nat pool bypass-pool 127.0.0.1 127.0.0.1 prefix-length 24
ip access list extended nat-bypass-acl
permit ip host 10.10.10.10 host 10.77.64.17
ip nat inside source list nat-bypass-acl pool bypass-pool
ip access list extended nat-acl
permit ip any 10.77.64.0 0.0.15.255
ip nat inside source list nat-acl pool nat-pool
end
```

Configuration Examples for Configuring NAT for IP Address Conservation

Example: Configuring Static Translation of Inside Source Addresses

The following example shows how inside hosts addressed from the 10.114.11.0 network are translated to the globally unique 172.31.233.208/28 network. Further, packets from outside hosts that are addressed from the 10.114.11.0 network (the true 10.114.11.0 network) are translated to appear from the 10.0.1.0/24 network.

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface gigabitethernet 1/1/1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
!
access-list 1 permit 10.114.11.0 0.0.0.255
```

The following example shows NAT configured on the provider edge (PE) device with a static route to the shared service for the vrf1 and vrf2 VPNs. NAT is configured as inside source static one-to-one translation.

```
ip nat pool outside 10.4.4.1 10.4.4.254 netmask 255.255.255.0
ip nat outside source list 1 pool mypool
access-list 1 permit 172.16.18.0 0.0.0.255
ip nat inside source static 192.168.121.33 10.2.2.1 vrf vrf1
ip nat inside source static 192.168.121.33.10.2.2.2 vrf vrf2
```

Example: Configuring Dynamic Translation of Inside Source Addresses

The following example shows how inside hosts addressed from either the 192.168.1.0 or the 192.168.2.0 network are translated to the globally unique 172.31.233.208/28 network:

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 9
ip nat inside source list 1 pool net-208
!
interface gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface gigabitethernet 1/1/1
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 permit 192.168.2.0 0.0.0.255
!
```

The following example shows how only traffic local to the provider edge (PE) device running NAT is translated:

```
ip nat inside source list 1 interface gigabitethernet 0/0/0 vrf vrf1 overload
ip nat inside source list 1 interface gigabitethernet 0/0/0 vrf vrf2 overload
!
ip route vrf vrf1 0.0.0.0 0.0.0.0 192.168.1.1
ip route vrf vrf2 0.0.0.0 0.0.0.0 192.168.1.1
!
access-list 1 permit 10.1.1.1.0 0.0.0.255
!
ip nat inside source list 1 interface gigabitethernet 1/1/1 vrf vrf1 overload
ip nat inside source list 1 interface gigabitethernet 1/1/1 vrf vrf2 overload
!
ip route vrf vrf1 0.0.0.0 0.0.0.0 172.16.1.1 global
ip route vrf vrf2 0.0.0.0 0.0.0.0 172.16.1.1 global
access-list 1 permit 10.1.1.0 0.0.0.255
!
```

Example: Using NAT to Allow Internal Users Access to the Internet

The following example shows how to create a pool of addresses that is named net-208. The pool contains addresses from 172.31.233.208 to 172.31.233.233. Access list 1 allows packets with SA from 192.168.1.0 to 192.168.1.255. If no translation exists, packets matching access list 1 is translated to an address from the pool. The router allows multiple local addresses (192.168.1.0 to 192.168.1.255) to use the same global address. The router retains port numbers to differentiate the connections.

```
ip nat pool net-208 172.31.233.208 172.31.233.233 netmask 255.255.255.240
access-list 1 permit 192.168.1.0 0.0.0.255
ip nat inside source list 1 pool net-208 overload
interface gigabitethernet 1/1/1
 ip address 192.168.201.1 255.255.255.240
 ip nat inside
!
interface gigabitethernet 0/0/0
 ip address 192.168.201.29 255.255.255.240
 ip nat outside
!
```

Example: Allowing Overlapping Networks to Communicate Using NAT

Example: Configuring Static Translation of Overlapping Networks

```
ip nat inside source static 192.168.121.33 10.2.2.1
!
interface gigabitethernet 1/1/1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
!
interface gigabitethernet 0/0/0
 ip address 172.16.232.182 255.255.255.240
 ip nat outside
!
```

Example: Configuring Dynamic Translation of Overlapping Networks

In the following example, the addresses in the local network are being used legitimately by someone else on the Internet. An extra translation is required to access the external network. The pool net-10 is a pool of outside local IP addresses. The **ip nat outside source list 1 pool** *net-10* command translates the addresses of hosts from the outside overlapping network to addresses in that pool.

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
access-list 1 permit 10.114.11.0 0.0.0.255
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface gigabitethernet 1/1/1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
!
interface gigabitethernet 0/0/0
 ip address 172.16.232.182 255.255.255.240
 ip nat outside
!
```

Example: Configuring Server TCP Load Balancing

In the following example, the goal is to define a virtual address, connections to which are distributed among a set of real hosts. The pool defines addresses of real hosts. The access list defines the virtual address. If a translation does not exist, TCP packets from serial interface 0 (the outside interface), whose destination matches the access list, are translated to an address from the pool.

```
ip nat pool real-hosts 192.168.15.2 192.168.15.15 prefix-length 28 type rotary
access-list 2 permit 192.168.15.1
ip nat inside destination list 2 pool real-hosts
interface gigabitethernet 0/0/0
 ip address 192.168.15.129 255.255.255.240
 ip nat inside
!
interface serial 0
 ip address 192.168.15.17 255.255.255.240
```

```
 ip nat outside
!
```

Example: Enabling Route Maps on Inside Interfaces

```
ip nat inside source static 192.168.201.6 192.168.201.21
!
```

Example: Enabling NAT Route Maps Outside-to-Inside Support

The following example shows how to configure a route map A and route map B to allow outside-to-inside translation for a destination-based Network Address Translation (NAT):

```
ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128
ip nat pool POOL-B 192.168.201.7 192.168.201.9 netmask 255.255.255.128
ip nat inside source route-map MAP-A pool POOL-A reversible
ip nat inside source route-map MAP-B pool POOL-B reversible
```

Example: Configuring NAT of External IP Addresses Only

```
ip nat inside source static network 10.1.1.1 192.168.2510/24 no-payload
ip nat inside source static tcp 10.1.1.1 2000 192.168.1.1 2000 no-payload
ip nat inside source static 10.1.1.1 192.168.1.1 no-payload
ip nat outside source static 10.1.1. 192.168.1.1 no-payload
ip nat outside source static tcp 10.1.1.1 20000 192.168.1.1 20000 no-payload
ip nat outside source static network 10.1.1.1 192.168.251.0/24 no-payload
```

Example: Configuring Support for Users with Static IP Addresses

```
interface gigabitethernet 1/1/1
 ip nat inside
!
ip nat allow-static-host
ip nat pool pool1 172.16.0.0 172.16.0.254 netmask 255.255.255.0 accounting WLAN-ACCT
ip nat inside source list 1 pool net-208
access-list 1 deny ip 192.168.196.51
```

Example: Configuring NAT Static IP Support

The following example shows how to enable static IP address support for the device at 192.168.196.51:

```
interface gigabitethernet 1/1/1
 ip nat inside
!
ip nat allow-static-host
ip nat pool net-208 172.16.1.1 172.16.1.10 netmask 255.255.255.0 accounting WLAN-ACCT
ip nat inside source list 1 pool net-208
access-list 1 deny ip 192.168.196.51
```

Example: Creating a RADIUS Profile for NAT Static IP Support

The following example shows how to create a RADIUS profile for use with the NAT Static IP Support feature:

```
aaa new-model
!
aaa group server radius WLAN-RADIUS
 server 172.16.88.1 auth-port 1645 acct-port 1645
 server 172.16.88.1 auth-port 1645 acct-port 1646
!
aaa accounting network WLAN-ACCT start-stop group WLAN-RADIUS
aaa session-id common
ip radius source-interface gigabitethernet3/0
radius-server host 172.31.88.1 auth-port 1645 acct-port 1646
radius-server key cisco
```

Example: Configuring the Rate Limiting NAT Translation Feature

The following example shows how to limit the maximum number of allowed NAT entries to 300:

```
ip nat translation max-entries 300
```

The following example shows how to limit the VRF instance named “vrf1” to 150 NAT entries:

```
ip nat translation max-entries vrf vrf1 150
```

The following example shows how to limit each VRF instance to 200 NAT entries:

```
ip nat translation max-entries all-vrf 200
```

The following example shows how to limit the VRF instance, “vrf2” to 225 NAT entries, but limit all other VRF instances to 100 NAT entries each:

```
ip nat translation max-entries all-vrf 100
ip nat translation max-entries vrf vrf2 225
```

The following example shows how to limit the access control list named “vrf3” to 100 NAT entries:

```
ip nat translation max-entries list vrf3 100
```

The following example shows how to limit the host at IP address 10.0.0.1 to 300 NAT entries:

```
ip nat translation max-entries host 10.0.0.1 300
```

Example: Setting a Global NAT Rate Limit

The following example shows how to limit the maximum number of allowed NAT entries to 300:

```
ip nat translation max-entries 300
```

Example: Setting NAT Rate Limits for a Specific VRF Instance

The following example shows how to limit the VRF instance named “vrf1” to 150 NAT entries:

```
ip nat translation max-entries vrf vrf1 150
```

Example: Setting NAT Rate Limits for All VRF Instances

The following example shows how to limit each VRF instance to 200 NAT entries:

```
ip nat translation max-entries all-vrf 200
```

The following example shows how to limit the VRF instance, “vrf2” to 225 NAT entries, but limit all other VRF instances to 100 NAT entries each:

```
ip nat translation max-entries all-vrf 100
ip nat translation max-entries vrf vrf2 225
```

Example: Setting NAT Rate Limits for Access Control Lists

The following example shows how to limit the access control list named "vrf3" to 100 NAT entries:

```
ip nat translation max-entries list vrf3 100
```

Example: Setting NAT Rate Limits for an IP Address

The following example shows how to limit the host at IP address 10.0.0.1 to 300 NAT entries:

```
ip nat translation max-entries host 10.0.0.1 300
```

Where to Go Next

- To configure NAT for use with application-level gateways, see the "Using Application Level Gateways with NAT" module.
- To verify, monitor, and maintain NAT, see the "Monitoring and Maintaining NAT" module.
- To integrate NAT with Multiprotocol Label Switching (MPLS) VPNs, see the "Integrating NAT with MPLS VPNs" module.
- To configure NAT for high availability, see the "Configuring NAT for High Availability" module.

Additional References for Configuring NAT for IP Address Conservation

Related Documents

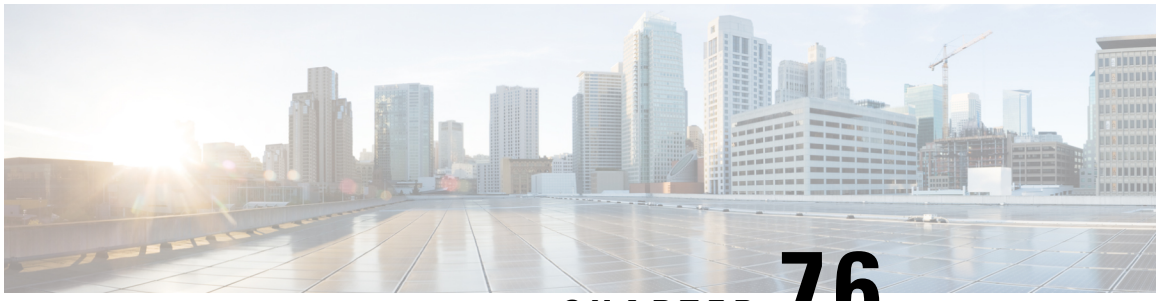
Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands: complete command syntax, command mode command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
Application-level gateways	*Using Application Level Gateways with NAT* module
IP access list sequence numbering	IP Access List Entry Sequence Numbering document
RADIUS attributes overview	*RADIUS Attributes Overview and RADIUS IETF Attributes* module

Standards and RFCs

Standard/RFC	Title
IETF Behave Draft NAT MIB	Definitions of Managed Objects for Network Address Translators (NAT) draft-ietf-behave-nat-mib-11
RFC 1597	Internet Assigned Numbers Authority
RFC 1631	The IP Network Address Translation (NAT)
RFC 1918	Address Allocation for Private Internets
RFC 2663	IP Network Address Translation (NAT) Terminology and Considerations
RFC 3022	Traditional IP Network Address Translation (Traditional NAT)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services. These services are the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support



CHAPTER 76

Using Application-Level Gateways with NAT

This module describes the basic tasks to configure an application-level gateway (ALG) with Network Address Translation (NAT). This module also provides information about the protocols that use ALGs for IP header translation.

NAT performs translation services on any TCP/UDP traffic that does not carry source and destination IP addresses in the application data stream. Protocols that do not carry the source and destination IP addresses include HTTP, TFTP, telnet, archie, finger, Network Time Protocol (NTP), Network File System (NFS), remote login (rlogin), remote shell (rsh) protocol, and remote copy (rcp).

Specific protocols that embed the IP address information within the payload require the support of an ALG. NAT requires a variety of ALGs to handle application data stream (Layer 7) protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels.

NAT supports virtual routing and forwarding (VRF) for protocols that have a supported ALG.

The Support for IPsec ESP Through NAT feature provides the ability to support multiple concurrent IPsec Encapsulating Security Payload (ESP) tunnels or connections through a NAT device configured in Overload or Port Address Translation (PAT) mode. You can use the **ip nat service dns-v6** command to control processing of IPv6 DNS packets by ALG

Prerequisites for Using Application Level Gateways with NAT

- Before performing the tasks in this module, you should be familiar with the concepts described in the “Configuring NAT for IP Address Conservation” module.
- All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “IP Access List Sequence Numbering” document.

- Before performing the tasks in this module, you should verify that the Session Initiation Protocol (SIP) and H.323 are not disabled. SIP and H.323 are enabled by default.

Restrictions for Using Application-Level Gateways with NAT

- Configuring EDM (end-point dependent mapping) using NAT is not supported on ALG.
- The H.323 functionality is deprecated in Cisco IOS 15.9(3)M release and this change can impact the NAT H.323 ALG functionality. The Cisco Technical Support team does not provide support for the ALG functionality issues related to the H.323 deprecation. If you are impacted by this change, it is recommended to use SIP as a migration path.

Information About Using Application-Level Gateways with NAT

IPsec

IPsec is a set of extensions to the IP protocol family in a framework of open standards for ensuring secure private communications over the Internet. Based on standards developed by the IETF, IPsec ensures confidentiality, integrity, and authenticity of data communications across the public network and provides cryptographic security services.

Secure tunnels between two peers, such as two routers, are provided and decisions are made as to which packets are considered sensitive and should be sent through these secure tunnels, and which parameters should be used to protect these sensitive packets by specifying characteristics of these tunnels. When the IPsec peer receives a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

IPsec using Encapsulating Security Payload (ESP) can pass through a router running NAT without any specific support from it as long as Network Address Port Translation (NAPT) or address overloading is not configured. You can enable IPsec packet processing using ESP with the **ip nat service ipsec-esp enable** command.

There are a number of factors to consider when attempting an IPsec VPN connection that traverses a NAPT device that represents multiple private internal IP addresses as a single public external IP address. Such factors include the capabilities of the VPN server and client, the capabilities of the NAPT device, and whether more than one simultaneous connection is attempted across the NAPT device.

There are two possible methods for configuring IPsec on a router with NAPT:

- Encapsulate IPsec in a Layer 4 protocol such as TCP or UDP. In this case, IPsec is *sneaking* through NAT. The NAT device is unaware of the encapsulation.
- Add IPsec-specific support to NAPT. IPsec works with NAT in this case as opposed to *sneaking* through NAT. The NAT Support for IPsec ESP-- Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a Cisco IOS router configured with NAPT.

We recommend that TCP and UDP be used when conducting IPsec sessions that traverse a NAPT device. However, not all VPN servers or clients support TCP or UDP.

SPI Matching

SPI matching is used to establish VPN connections between multiple pairs of destinations. NAT entries will immediately be placed in the translation table for endpoints matching the configured access list..

Benefits of Configuring NAT IPsec

- NAT enables customers to deploy private IP addresses within their network and translate the private IP addresses to public IP addresses when connecting to the Internet or when interconnecting with another corporate network.
- NAT support for the Session Initiation Protocol (SIP) adds the ability to deploy NAT on VoIP solutions based on SIP.
- With NAT ALGs, customers can control their IP address scheme and include complete support for H.323 v2 gatekeeper designs.
- ESP entries in the translation table are normally delayed from being transmitted until a reply is received from the destination. With predictable security parameter indexes (SPIs) and SPI matching, the delay can be eliminated because SPI entries are matched. Some third-party concentrators require both source ports and incoming ports to use port 500. Use the **ip nat service preserve-port** command to preserve the ports rather than changing them, which is required with regular NAT.

Voice and Multimedia over IP Networks

SIP is a protocol developed by the IETF Multiparty Multimedia Session Control (MMUSIC) Working Group. The Cisco SIP functionality equips Cisco routers to signal the setup of voice and multimedia calls over IP networks. SIP provides an alternative to H.323 within the VoIP internetworking software.

Session Description Protocol (SDP) is a protocol that describes multimedia sessions. SDP may be used in SIP message bodies to describe multimedia sessions used for creating and controlling multimedia sessions with two or more participants.

The NAT Support for SIP feature allows SIP embedded messages passing through a router configured with NAT to be translated and encoded back to the packet. An ALG is used with NAT to translate the SIP or SDP messages.

Note By default support for SIP is enabled on port 5060. Therefore, NAT-enabled devices interpret all packets on this port as SIP call messages. If other applications in the system use port 5060 to send packets, the NAT service may corrupt the packet as it attempts to interpret the packet as a SIP call message.

NAT Support of H.323 v2 RAS

Cisco IOS NAT supports all H.225 and H.245 message types, including those sent in the Registration, Admission, and Status (RAS) protocol. RAS provides a number of messages that are used by software clients and VoIP devices to register their location, request assistance in call setup, and control bandwidth. The RAS messages are directed toward an H.323 gatekeeper.

Some RAS messages include IP addressing information in the payload, typically meant to register a user with the gatekeeper or learn about another user already registered. If these messages are not known to NAT, they cannot be translated to an IP address that will be visible to the public.

In Cisco IOS Release 12.2(2)T and later releases, embedded IP addresses can be inspected for potential address translation. Prior to Cisco IOS Release 12.2(2)T, NAT did not support H.323 v2 RAS messages.

NAT Support for H.323 v3 and v4 in v2 Compatibility Mode

H.323 is an ITU-T specification for transmitting audio, video, and data across packet networks. NAT supports four versions of the H.323 protocols: Version 1, Version 2, Version 3, and Version 4. The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables NAT routers to support messages coded in H.323 Version 3 and Version 4 when these messages contain fields that are compatible with H.323 Version 2. This feature does not support H.323 capabilities introduced in H.323 Version 3 and Version 4, such as new message types or new fields that require address translation.

NAT H.245 Tunneling Support

The NAT H.245 Tunneling Support feature supports H.245 tunneling in H.323 ALGs. The H.245 tunneling supports H.245 tunnel messages that are needed to create a media channel setup.

For an H.323 call to take place, an H.225 connection on TCP port 1720 must be opened. When the H.225 connection is opened, the H.245 session is initiated and established. The H.323 connection can take place on a separate channel other than the H.225 or it can be done by using H.245 tunneling on the same H.225 channel whereby the H.245 messages are embedded in H.225 messages and sent on the previously established H.225 channel.

If the H.245 tunneled message is not understood by NAT, the media address or the port number is left untranslated by NAT, resulting in media traffic failure. The H.245 FastConnect procedures will not help if the H.245 tunneled message is not understood by NAT because FastConnect is terminated as soon as an H.245 tunneled message is sent.

NAT Support of Skinny Client Control Protocol

Cisco IP phones use the SCCP to connect with and register to Cisco CallManager.

To be able to configure Cisco IOS NAT between the IP phone and Cisco CallManager in a scalable environment, NAT needs to be able to detect the SCCP and understand the information passed within the messages. Messages flow back and forth that include IP address and port information used to identify other IP phone users with which a call can be placed.

The SCCP client to Cisco CallManager communication typically flows from inside to outside. Domain Name System (DNS) should be used to resolve the Cisco CallManager IP address connection when the Cisco CallManager is on the inside (behind the NAT device), or static NAT should be configured to reach the Cisco CallManager in the inside.

When an IP phone attempts to connect to the Cisco CallManager and it matches the configured NAT rules, NAT will translate the original source IP address and replace it with one from the configured pool. This new address will be reflected in the Cisco CallManager and be visible to other IP phone users.

NAT Support of SCCP Fragmentation

Skinny Client Control Protocol (SCCP) messages, also called Skinny control messages, are exchanged over TCP. If either the IP phone or the Cisco Unified CallManager is configured to have a TCP maximum segment size (MSS) lower than the Skinny control message payload, the Skinny control message is segmented across

multiple TCP segments. Prior to the introduction of this feature, Skinny control message exchanges used to fail during TCP segmentation because the NAT Skinny ALG was not able to reassemble Skinny control messages. The NAT SCCP Fragmentation Support feature adds support for TCP segments for the NAT Skinny ALG and fragmented payloads that requires an IP translation or a port translation is no longer dropped.

Skinny control messages can also be IP fragmented by using Virtual Fragmentation Reassembly (VFR).

In Cisco IOS Release 15.1(3)T and later releases, NAT works with SCCP phones Version 17 and higher.

NAT Segmentation with Layer 4 Forwarding

The NAT Segmentation with Layer 4 Forwarding feature is implemented for the H.323, Skinny Client Control Protocol (SCCP), and the TCP Domain Name System (DNS) protocol. NAT supports the processing of segmented H.323, SCCP, or TCP DNS messages that are split across multiple packets.

Layer 4 forwarding or TCP proxy is responsible for session handling that includes setting sequence numbers in order, acknowledging the numbers in a packet, resegmenting the translated packet if it is larger than the maximum segment size (MSS), and handling retransmissions in case of packet loss. Layer 4 forwarding also handles out-of-order packets and these packets are buffered and not dropped. Layer 4 forwarding buffers received packets and notifies the NAT ALG when an in-order packet is available, sends acknowledgments to end hosts for received packets, and sends translated packets that it receives from the NAT ALG back into the output packet path.

Restrictions

The NAT Segmentation with Layer 4 Forwarding feature does not work when:

- Firewalls are configured using the **ip inspect name** command. (Context-Based Access Control (CBAC) firewalls are not supported. Zone-based firewalls are supported.)
- H.323, SCCP, or TCP DNS messages are larger than 18 KB.
- Multiprotocol Label Switching (MPLS) is configured.
- NAT and the Cisco Unified CallManager are configured on the same device. In this case, a colocated solution in Call Manager Express is used.
- NAT Virtual Interface (NVI) is configured.
- Stateful Network Address Translation (SNAT) is enabled.

Note Effective January 31, 2014, Stateful NAT is not available in Cisco IOS software. For more information, see End-of-Sale and End-of-Life Announcement for the Cisco IOS Stateful Failover of Network Address Translation (SNAT).

- The **match-in-vrf** keyword is configured along with the **ip nat inside source** command for packet translation.
- The packets are IPv6 packets.

How to Configure Application-Level Gateways with NAT

Configuring IPsec Through NAT

Configuring IPsec ESP Through NAT

IPsec ESP Through NAT provides the ability to support multiple concurrent IPsec ESP tunnels or connections through a Cisco IOS NAT device configured in Overload or PAT mode.

Perform this task to configure IPsec ESP through NAT.

Note IPsec can be configured for any NAT configuration, not just static NAT configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat** [**inside** | **outside**] **source static** *local-ip global-ip* [**vrf** *vrf-name*]
4. **exit**
5. **show ip nat translations**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat** [**inside**	**outside**] **source static** *local-ip global-ip* [**vrf** *vrf-name*] **Example:** `Router(config)# ip nat inside source static 10.10.10.10 192.168.30.30`	Enables static NAT.
Step 4	**exit** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config)# exit`	
Step 5	**show ip nat translations** **Example:** `Router# show ip nat translations`	(Optional) Displays active NATs.

Enabling the Preserve Port

Note This task is required by certain VPN concentrators. Cisco VPN devices generally do not use this feature.

This task is used for IPsec traffic using port 500 for the source port. Perform this task to enable port 500 to be preserved for the source port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **IKE preserve-port**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service list** *access-list-number* **IKE preserve-port** **Example:** `Router(config)# ip nat service list 10 IKE preserve-port` **Note** When you configure the **ip nat service list** *list* **IKE preserve-port**, ensure that you define the access list for both in2out and out2in traffic.	Specifies IPsec traffic that matches the access list to preserve the port.

Enabling SPI Matching on the NAT Device

Note SPI matching is disabled by default.

Security parameter index (SPI) matching is used to establish VPN connections between multiple pairs of destinations. NAT entries are immediately placed in the translation table for endpoints matching the configured access list. SPI matching is available only for endpoints that choose SPIs according to the predictive algorithm implemented in Cisco IOS Release 12.2(15)T.

The generation of SPIs that are predictable and symmetric is enabled. SPI matching should be used in conjunction with NAT devices when multiple ESP connections across a NAT device are desired.

Before you begin

Cisco IOS software must be running on both the source router and the remote gateway enabling parallel processing.

Note SPI matching must be configured on the NAT device and both endpoint devices.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **ESP spi-match**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service list** *access-list-number* **ESP spi-match** **Example:** `Router(config)# ip nat service list 10 ESP spi-match`	Specifies an access list to enable SPI matching. • This example shows how to enter ESP traffic matching list 10 into the NAT table, making the assumption that both devices are Cisco devices and are configured to provide matchable SPIs.

Enabling SPI Matching on Endpoints

Before you begin

Cisco software must be running on both the source device and the remote gateway, enabling parallel processing.

Note Security parameter index (SPI) matching must be configured on the Network Address Translation (NAT) device and on both endpoint devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec nat-transparency spi-matching**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec nat-transparency spi-matching** **Example:** `Device(config)# crypto ipsec nat-transparency spi-matching`	Enables SPI matching on both endpoints.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Enabling MultiPart SDP Support for NAT

The MultiPart SDP Support for NAT feature provides support for the multipart Session Description Protocol (SDP) in a SIP ALG. MultiPart SDP support for NAT is disabled by default.

Note NAT translates only embedded IPv4 addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service allow-multipart**
4. **exit**
5. **show ip nat translations**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service allow-multipart** **Example:** `Device(config)# ip nat service allow-multipart`	Enables multipart SDP.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 5	**show ip nat translations** **Example:** `Device# show ip nat translations`	(Optional) Displays active NATs.

Configuring NAT Between an IP Phone and Cisco CallManager

This section describes configuring Cisco's Skinny Client Control Protocol (SCCP) for Cisco IP phone to Cisco CallManager communication. The task in this section configures NAT between an IP phone and Cisco CallManager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service skinny tcp port** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service skinny tcp port** *number* **Example:** `Router(config)# ip nat service skinny tcp port 20002`	Configures the skinny protocol on the specified TCP port.

Configuration Examples for Using Application-Level Gateways with NAT

Example: Specifying a Port for NAT Translation

```
ip nat service skinny tcp port 20002
```

Example: Enabling the Preserve Port

The following example shows how to configure TCP port 500 of the third-party concentrator. Access list 10 is configured.

```
ip nat service list 10 IKE preserve-port
access-list 10 permit 10.1.1.1
```

Example Enabling SPI Matching

The following example shows how to enable SPI matching. Access list 10 is configured:

```
ip nat service list 10 ESP spi-match
access-list 10 permit 10.1.1.1
```

Example: Enabling SPI Matching on Endpoints

```
crypto ipsec nat-transparency spi-matching
```

Example: Enabling MultiPart SDP Support for NAT

```
ip nat service allow-multipart
```

Example: Specifying a Port for NAT Translation

```
ip nat service skinny tcp port 20002
```

Where to Go Next

- To learn about NAT and configure NAT for IP address conservation, see the "Configuring NAT for IP Address Conservation" module.
- To verify monitor, and maintain NAT, see the "Monitoring and Maintaining NAT" module.
- To integrate NAT with MPLS VPNs, see the "Integrating NAT with MPLS VPNs" module.
- To configure NAT for high availability, see the "Configuring NAT for High Availability" module.

Additional References for Using Application-Level Gateways with NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
IP access list sequence numbering	*IP Access List Sequence Numbering*
NAT IP address conservation	*Configuring NAT for IP Address Conservation*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

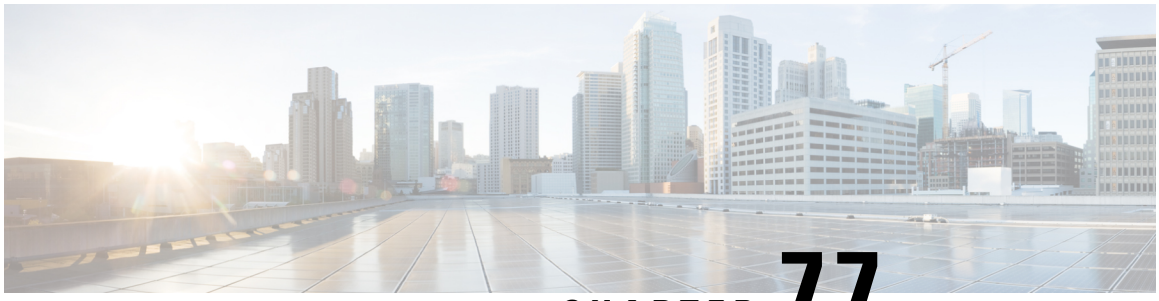
Feature Information for Using Application-Level Gateways with NAT

Table 118: Feature Information for Using Application-Level Gateways with NAT

Feature Name	Releases	Feature Configuration Information
ALG—H.323 v6 Support	Cisco IOS XE Release 3.6S	The ALG—H.323 v6 supports the parsing of H.323 v6 packets and the inspection and translation of IPv4 address information in H.323 messages.
ALG—SCCP Version 17 Support	Cisco IOS XE Release 3.5S	The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP Version 17 packets. Cisco Unified Communications Manager 7.0 and IP phones that use Cisco Unified Communications Manager 7.0 support only SCCP Version 17 messages. The SCCP Version 17 packets support IPv6 packets. The SCCP ALG supports the inspection and translation of IPv4 address information in SCCP messages.
NAT ALG—SIP REFER Method	Cisco IOS XE Release 3.2S	The NAT ALG—SIP REFER method feature supports two types of call transfers, unattended (blind) transfer and attended (consultative) transfer.
NAT ALG—SIP Trunking Support	Cisco IOS XE Release 3.2S	The NAT ALG—SIP Trunking Support feature uses a local database to store all media-related information within a SIP trunk. Call IDs of each call are used to index this local database.
NAT Basic H.323 ALG Support	Cisco IOS XE Release 2.1	NAT requires a variety of ALGs to handle Layer 7 protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels. The NAT Basic H.323 ALG support feature provides these specific services for H.323 messages.
NAT DNS ALG Support	Cisco IOS XE Release 2.1	The NAT DNS ALG Support feature supports translation of DNS packets.
NAT FTP ALG Support	Cisco IOS XE Release 2.1	The NAT FTP ALG Support feature supports translation of FTP packets.

Feature Name	Releases	Feature Configuration Information
NAT H.323 RAS	Cisco IOS XE Release 2.4	NAT supports all H.225 and H. 245 message types, including those sent in the Registration, Admission, and Status (RAS) protocol. RAS provides a number of messages that are used by software clients and VoIP devices to register their location, request assistance in call setup, and control bandwidth. The RAS messages are directed toward an H.323 gatekeeper.
NAT ICMP ALG Support	Cisco IOS XE Release 2.1	The NAT ICMP ALG Support feature supports translation of ICMP packets.
NAT NetBIOS ALG Support	Cisco IOS XE Release 3.1S	NAT provides Network Basic Input Output System (NetBIOS) message translation support. The NAT NetBIOS ALG Support feature introduced the following command to display NetBIOS-specific information for a device: **show platform hardware qfp [active | standby] feature alg statistics netbios**.
NAT NetMeeting Directory (LDAP)	Cisco IOS XE Release 2.4	The NAT NetMeeting Directory (LDAP) feature provides ALG support for NetMeeting directory LDAP messages.
NAT RCMD ALG Support	Cisco IOS XE Release 3.1S	NAT provides remote command execution service (RCMD) message translation support.The NAT RCMD ALG Support feature introduced the following command to display RCMD-specific information for a device: **show platform software trace message process qfp active**.
NAT RTSP ALG Support	Cisco IOS XE Release 3.1S	The NAT RTSP ALG Support feature provides RTSP message translation support.
NAT—SCCP for Video	Cisco IOS XE Release 2.4	The NAT—SCCP for Video feature provides SCCP video message translation support.
NAT—SIP ALG Enhancement for T.38 Fax Relay	Cisco IOS XE Release 2.4.1	The NAT—SIP ALG Enhancement for T.38 Fax Relay feature provides translation support for SIP ALG support of T.38 Fax Relay over IP.
NAT—SIP Extended Methods	Cisco IOS XE Release 2.4	The NAT—SIP Extended Methods feature supports extended methods for SIP.
NAT Support of IP Phone to Cisco CallManager	Cisco IOS XE Release 2.1	The NAT Support of IP Phone to Cisco CallManager feature adds NAT support for configuring Cisco SCCP for a Cisco IP phone-to-Cisco CallManager communication.

Feature Name	Releases	Feature Configuration Information
NAT Support for IPsec ESP—Phase II	Cisco IOS XE Release 2.1	The NAT Support for IPsec ESP-- Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a device configured with NAPT.
NAT Support for SIP	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2S	The NAT Support for SIP feature adds the ability to deploy NAT between VoIP solutions based on SIP.
NAT TFTP ALG Support	Cisco IOS XE Release 2.1	The NAT TFTP ALG Support feature supports translation of TFTP packets.
NAT VRF-Aware ALG Support	Cisco IOS XE Release 2.5	The NAT VRF-Aware ALG Support feature supports VPN routing and forwarding (VRF) for protocols that have a supported ALG.
NAT vTCP ALG Support	Cisco IOS XE Release 3.1S Cisco IOS XE Release 3.2S	The NAT vTCP ALG Support feature provides vTCP support to handle TCP segmentation and reassembling for ALG.
Support for IPsec ESP Through NAT	Cisco IOS XE Release 2.1	The Support for IPsec ESP Through NAT feature provides the ability to support multiple, concurrent IPsec ESP tunnels or connections through a NAT device configured in Overload or PAT mode.



CHAPTER 77

Carrier Grade Network Address Translation

Carrier Grade Network Address Translation (CGN) is a large-scale NAT that translates private IPv4 addresses into public IPv4 addresses. CGN employs Network Address and Port Translation methods to aggregate multiple private IPv4 addresses into fewer public IPv4 addresses.

This module provides an overview of CGN and describes how to configure CGN.

Restrictions for Carrier Grade Network Address Translation

- Asymmetric routing with box-to-box (B2B) redundancy is not supported in Carrier Grade Network Address Translation (CGN) mode.
- B2B redundancy is not supported on broadband with CGN; B2B is supported on standalone CGN.
- Broadband is not supported with traditional NAT.
- CGN does not support IP sessions.
- NAT outside mappings are disabled automatically when CGN operating mode is configured using the **ip nat settings mode cgn** command.
- CGN does not support integration with Cisco Performance Routing (PfR). Commands with the **oer** keyword are not supported. For example, the **ip nat inside source route-map pool overload oer** and the **ip nat inside source list pool overload oer** commands are not supported.
- The **match-in-vrf** keyword for intra-VPN NAT is not supported with CGN.
- If you specify a destination port to configure timeout in CGN mode, the destination port is ignored and the local port is considered for timeout.
- The **ip nat settings log-destination** command is not supported in a Box-to-Box High-Availability set up.

Information About Carrier Grade Network Address Translation

Carrier Grade NAT Overview

Network Address Translation (NAT) is positioned between a private and public IP network and uses nonglobal, private IP addresses and a public IP address for translation. NAT dynamically maps one or more private IP addresses into one or more public (globally routable) IP addresses that use Network Address and Port Translation (NAPT) techniques. Traditionally, NAT boxes are deployed in residential home gateways (HGWs) to translate multiple private IP addresses that are configured on multiple devices inside the home to a single public IP address that is configured and provisioned on the HGW by the service provider. Service providers deploy NAT in such a way that multiple subscribers can share a single global IP address. The service provider NAT scales to several millions of NAT translations, making it a Carrier Grade NAT (CGN).

In CGN, packets that traverse from inside the network to outside require only the source address port translation; destination address port translation is not required. CGN can be standalone like traditional NAT or you can use it along with broadband access aggregation. CGN coexists with Intelligent Services Gateway (ISG) features such as Layer 4 Redirect and subscriber services such as traffic classes.

You can configure CGN by using the **ip nat settings mode cgn** command. Use the **ip nat settings mode default** command to change to the default or traditional NAT operating mode. In the CGN mode, you cannot configure any NAT outside mappings. Mode changes on an active NAT device are not allowed. However, when you change from the default NAT mode to CGN mode, all existing outside mappings have to be removed. Use the **no ip nat settings support mapping outside** command to remove all outside mappings and to prevent any new outside mappings from being configured. You can also remove outside mappings by using the **no** form of commands used to configure NAT outside. In case there are specific ports configured with TCP or UDP timeout values, remove the configuration of **ip nat translation port** *protocol port timeout* completely and configure the timeout values for these protocols using the same command. Alternatively, reload the device. Note, if you specify a destination port to configure timeout in CGN mode, the destination port is ignored and the local port is considered for timeout.

CGN increases the scalability of the number of NAT translations that can be supported because destination information is not stored.

CGN supports the following:

- All application-level gateways (ALGs) that are supported by traditional NAT. For more information about supported ALGs, see the *Using Application-Level Gateways with NAT* module of the *IP Addressing: NAT Configuration Guide*.
- Endpoint independent mapping and endpoint independent filtering.
- Hairpinning by using VRF-Aware Software Infrastructure (VASI) and policy-based routing (PBR). Hairpinning occurs when two subscribers are behind the same NAT device but can see each other only by using the global IP address.
- Interbox and intrabox redundancy.
- Lawful intercept.
- Logging of NAT high-speed logging (HSL) records. For more information about HSL, see the section "High-Speed Logging for NAT" in the *Maintaining and Monitoring NAT* module of the *IP Addressing: NAT Configuration Guide*.

- Multihoming, which is the ability to support multiple outside interfaces to provide connectivity through redundant or standby exit points. Depending on the configured routing topology, any exit interface that is marked as an outside interface can use a translation that was created previously.
- TCP timeout value of 2 hours and 4 minutes.
- VPN routing and forwarding (VRF)-aware NAT.
- CGN NAT can scale to higher number of translations on ESP200 using the **ip nat settings scale bind** command.

Carrier Grade NAT Support for Broadband Access Aggregation

You can configure Carrier Grade Network Address Translation (CGN) as an independent feature or use CGN along with broadband access aggregation.

Broadband access aggregation enables connections between multiple technologies such as cable, digital subscriber line (DSL), Ethernet, ISDN, and wireless devices that are connected to corporate VPNs, third-party applications, and the Internet.

PPP over Ethernet (PPPoE) connects hosts on a network over a simple bridging device to a remote aggregation concentrator. PPPoE is the predominant access protocol in broadband networks worldwide.

For PPPoE to work with CGN, either the virtual templates or the RADIUS server must provide the Network Address Translation (NAT) inside configuration. The NAT inside configuration can be downloaded as part of the RADIUS authentication or alternatively configure the **ip nat inside** command on the virtual template. This gets cloned into a virtual access interface that inherits the ip nat inside configuration. For the RADIUS server to provide the NAT inside configuration, configure the **aaa policy interface-config allow-subinterface** global command or configure the Cisco attribute-value pairs (AV pairs) lcp:allow-subinterface=yes and then include lcp:interface-config=ip nat inside in the RADIUS profile on a per-subscriber basis.

You can terminate a PPPoE session either in the global routing table or at a VRF instance.

CGN supports dual-stack (IPv4 and IPv6) PPP sessions. However, only IPv4 traffic is subject to NAT. The IPv6 traffic is not translated; it is routed as per the IPv6 routing configuration.

How to Configure Carrier Grade Network Address Translation

Based on your network configuration, you can configure static, dynamic, or dynamic PAT Carrier Grade NAT.

Note You must use at least one of the configurations described in the following tasks for Carrier Grade NAT to work.

Configuring Static Carrier Grade NAT

Static address translation (static NAT) allows one-to-one mapping between local and global addresses. Use the **ip nat inside source static** command to enable static NAT of the inside source address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat settings mode cgn**
4. **ip nat inside source static** *local-ip global-ip*
5. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip nat outside**
10. **end**
11. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat settings mode cgn** **Example:** `Device(config)# ip nat settings mode cgn`	Enables CGN operating mode.
Step 4	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 192.168.2.1 192.168.34.2`	Enables static Carrier Grade NAT of the inside source address.
Step 5	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# interface gigabitethernet 0/0/4`	Configures an interface and enters interface configuration mode. **Note** The NAT inside network can be applied to interface virtual-template when the router is used for broadband aggregation deployment.
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Indicates that the interface is connected to the inside network (the network that is subject to NAT translation).
Step 7	**exit** **Example:**	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 9	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.
Step 11	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	Displays active NAT translations.

Example

The following is sample output from the **show ip nat translations** command:

```
Device# show ip nat translations

Pro  Inside global         Inside local          Outside local         Outside global
udp  10.5.5.1:1025          192.0.2.1:4000        ---                    ---
udp  10.5.5.1:1024          192.0.2.3:4000        ---                    ---
udp  10.5.5.1:1026          192.0.2.2:4000        ---                    ---

Total number of translations: 3
```

The following is sample output from the **show ip nat translations verbose** command:

```
Device# show ip nat translations verbose

Pro  Inside global         Inside local          Outside local         Outside global
udp  10.5.5.1:1025          192.0.2.1:4000        ---                    ---
  create: 02/15/12 11:38:01, use: 02/15/12 11:39:02, timeout: 00:00:00
  Map-Id(In): 1
  Mac-Address: 0000.0000.0000    Input-IDB: TenGigabitEthernet1/1/0
  entry-id: 0x0, use_count:1

udp  10.5.5.1:1024          192.0.2.3:4000        ---                    ---
  create: 02/15/12 11:38:00, use: 02/15/12 11:39:02, timeout: 00:00:00
  Map-Id(In): 1
  Mac-Address: 0000.0000.0000    Input-IDB: TenGigabitEthernet1/1/0
  entry-id: 0x0, use_count:1

udp  10.5.5.1:1026          192.0.2.2:4000        ---                    ---
  create: 02/15/12 11:38:00, use: 02/15/12 11:39:02, timeout: 00:00:00
  Map-Id(In): 1
  Mac-Address: 0000.0000.0000    Input-IDB: TenGigabitEthernet1/1/0
  entry-id: 0x0, use_count:1
```

```
Total number of translations: 3
```

Configuring Dynamic Carrier Grade NAT

Dynamic address translation (dynamic NAT) maps unregistered IP addresses to registered IP addresses from a pool of registered IP addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat settings mode cgn**
4. **access-list** *standard-access-list-number* **permit** *source wildcard*
5. **access-list** *standard-access-list-number* **permit** *source wildcard*
6. **route-map** *map-tag*
7. **match ip address** [*access-list-number*]
8. **match ip next-hop** [*access-list-number*]
9. **exit**
10. **ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length*
11. **ip nat inside source route-map** *name* **pool** *name*
12. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
13. **ip nat inside**
14. **exit**
15. **interface** *type number*
16. **ip nat outside**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat settings mode cgn** **Example:** `Device(config)# ip nat settings mode cgn`	Enables CGN operating mode.
Step 4	**access-list** *standard-access-list-number* **permit** *source wildcard* **Example:**	Defines a standard access list and specifies a host. • Access list 1 defined in this step is used by the **match ip address** command.

	Command or Action	Purpose
	`Device(config)# access-list 1 permit 10.1.1.0 0.0.0.255`	
Step 5	**access-list** *standard-access-list-number* **permit** *source wildcard* **Example:** `Device(config)# access-list 2 permit 10.5.5.0 0.0.0.255`	Defines a standard access list and specifies a host. • Access list 2 defined in this step is used by the **match ip next-hop** command.
Step 6	**route-map** *map-tag* **Example:** `Device(config)# route-map nat-route-map`	Defines conditions for redistributing routes from one routing protocol into another or enables policy routing and enters route-map configuration mode.
Step 7	**match ip address** [*access-list-number*] **Example:** `Device(config-route-map)# match ip address 1`	Distributes any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list or performs policy routing on packets.
Step 8	**match ip next-hop** [*access-list-number*] **Example:** `Device(config-route-map)# match ip next-hop 2`	Redistributes any routes that have a next-hop router address passed by one of the specified access lists.
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 10	**ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length* **Example:** `Device(config)# ip nat pool nat-pool 10.1.1.1 10.1.254.254 prefix-length 16`	Defines a pool of IP addresses for NAT.
Step 11	**ip nat inside source route-map** *name* **pool** *name* **Example:** `Device(config)# ip nat inside source route-map nat-route-map pool nat-pool`	Enables dynamic NAT of the inside source address.
Step 12	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# interface gigabitethernet 0/0/5`	Configures an interface and enters interface configuration mode. **Note** The NAT inside network can be applied to interface virtual-template when the router is used for broadband aggregation deployment.
Step 13	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Indicates that the interface is connected to the inside network (the network that is subject to NAT translation).

	Command or Action	Purpose
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 16	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring Dynamic Port Address Carrier Grade NAT

Port Address Translation (PAT) or overloading is a form of dynamic NAT that maps multiple unregistered IP addresses to a single registered IP address (many-to-one mapping) by using different ports. PAT enables thousands of users to connect to the Internet by using only one real global IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat settings mode cgn**
4. **ip nat inside source list** *number* **pool** *name* [**overload**]
5. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
6. **access-list** *standard-access-list-number* **permit** *source wildcard*
7. **interface gigabitethernet** *card/spaslot/port.subinterface-number*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip nat outside**
12. **end**
13. **show ip nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat settings mode cgn** **Example:** `Device(config)# ip nat settings mode cgn`	Enables CGN operating mode.
Step 4	**ip nat inside source list** *number* **pool** *name* [**overload**] **Example:** `Device(config)# ip nat inside source list 1 pool nat-pool overload`	Enables the router to use one global address for many local addresses. • When you configure the **overload** keyword, the TCP or UDP port number of each inside host distinguishes between multiple conversations using the same local IP address. • The **overload** keyword configures overloading or PAT.
Step 5	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool nat-pool 10.1.1.1 10.1.254.254 netmask 255.255.0.0`	Defines a pool of IP addresses for NAT.
Step 6	**access-list** *standard-access-list-number* **permit** *source wildcard* **Example:** `Device(config)# access-list 1 permit 172.16.0.0 255.255.0.0`	Defines a standard access list and specifies a host.
Step 7	**interface gigabitethernet** *card/spaslot/port.subinterface-number* **Example:** `Device(config)# interface gigabitethernet 0/0/6`	Configures an interface and enters interface configuration mode. **Note** The NAT inside network can be applied to interface virtual-template when the router is used for broadband aggregation deployment.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Indicates that the interface is connected to the inside network (the network that is subject to NAT translation).
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:**	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface gigabitethernet 0/0/2`	
Step 11	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.
Step 13	**show ip nat statistics** **Example:** `Device# show ip nat statistics`	Displays NAT statistics.

Example

The following is sample output from the **show ip nat statistics** command:

```
Device# show ip nat statistics

Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Outside interfaces:
  TenGigabitEthernet2/0/0, TenGigabitEthernet2/1/0, TenGigabitEthernet2/2/0
  TenGigabitEthernet2/3/0
Inside interfaces:
  TenGigabitEthernet1/0/0, TenGigabitEthernet1/1/0, TenGigabitEthernet1/2/0
  TenGigabitEthernet1/3/0
Hits: 59230465  Misses: 3
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 102 pool mypool refcount 3
 pool mypool: netmask 255.255.255.0
        start 10.5.5.1 end 10.5.5.5
        type generic, total addresses 5, allocated 1 (20%), misses 0
nat-limit statistics:
 max entry: max allowed 2147483647, used 3, missed 0
Pool stats drop: 0  Mapping stats drop: 0
Port block alloc fail: 0
IP alias add fail: 0
Limit entry add fail: 0
```

Logging Destination IP Address and Port Details in Carrier Grade NAT (CGN) Mode

In the Carrier Grade NAT (CGN) mode, the destination IP address and port details are not logged when High Speed Logging (HSL) records are generated. You can still log the destination IP address and destination port details using the classic NAT mode, but that does not support Endpoint-independent filtering (EIF).

Once the **ip nat settings log-destination** command is configured in the Carrier Grade NAT (CGN) mode, the destination IP address and destination port details are included in the add and delete HSL records.

To enable including the destination IP and destination port information in the HSL messages for Carrier Grade NAT (CGN) mode, use the following **ip nat settings log-destination** command.

Example

```
Device# show run | in log
ip nat settings log-destination
ip nat log translations flow-export v9 udp ipv6-destination 2001::2 30000 source
GigabitEthernet0/0/3
ip nat log translations flow-export v9 udp destination 172.27.61.85 20000
```

Configuration Examples for Carrier Grade Network Address Translation

Example: Configuring Static Carrier Grade NAT

```
Device# configure terminal
Device(config)# ip nat settings mode cgn
Device(config)# ip nat inside source static 192.168.2.1 192.168.34.2
Device(config)# interface gigabitethernet 0/0/6
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip nat outside
Device(config-if)# end
```

Example: Configuring Dynamic Carrier Grade NAT

```
Device# configure terminal
Device(config)# ip nat settings mode cgn
Device(config)# access-list 1 permit 10.1.1.0 0.0.0.255
Device(config)# access-list 2 permit 10.5.5.0 0.0.0.255
Device(config)# route-map nat-route-map
Device(config-route-map)# match ip address 1
Device(config-route-map)# match ip next-hop 2
Device(config-route-map)# exit
Device(config)# ip nat pool nat-pool 10.1.1.1 10.1.254.254 prefix-length 16
Device(config)# ip nat inside source route-map nat-route-map pool nat-pool
Device(config)# interface gigabitethernet 0/0/5
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# ip nat outside
Device(config-if)# end
```

Example: Configuring Dynamic Port Address Carrier Grade NAT

```
Device# configure terminal
Device(config)# ip nat settings mode cgn
Device(config)# ip nat inside source list 1 pool nat-pool overload
Device(config)# ip nat pool nat-pool 10.1.1.1 10.1.254.254 netmask 255.255.0.0
Device(config)# access-list 1 permit 172.16.0.0 255.255.0.0
Device(config)# interface gigabitethernet 0/0/4
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/3
Device(config-if)# ip nat outside
Device(config-if)# end
```

Additional References for Carrier Grade Network Address Translation

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
NAT commands	IP Addressing Command Reference
NAT ALGs	“Using Application-Level Gateways with NAT”
HSL messages	“Monitoring and Maintaining NAT”

Standards and RFCs

Standard/RFC	Title
RFC 4787	*Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*
RFC 5582	*Location-to-URL Mapping Architecture and Framework*

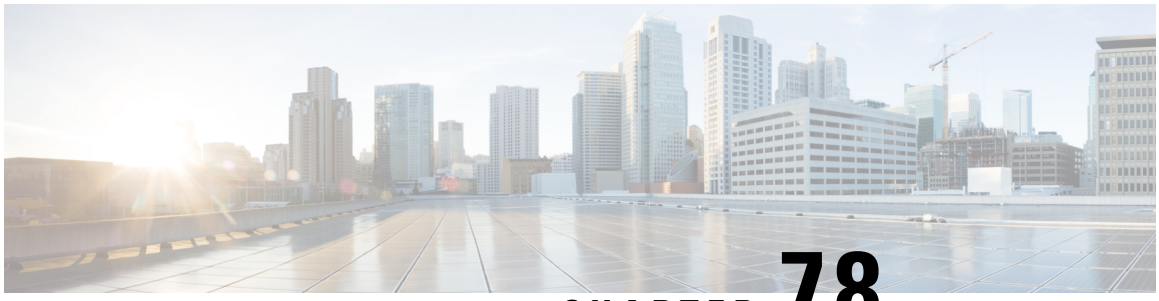
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Carrier Grade Network Address Translation

Table 119: Feature Information for Carrier Grade Network Address Translation

Feature Name	Releases	Feature Information
Carrier Grade Network Address Translation	Cisco IOS XE Release 3.6S	Carrier Grade Network Address Translation (CGN) is a large-scale NAT that translates private IPv4 addresses into public IPv4 addresses. CGN employs Network Address and Port Translation methods to aggregate multiple private IPv4 addresses into fewer public IPv4 addresses. The following commands were introduced or modified: **ip nat settings mode** and **ip nat settings support mapping outside**. **Note** This feature is not supported on ISR 4000 platform.



CHAPTER 78

Static NAT Mapping with HSRP

This module contains procedures for configuring Network Address Translation (NAT) to support the increasing need for highly resilient IP networks. This network resiliency is required where application connectivity needs to continue unaffected by failures to links and routers at the NAT border.

Prerequisites for Static NAT Mapping with HSRP

To understand how high availability is implemented see the “High Availability Overview” module in the .

Restrictions for Static NAT Mapping with HSRP

- Using any IP address configured on a device IP address as an address pool or in a NAT static rule is not supported. NAT can share the physical interface address (not any other IP address) of a device only by using the NAT interface overload configuration. A device uses the ports of its physical interface and NAT must receive communication about the ports that it can safely use for translation. This communication happens only when the NAT interface overload is configured.
- Virtual routing and forwarding (VRF) NAT with Hot Standby Router Protocol (HSRP) is not supported. Effective with Cisco IOS XE Denali 16.3.3, this restriction is not applicable. Upgrade to this release if you want your device to support VRF NAT with HSRP.
- Static NAT mappings must be mirrored on two or more HSRP devices, because the NAT state is not exchanged between devices running NAT in an HSRP group.
- If you configure both HSRP devices with the same static NAT and the **hsrp** keyword to link these devices to the same HSRP group is not configured, the behavior of the devices will be unpredictable.

Information About Static NAT Mapping with HSRP

Static Mapping Support with HSRP for High Availability Feature Overview

When an Address Resolution Protocol (ARP) query is triggered for an address that is configured with NAT static mapping and owned by the device, NAT responds with the burned in MAC (BIA MAC) address on the interface to which the ARP is pointing. Two devices act as the Hot Standby Router Protocol (HSRP) active and standby. You must enable and configure the NAT outside interfaces of the active and standby devices to belong to a group.

Address Resolution with ARP

A device in IP can have both a local address (which uniquely identifies the device on its local segment or LAN) and a network address (which identifies the network to which the device belongs). The local address is known as a data link address because it is contained in the data link layer (Layer 2 of the OSI model) part of the packet header and is read by data-link devices such as bridges, all device interfaces and so on. The local address is referred to as the MAC address, because the MAC sublayer within the data-link layer processes addresses for the layer.

To communicate with a device on Ethernet, for example, the Cisco IOS software must first determine the 48-bit MAC or local data-link address of that device. The process of determining the local data-link address from an IP address is called address resolution. The process of determining the IP address from a local data-link address is called reverse address resolution.

The software uses three forms of address resolution: Address Resolution Protocol (ARP), proxy ARP, and Probe (similar to ARP). The software also uses the Reverse Address Resolution Protocol (RARP). ARP, proxy ARP, and RARP are defined in RFCs 826, 1027, and 903, respectively. Probe is a protocol developed by the Hewlett-Packard Company (HP) for use on IEEE-802.3 networks.

ARP is used to associate IP addresses with media or MAC addresses. Taking an IP address as input, ARP determines the associated media address. Once a media or MAC address is determined, the IP address or media address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP).

Gratuitous ARP

When a host sends an ARP request to resolve its own IP address, it is called gratuitous ARP. In the ARP request packet, the source and destination IP addresses are filled with the same source IP address itself. The destination MAC address is the Ethernet broadcast address.

When a router becomes active, it broadcasts a gratuitous ARP packet with the Hot Standby Router Protocol (HSRP) virtual MAC address to the affected LAN segment. If the segment uses an Ethernet switch, this allows the switch to change the location of the virtual MAC address so that packets flow to the new router instead of the one that is no longer active. End devices do not actually need gratuitous ARP if routers use the default HSRP MAC address.

How to Configure Static NAT Mapping with HSRP

Configuring NAT Static Mapping Support for HSRP

When an Address Resolution Protocol (ARP) query is triggered for an address that is configured with NAT static mapping and owned by the router, NAT responds with the burned in MAC (BIA MAC) address on the interface to which the ARP is pointing. Two routers are acting as HSRP active and standby. Their NAT outside interfaces must be enabled and configured to belong to a group.

Benefits of Configuring Static Mapping Support for HSRP are the following:

- Using static mapping support for HSRP, failover is ensured without having to time out and repopulate upstream ARP caches in a high-availability environment, where HSRP router pairs have identical NAT configuration for redundancy.
- Static mapping support for HSRP allows the option of having only the HSRP active router respond to an incoming ARP for a router configured with a NAT address.

Note Static mapping for HSRP with outside-source NAT, via command **ip nat outside source static local-ip global-ip redundancy group-name** is not supported.

Both of the following tasks are required and must be performed on both the active and standby routers to configure NAT static mapping support for HSRP:

Enabling HSRP on the NAT Interface

Perform this task to enable HSRP on the NAT interface of both the active and standby routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **no ip redirects**
6. **ip nat** {**inside** | **outside**}
7. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
8. **standby** [*group-number*] **preempt**
9. **standby** [*group-number*] **ip** [*ip-address* | **secondary**]
10. **standby** [*group-number*] **name** [*group-name*]
11. **standby** [*group-number*] **track** *interface-number*
12. **end**
13. **show standby**
14. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/1/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.5.27 255.255.255.0`	Sets the primary IP address on the interface.
Step 5	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables the sending of redirect messages
Step 6	**ip nat** {**inside** | **outside**} **Example:** `Device(config)# ip nat outside`	Connects the interface to the inside network.
Step 7	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 10 priority 105`	Enables the HSRP protocol.
Step 8	**standby** [*group-number*] **preempt** **Example:** `Device(config-if)# standby 10 preempt`	Configures HSRP preemption.
Step 9	**standby** [*group-number*] **ip** [*ip-address* | **secondary**] **Example:** `Device(config-if)# standby 10 ip 192.168.5.30`	Enables the HSRP protocol.
Step 10	**standby** [*group-number*] **name** [*group-name*] **Example:** `Device(config-if)# standby 10 name HSRP1`	Sets the HSRP group name.
Step 11	**standby** [*group-number*] **track** *interface-number* **Example:**	Configures HSRP to track an object and to change the hot standby priority on the basis of the state of the object.

	Command or Action	Purpose
	`Device(config-if)# standby 10 track gigabitethernet1/1/1`	
Step 12	**end** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 13	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information
Step 14	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations verbose`	(Optional) Displays active NAT translations.

Enabling Static NAT for HSRP

Before you begin

To enable static mapping support with HRSP for high availability, perform this task on both the active and standby devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip* **redundancy** *group-name*
4. **ip classless**
5. **ip route prefix mask** *interface-type interface-number*
6. **no ip http server**
7. **end**
8. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 3	**ip nat inside source static** *local-ip global-ip* **redundancy** *group-name* **Example:** `Device(config)# ip nat inside source static 10.10.10.5 192.168.5.33 redundancy HSRP1`	Enables a device to respond to Address Resolution Protocol (ARP) queries using BIA MAC, if HSRP is configured on the NAT outside interface.
Step 4	**ip classless** **Example:** `Device(config)# ip classless`	Enables a device to forward packets that are destined for a subnet of a network that has no network default route, to the best supernet route possible.
Step 5	**ip route prefix mask** *interface-type interface-number* **Example:** `Device(config)# ip route 10.10.10.0 255.255.255.0 gigabitethernet 0/0/0`	Establishes static routes.
Step 6	**no ip http server** **Example:** `Device(config)# no ip http server`	Enables the HTTP server on your IP system.
Step 7	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations verbose`	(Optional) Displays active NAT translations. **Note** Static mapping for HSRP with outside-source NAT, via command **ip nat outside source static local-ip global-ip redundancy group-name** is not supported.

Configuration Example for Static NAT Mapping with HSRP

Example: Configuring Static NAT in an HSRP Environment

The following example shows support for NAT with a static configuration in an HSRP environment. Two devices act as HSRP active and standby, and the NAT outside interfaces are HSRP enabled and configured to belong to group HSRP1.

Active Device Configuration

```
interface BVI10
 ip address 192.168.5.54 255.255.255.255.0
 no ip redirects
 ip nat outside
 standby 10 priority 105 preempt
 standby 10 name HSRP1
```

```
 standby 10 ip 192.168.5.30
 standby 10 track gigabitethernet1/1/1
!
!
 ip default-gateway 10.0.18.126
 ip nat inside source static 10.10.10.5 192.168.5.33 redundancy HSRP1
 ip classless
 ip route 10.10.10.0 255.255.255.0 gigabitethernet1/1/1
 ip route 172.22.33.0 255.255.255.0 gigabitethernet1/1/1
 no ip http server
```

Standby Device Configuration

```
interface BVI10
 ip address 192.168.5.56 255.255.255.255.0
 no ip redirects
 ip nat outside
 standby 10 priority 100 preempt
 standby 10 name HSRP1
 standby 10 ip 192.168.5.30
 standby 10 track gigabitethernet0/0/1
!
 ip default-gateway 10.0.18.126
 ip nat inside source static 10.10.10.5 192.168.5.33 redundancy HSRP1
 ip classless
 ip route 10.0.32.231 255.255.255 gigabitethernet0/0/1
 ip route 10.10.10.0 255.255.255.0 gigabitethernet0/0/1
 no ip http server
```

Additional References for Static NAT Mapping with HSRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands: complete command syntax, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
IP Access List Sequence Numbering	*IP Access List Sequence Numbering* document
NAT configuration tasks	"Configuring NAT for IP Address Conservation" module
NAT maintenance	"Monitoring and Maintaining NAT" module
Using NAT with MPLS VPNs	"Integrating NAT with MPLS VPNs" module

Standards and RFCs

Standard/RFC	Title
RFC 903	*Reverse Address Resolution Protocol*

Standard/RFC	Title
RFC 826	*Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*
RFC 1027	*Using ARP to implement transparent subnet gateways*

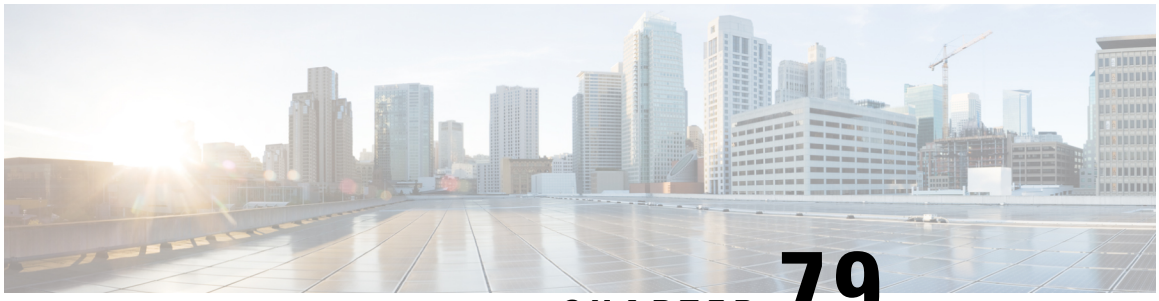
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Static NAT Mapping with HSRP

Table 120: Feature Information for Static NAT Mapping with HSRP

Feature Name	Releases	Feature Configuration Information
NAT—Static Mapping Support with HSRP for High Availability	Cisco IOS XE Release 2.1	Static mapping support for HSRP allows the option of having only the HSRP active router respond to an incoming ARP for a router configured with a NAT address.



CHAPTER 79

VRF-Aware Dynamic NAT Mapping with HSRP

The VRF-Aware Dynamic NAT Mapping with HSRP feature supports stateless redundancy using HSRP with dynamic Network Address Translation (NAT), Port Address Translation (PAT), and interface overload configuration. Dynamic NAT, PAT and interface overload support HSRP with and without virtual routing and forwarding (VRF) instances. All these configurations are supported in the Carrier Grade NAT (CGN) mode.

This module describes the feature and explains how to configure it.

Prerequisites for VRF-Aware Dynamic NAT Mapping with HSRP

- Both the active and standby devices must be configured with the same Network Address Translation (NAT) rules.
- Hot Standby Router Protocol (HSRP) must be configured between the active and standby devices.

Restrictions for VRF-Aware Dynamic NAT Mapping with HSRP

- During failovers, NAT translated IP addresses on devices may be different from the IP address before the failover, because no state information is exchanged between active and standby devices.
- During a failover, all existing NAT sessions are destroyed and new sessions are established in the active device.
- HSRP Virtual IP Address (VIP) cannot be used by NAT pools.
- Active/active configuration is not supported; only active/standby configuration is supported.
- IPv6 is not supported; only IPv4 is supported.

Information About VRF-Aware Dynamic NAT Mapping with HSRP

VRF-Aware Dynamic NAT Mapping with HSRP Overview

The VRF-Aware Dynamic NAT Mapping with HSRP feature supports stateless redundancy using HSRP with dynamic Network Address Translation (NAT), Port Address Translation (PAT), and interface overload configuration. Dynamic NAT, PAT and interface overload support HSRP with and without virtual routing and forwarding (VRF) instances. All these configurations are supported in the Carrier Grade NAT (CGN) mode.

Hot Standby Router Protocol (HSRP) provides high network availability by providing first-hop routing redundancy for IP hosts on networks configured with a default gateway IP address. HSRP is used in a group of routers for selecting an active device and a standby device. HSRP provides redundancy for routing IP traffic without being dependent on the availability of a single router. In a group of device interfaces, the active device is the device of choice for routing packets; the standby device is the device that takes over when the active device fails or when preset conditions are met.

Devices running HSRP send and receive multicast UDP-based hello packets to detect router failure and to designate active and standby devices. Selection of active and standby devices is based on the assigned priority. The device with the highest priority is selected as the active device. After failover, a new active device sends a gratuitous Address Resolution Protocol (ARP) request to LAN users to notify about the change in MAC address for the virtual IP address (VIP).

To enable this feature, both the active and standby devices must be configured with the same NAT rules and HSRP must be configured on both the devices. Based on the configured priority one of the devices will be active and the other standby. This feature supports VRF-aware NAT translation and Carrier Grade NAT (CGN) mode.

This feature supports the LAN-LAN topology as well as the LAN-WAN topology. In the LAN-WAN topology, only symmetric routing is supported.

When an Address Resolution Protocol (ARP) query is triggered for an address that is configured with dynamic NAT mapping and owned by the device, NAT responds with the burned-in MAC (BIA MAC) address on the interface to which the ARP is pointing. You must enable and configure the NAT inside interfaces of the active and standby devices to belong to a group.

In Cisco IOS XE Denali 16.3 release, the Allow same ACL/router-map on multiple NAT statements feature was introduced to support usage of same ACL for configuring both dynamic mapping and static mapping in NAT. Dynamic mapping is given the precedence over static mapping regardless of the configuration order. The precedence of dynamic mapping over static mapping using the sequence number of the class ensures class order consistency in NAT.

Address Resolution with ARP

A device in IP can have both a local address (which uniquely identifies the device on its local segment or LAN) and a network address (which identifies the network to which the device belongs). The local address is known as a data link address because it is contained in the data link layer (Layer 2 of the OSI model) part of the packet header and is read by data-link devices such as bridges, all device interfaces and so on. The local

address is referred to as the MAC address, because the MAC sublayer within the data-link layer processes addresses for the layer.

To communicate with a device on Ethernet, for example, the Cisco IOS software must first determine the 48-bit MAC or local data-link address of that device. The process of determining the local data-link address from an IP address is called address resolution. The process of determining the IP address from a local data-link address is called reverse address resolution.

The software uses three forms of address resolution: Address Resolution Protocol (ARP), proxy ARP, and Probe (similar to ARP). The software also uses the Reverse Address Resolution Protocol (RARP). ARP, proxy ARP, and RARP are defined in RFCs 826, 1027, and 903, respectively. Probe is a protocol developed by the Hewlett-Packard Company (HP) for use on IEEE-802.3 networks.

ARP is used to associate IP addresses with media or MAC addresses. Taking an IP address as input, ARP determines the associated media address. Once a media or MAC address is determined, the IP address or media address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP).

Gratuitous ARP

When a host sends an ARP request to resolve its own IP address, it is called gratuitous ARP. In the ARP request packet, the source and destination IP addresses are filled with the same source IP address itself. The destination MAC address is the Ethernet broadcast address.

When a router becomes active, it broadcasts a gratuitous ARP packet with the Hot Standby Router Protocol (HSRP) virtual MAC address to the affected LAN segment. If the segment uses an Ethernet switch, this allows the switch to change the location of the virtual MAC address so that packets flow to the new router instead of the one that is no longer active. End devices do not actually need gratuitous ARP if routers use the default HSRP MAC address.

How to Configure VRF-Aware Dynamic NAT Mapping with HSRP

Enabling HSRP for VRF-Aware Dynamic NAT

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **interface** *type number* {**ip** | **ipv6** | **line-protocol**}
4. **exit**
5. **interface** *type number*
6. **ip nat inside**
7. **ip address** *ip-address mask*
8. **standby** *group-number* **ip** [*ip-address*]
9. **standby use-bia**
10. **standby** *group-number* **priority** *priority*
11. **standby** *group-number* **preempt** [*delay*]
12. **standby** *group-number* **track** *object-number* [**decrement** *priority-decrement*]

13. **exit**
14. **ip nat pool** *pool-name start-ipend-ip* **netmask** *netmask*
15. **access-list** *standard-access-list* **permit** *ip-address mask*
16. **ip nat inside source list** *list-name* **pool** *pool-name* [**overload**]
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *object-number* **interface** *type number* {**ip** | **ipv6** | **line-protocol**} **Example:** `Device(config)# track 10 interface gigabitethernet 0/0/0 line-protocol`	Configures an interface to be tracked where the Gateway Load Balancing Protocol (GLBP) weighting changes based on the state of the interface
Step 4	**exit** **Example:** `Device(config-track)# exit`	Exits tracking configuration mode and returns to global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/1`	Configures an interface and enters interface configuration mode.
Step 6	**ip nat inside** **Example:** `Device(config-f)# ip nat inside`	Connects the interface to the inside network, which is subject to Network Address Translation (NAT).
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.0.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 8	**standby** *group-number* **ip** [*ip-address*] **Example:** `Device(config-if)# standby 1 ip 192.0.0.1`	Activates the Hot Standby Router Protocol (HSRP).
Step 9	**standby use-bia** **Example:** `Device(config-if)# standby use-bia`	Configures HSRP to use the burned-in address of the interface as its virtual MAC address, instead of the preassigned MAC address or the functional address.

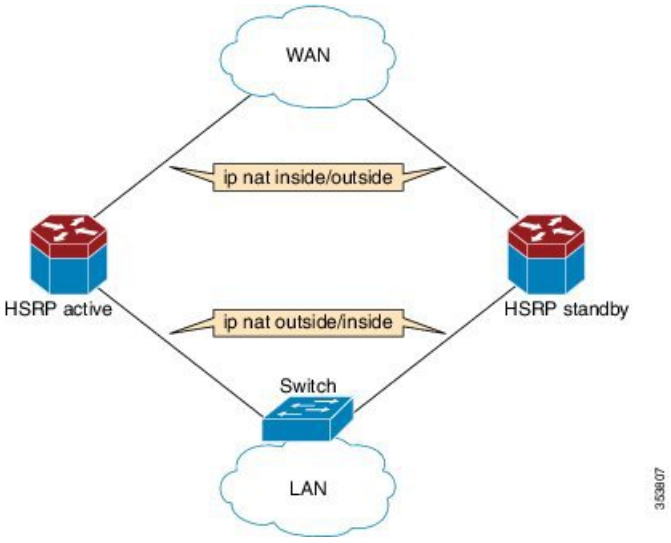
	Command or Action	Purpose
Step 10	**standby** *group-number* **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 120`	Configures the HSRP priority. • The priority range is from 1 to 255, where 1 denotes the lowest priority and 255 denotes the highest priority. The default priority value is 100. The device in the HSRP group with the highest priority value becomes the active device.
Step 11	**standby** *group-number* **preempt** [*delay*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption and preemption delay. • If you configure this command, when a local device has an HSRP priority higher than the current active device, the local device assumes control as the active device. If preemption is not configured, the local device assumes control as the active device only if it receives information indicating no device is in the active state (acting as the designated device).
Step 12	**standby** *group-number* **track** *object-number* [**decrement** *priority-decrement*] **Example:** `Device(config-if)# standby 1 track 10 decrement 15`	Configure HSRP to track an object, and change the HSRP priority on the basis of the state of the object. • When a tracked object goes down, the HSRP priority decreases by 10. If an object is not tracked, state changes do not affect the priority.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 14	**ip nat pool** *pool-name start-ipend-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool pool1 10.1.1.10 10.1.1.52 netmask 255.255.255.0`	Defines a pool of IP addresses for Network Address Translation (NAT) translations.
Step 15	**access-list** *standard-access-list* **permit** *ip-address mask* **Example:** `Device(config)# acces-list 1 permit 190.0.0.0 0.255.255.255`	
Step 16	**ip nat inside source list** *list-name* **pool** *pool-name* [**overload**] **Example:** `Device(config)# ip nat inside source list list1 pool pool1 overload`	Enables NAT of the inside source address. • When overloading is configured, it enables the device to use one global address for many local addresses. The TCP or UDP port number of each inside host distinguishes between the multiple conversations using the same local IP address.
Step 17	**end** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configuration Examples for VRF-Aware Dynamic NAT Mapping with HSRP

Example: Enabling HSRP for VRF-Aware Dynamic NAT

Figure 84: HSRP NAT LAN-WAN Topology



The following example shows a LAN-WAN configuration for dynamic Network Address Translation (NAT) overload mapping with Hot Standby Router Protocol (HSRP). A virtual routing and forwarding (VRF) instance is enabled for this configuration. Devices that are configured with NAT do not have any route configurations related to HSRP Virtual IP Address (VIP). LAN users using static routes have to set the default route or next-hop to the HSRP VIP; for example configure the **ip route 0.0.0.0 0.0.0.0 192.0.2.1** command.

```
! Active device configuration:
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# exit
Device(config)# track 10 interface fastethernet 1/1/1 line-protocol
Device(config-track)# exit
Device(config)# interface fastethernet 1/1/0
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ip nat inside
Device(config-if)# ip address 192.0.2.2 255.255.255.240
Device(config-if)# standby 1 ip 192.0.2.1
Device(config-if)# standby use-bia
Device(config-if)# standby 1 priority 120
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 track 10 decrement 15
```

```
Device(config-if)# exit
Device(config)# interface fastethernet 1/1/1
Device(config-if)# ip address 198.51.100.1 255.255.255.240
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# ip nat pool pool1 10.1.1.1 10.1.1.255 netmask 255.255.255.0
Device(config)# access-list 1 permit 203.0.113.0 255.255.255.240
Device(config)# ip nat inside source list1 pool pool1 vrf vrf1 overload
Device(config)# end


! Standby device configuration:
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# exit
Device#(config)# interface fastethernet 1/2/0
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ip nat inside
Device(config-if)# ip address 192.0.2.3 255.255.255.240
Device(config-if)# standby 1 ip 192.0.2.1
Device(config-if)# standby use-bia
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# exit
Device(config)# interface fastethernet 1/2/1
Device(config-if)# ip address 172.16.0.1 255.255.224.0
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# ip nat pool pool1 10.1.1.1 10.1.1.255 netmask 255.255.255.0
Device(config)# access-list 1 permit 2013.0.113.0 255.255.255.240
Device(config)# ip nat inside source list1 pool pool1 vrf vrf1 overload
Device(config)# end
```

Verifying HSRP for VRF-Aware Dynamic NAT

Before you begin

-

SUMMARY STEPS

1. enable
2. show arp
3. show ip alias
4. show ip nat translations
5. show standby brief

DETAILED STEPS

Step 1 enable

Example:

```
Device> enable
```

enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show arp**

Displays the entries in the Address Resolution Protocol (ARP) table.

Example:

```
Device# show arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.0.0.1              -   0023.eb85.7650  ARPA   GigabitEthernet1/1/0
Internet  192.0.0.2              -   0023.eb85.7650  ARPA   GigabitEthernet1/1/0
```

Step 3 **show ip alias**

Displays the IP addresses that are mapped to TCP ports (aliases) and Serial Line Internet Protocol (SLIP) addresses, which are treated similar to aliases.

Example:

```
Device# show ip alias

Address Type             IP Address      Port
Interface                10.39.21.3
Dynamic                  192.0.0.1
Interface                192.0.0.2
```

Step 4 **show ip nat translations**

Displays active Network Address Translation (NAT) translations.

Example:

```
Device# show ip nat translations

Pro  Inside global            Inside local          Outside local         Outside global
udp  10.1.1.4:512             190.0.0.1:435         193.0.0.1:80          193.0.0.1:80
udp  10.1.1.4:515             190.0.0.5:435         193.0.0.1:80          193.0.0.1:80
udp  10.1.1.4:514             190.0.0.4:435         193.0.0.1:80          193.0.0.1:80
udp  10.1.1.4:518             190.0.0.3:435         193.0.0.1:80          193.0.0.1:80
```

Step 5 **show standby brief**

Displays Hot Standby Router Protocol (HSRP) information in a single line of output for each standby group.

Example:

```
Device# show standby brief

                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State   Active          Standby         Virtual IP
Ga1/1/0     1    120 P Active  local           192.0.0.3       192.0.0.1
```

Additional References VRF-Aware Dynamic NAT Mapping with HSRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference
Static NAT with HSRP	"Static NAT Mapping with HSRP" module of the *IP Addressing: NAT Configuration Guide*

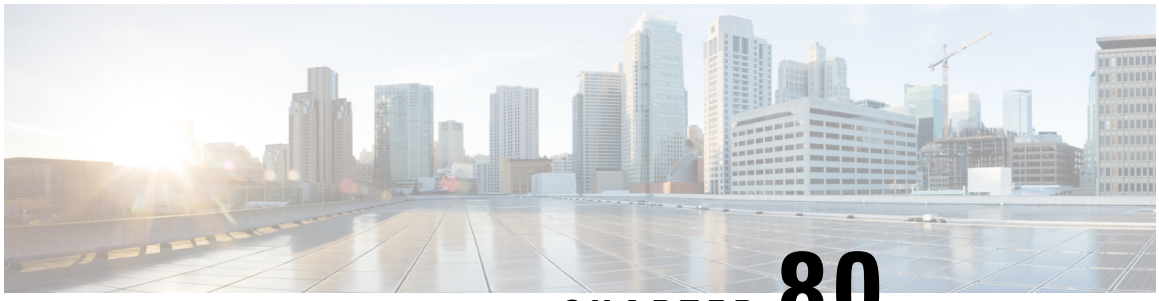
Standards & RFCs

Standard/RFC	Title
RFC 826	*An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses*
RFC 903	*A Reverse Address Resolution Protocol*
RFC 1027	*Using ARP to Implement Transparent Subnet Gateways*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for VRF-Aware Dynamic NAT Mapping with HSRP



CHAPTER 80

Configuring Stateful Interchassis Redundancy

The Stateful Interchassis Redundancy feature enables you to configure pairs of devices to act as backups for each other.

This module describes conceptual information about and tasks for configuring stateful interchassis redundancy.

Prerequisites for Stateful Interchassis Redundancy

All application redundancy configurations, including Network Address Translation (NAT) rules that have redundancy group associations and mapping IDs, must be identical on both devices, or NAT sessions will not be synchronized between devices and NAT redundancy will not work.

Restrictions for Stateful Interchassis Redundancy

- By default, Network Address Translation (NAT) high availability (inter and intrabox) does not replicate HTTP sessions to the standby device. To replicate HTTP sessions on the standby device during a switchover, you must configure the **ip nat switchover replication http** command.
- During NAT payload translations with certain applications, there can be IP addresses in the payload that require NAT translation. The application-level gateway (ALG) for that specific application parses the packet for these IP addresses, NAT translates these addresses, and the ALG writes the translated addresses back into the packet.

  Fixup denotes the writing of the translated IP address back into the packet. The write back of data can change the length of a packet, which results in the adjustment of the packet's TCP sequence (SEQ) or acknowledgment (ACK) values by NAT for the life of the TCP connection. NAT writes the new TCP SEQ/ACK values into the packet during SEQ/ACK fixup.

  For example, during a TCP ALG session, SEQ/ACK values may require fixup with mainly ASCII applications such as Domain Name System (DNS), FTP/FTP64, H.323, Real Time Streaming Protocol

(RTSP), and Session Initiation Protocol (SIP). This SEQ/ACK adjustment information gets associated with the NAT session and is synchronized to the standby device periodically.

During a stateful switchover, if the SEQ/ACK information is not completely synchronized to the new active device it is likely that the TCP connection would be reset by endpoints of the application.

- Stateful interchassis redundancy cannot coexist with intrachassis redundancy, including software redundancy.
- In Service Software Upgrade (ISSU) is not supported.
- When changing the paired-address-pooling, bulk port-allocation, or NAT mode settings the following steps must be followed:
  1. Shutdown the redundancy group and NAT interfaces on the standby device using the **shutdown** command. Clear NAT sessions on the standby device after shutting down the redundancy group.
  2. Change the paired-address-pooling, bulk port-allocation, or NAT mode on the standby device first and then on the active device.
  3. Configure the **no shutdown** command for the redundancy group and NAT interfaces on the standby device.
- In a NAT Stateful Interchassis Redundancy configuration, it is mandatory that both peers use the same inside and outside NAT interfaces. If the interfaces are not same, it can lead to duplicate NAT entries.
- The following translations are not synchronized to the standby router :
  - Translations created based on an interface overload rule
  - ICMP requests

Note For a standalone NAT router, shut down the NAT interfaces before you make a configuration change.

Information About Stateful Interchassis Redundancy

Stateful Interchassis Redundancy Overview

You can configure the Stateful Interchassis Redundancy feature to determine the active device from a group of devices, based on a number of failover conditions. When a failover occurs, the standby device seamlessly takes over, starts performing traffic forwarding services, and maintains a dynamic routing table.

Note Manually shutting down the control or data interface link on an active NAT router results in traffic outage as the NAT router never transitions to active state.

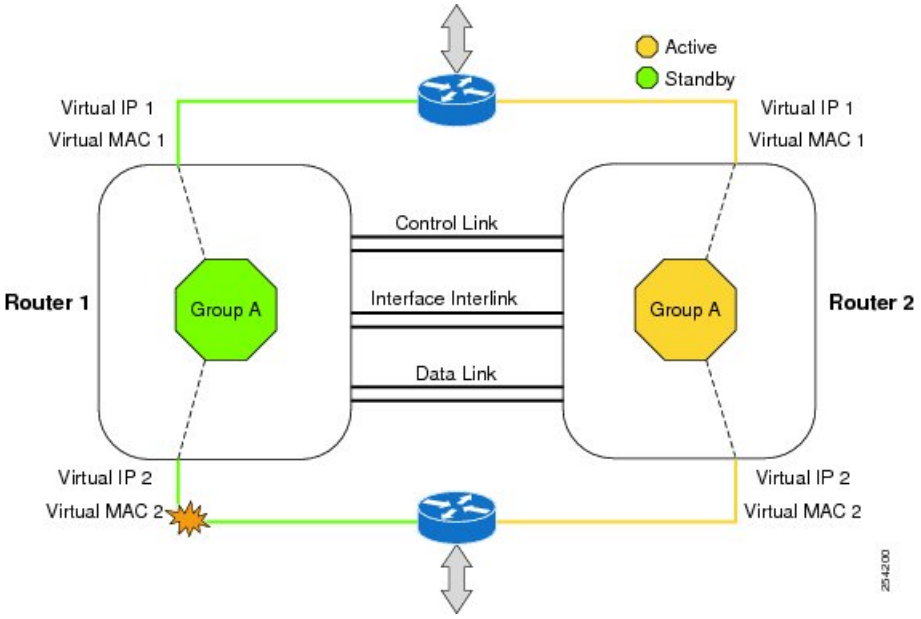
Stateful Interchassis Redundancy Operation

You can configure pairs of devices to act as hot standbys for each other. Redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups (RGs). Redundancy occurs at an application level and does not require a complete physical failure of the interface or device for a switchover of the application to occur. When a switchover occurs, the application activity continues to run seamlessly on the redundant interface.

The figure below depicts an active/standby load-sharing scenario. The figure shows how an RG is configured for a pair of devices that has one outgoing interface. Group A on Router 1 is the active RG and Group A on Router 2 is the standby RG.

Redundant devices are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of devices. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and synchronize the stateful database. The pairs of redundant interfaces are configured with the same unique ID number known as the redundant interface identifier (RII).

Figure 85: Redundancy Group Configuration—One Outgoing Interface



The status of redundancy group members is determined through the use of hello messages sent over the control link. The software considers either device not responding to a hello message within a configurable amount of time to be a failure and initiates a switchover. For the software to detect a failure in milliseconds, control links run the failover protocol that is integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for hello messages:

- Hello time—Interval at which hello messages are sent.
- Hold time—Amount of time before which the active or standby device is declared to be down.

The hello time defaults to 3 seconds to align with the Hot Standby Router Protocol (HSRP), and the hold time defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine the pairs of interfaces that are affected by the switchover, you must configure a unique ID for each pair of redundant interfaces. This ID is known as the RII that is associated with the interface.

A switchover to the standby device can occur when the priority setting that is configured on each device changes. The device with the highest priority value acts as the active device. If a fault occurs on either the active or standby device, the priority of the device is decremented by a configurable amount known as the weight. If the priority of the active device falls below the priority of the standby device, a switchover occurs and the standby device becomes the active device. This default behavior can be overridden by disabling the preemption attribute for the RG. You can also configure each interface to decrease the priority when the Layer 1 state of the interface goes down. The priority that is configured overrides the default priority of an RG.

Each failure event that causes a modification of an RG priority generates a syslog entry that contains a time stamp, the RG that was affected, the previous priority, the new priority, and a description of the failure event cause.

A switchover also can occur when the priority of a device or interface falls below a configurable threshold level.

A switchover to the standby device occurs under the following circumstances:

- Power loss or a reload occurs on the active device (including reloads).
- The run-time priority of the active device goes below that of the standby device (with preempt configured).
- The run-time priority of the active device goes below that of the configured threshold.
- The redundancy group on the active device is reloaded manually. Use the **redundancy application reload group** *rg-number* command for a manual reload.

Associations with Firewalls and NAT

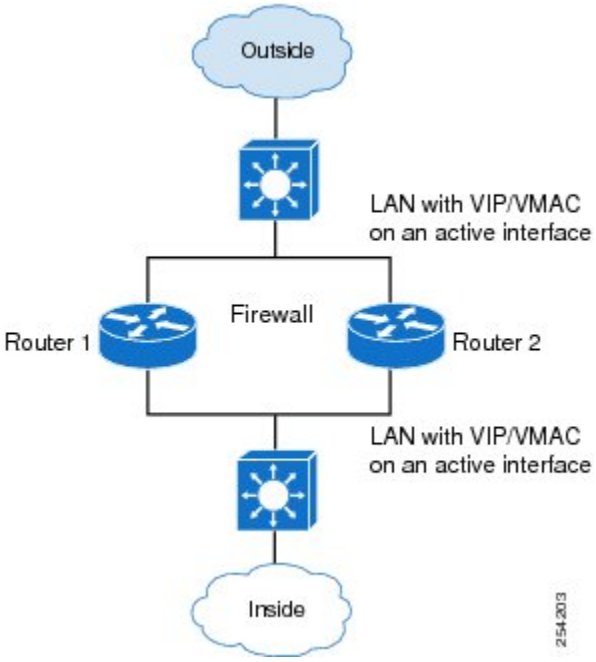
Firewalls use the association of the redundancy group with a traffic interface.

Network Address Translation (NAT) associates the redundancy group with a mapping ID.

LAN-LAN Topology

The figure below shows the LAN-LAN topology. In a LAN-LAN topology, all participating devices are connected to each other through LAN interfaces on both the inside and the outside. In this scenario, traffic is often directed to the correct firewall if static routing is configured on the upstream or downstream devices to an appropriate virtual IP address. This plaform participate in dynamic routing with upstream or downstream devices. The dynamic routing configuration supported on LAN-facing interfaces must not introduce a dependency on the routing protocol convergence; otherwise, fast failover requirements will not be met.

Figure 86: LAN-LAN Topology



How to Configure Stateful Interchassis Redundancy

Configuring the Control Interface Protocol

The configuration for the control interface protocol consists of the following elements:

- Authentication information
- Group name
- Hello time
- Hold time
- Protocol instance
- Use of the bidirectional forwarding direction (BFD) protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **mode none**
5. **application redundancy**
6. **protocol** *number*

7. **name** *instance-name*
8. **timers hellotime** [**msec**] *number* **holdtime** [**msec**] *number*
9. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* | **md5 key-chain** *key-chain-name*}
10. **bfd**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**mode none** **Example:** `Device(config-red)# mode none`	Sets the redundancy mode to none, which is required for this feature.
Step 5	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 6	**protocol** *number* **Example:** `Device(config-red-app)# protocol 4`	Specifies the protocol instance that will be attached to a control interface, and enters redundancy application protocol configuration mode.
Step 7	**name** *instance-name* **Example:** `Device(config-red-app-prot)# name rg1`	(Optional) Specifies an optional alias for the protocol instance.
Step 8	**timers hellotime** [**msec**] *number* **holdtime** [**msec**] *number* **Example:** `Device(config-red-app-prot)# timers hellotime 3 holdtime 10`	Specifies the interval between hello messages sent and the time before a device is declared to be down. • The default time for hello time is 3 seconds and for hold time is 10 seconds.
Step 9	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* | **md5 key-chain** *key-chain-name*} **Example:**	Specifies authentication information.

	Command or Action	Purpose
	`Device(config-red-app-prot)# authentication text password`	
Step 10	**bfd** **Example:** `Device(config-red-app-prot)# bfd`	(Optional) Enables the integration of the failover protocol running on the control interface with the BFD protocol to achieve failure detection in milliseconds. • BFD is enabled by default.
Step 11	**end** **Example:** `Device(config-red-app-prot)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Group

Redundancy groups consist of the following configuration elements:

- The amount by which the priority will be decremented for each object.
- Faults (objects) that will decrement the priority.
- Failover priority.
- Failover threshold.
- Group instance.
- Group name.
- Initialization delay timer.
- The interface that is associated with the redundancy group (RG).
- The interface that is used as the control interface.
- The interface that is used as the data interface.
- The redundancy interface identifier (RII) number of the RG interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** {**1** | **2**}
6. **name** *group-name*
7. **preempt**
8. **priority** *number* **failover-threshold** *number*
9. **track** *object-number* [**decrement** *number* | **shutdown**]
10. **timers delay** *seconds* [**reload** *seconds*]

11. **control** *interface-name* **protocol** *instance*
12. **data** *interface-name*
13. To create another redundancy group, repeat Steps 3 through 12.
14. **end**
15. **configure terminal**
16. **interface** *type number*
17. **redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*]
18. **redundancy rii** *number*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**group** {**1** | **2**} **Example:** `Device(config-red-app)# group 1`	Specifies the redundancy group instance and enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name rg1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the group and enables the standby device to preempt the active device regardless of which device has higher priority.
Step 8	**priority** *number* **failover-threshold** *number* **Example:** `Device(config-red-app-grp)# priority 120 failover-threshold 80`	Specifies the initial priority and failover threshold for the redundancy group.

	Command or Action	Purpose
Step 9	**track** *object-number* [**decrement** *number* | **shutdown**] **Example:** `Device(config-red-app-grp)# track 44 decrement 20`	Specifies the amount by which the priority of a redundancy group will be decremented if an event occurs. • You can track multiple objects that influence the priority of the redundancy group.
Step 10	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 10 reload 20`	Specifies the amount of time by which the redundancy group will delay role negotiations that start after a fault occurs or after the system is reloaded.
Step 11	**control** *interface-name* **protocol** *instance* **Example:** `Device(config-red-app-grp)# control GigabitEthernet0/1/0 protocol 1`	Specifies the control interface that is used by the redundancy group. • This interface is also associated with an instance of the control interface protocol.
Step 12	**data** *interface-name* **Example:** `Device(config-red-app-grp)# data GigabitEthernet0/1/2`	Specifies the data interface that is used by the redundancy group.
Step 13	To create another redundancy group, repeat Steps 3 through 12.	—
Step 14	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.
Step 15	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 16	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Selects an interface to associate with the redundancy group and enters interface configuration mode.
Step 17	**redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 ip 10.10.1.1 exclusive decrement 20`	Associates the interface with the redundancy group identified by the *number* argument.
Step 18	**redundancy rii** *number* **Example:** `Device(config-if)# redundancy rii 40`	Specifies a number for the RII associated with this interface. • This number must match the RII of the other interface in the redundancy group.

	Command or Action	Purpose
Step 19	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring a Redundant Traffic Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat outside**
6. **ip virtual-reassembly**
7. **negotiation auto**
8. **redundancy rii** *number*
9. **redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/5`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Configures the outside interface for IP address translation.

	Command or Action	Purpose
Step 6	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables Virtual Fragmentation Reassembly (VFR) on an interface.
Step 7	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 8	**redundancy rii** *number* **Example:** `Device(config-if)# redundancy rii 200`	Specifies a number for the redundancy interface identifier (RII) that is associated with this interface. • This number must match the RII of the other interface in the redundancy group.
Step 9	**redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 ip 10.1.1.200 exclusive decrement 10`	Associates the interface with the redundancy group identified by the *number* argument.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring NAT with Stateful Interchassis Redundancy

You must use a mapping ID to associate Network Address Translation (NAT) with a redundancy group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **ip nat inside source list** {{*access-list-number* | *access-list-name*} | **route-map** *name*} **pool** *name* [**redundancy** *redundancy-id* [**mapping-id** *map-id* | **overload** | **reversible** | **vrf** *name*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool VPN-18 10.10.0.0 10.10.255.255 netmask 255.255.0.0`	Defines a pool of IP addresses for NAT.
Step 4	**ip nat inside source list** {{*access-list-number* | *access-list-name*} | **route-map** *name*} **pool** *name* [**redundancy** *redundancy-id* [**mapping-id** *map-id* | **overload** | **reversible** | **vrf** *name*]] **Example:** `Device(config)# ip nat inside source list acl-18 pool VPN-18 redundancy 2 mapping-id 152`	Enables NAT of the inside source address. • You must use a mapping ID to associate NAT with the redundancy group.
Step 5	**end** **Example:** `Device(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Managing and Monitoring Stateful Interchassis Redundancy

All configuration commands in this task are optional. You can use the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **redundancy application reload group** *number* [**peer** | **self**]
3. **show redundancy application group** [*group-id* | **all**]
4. **show redundancy application transport** {**clients** | **group** [*group-id*]}
5. **show redundancy application protocol** {*protocol-id* | **group** [*group-id*]}
6. **show redundancy application faults group** [*group-id*]
7. **show redundancy application if-mgr group** [*group-id*]
8. **show redundancy application control-interface group** [*group-id*]
9. **show redundancy application data-interface group** [*group-id*]
10. **show monitor event-trace rg_infra all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**redundancy application reload group** *number* [**peer** | **self**] **Example:** `Device# redundancy application reload group 2 self`	Forces the active redundancy group (RG) to reload and the standby RG to become the active RG. • Use the **redundancy application reload** command to verify if the redundancy configuration is working. You must enter this command on the active RG.
Step 3	**show redundancy application group** [*group-id* | **all**] **Example:** `Device# show redundancy application group 2`	Displays summary information for the specified group or for all groups.
Step 4	**show redundancy application transport** {**clients** | **group** [*group-id*]} **Example:** `Device# show redundancy application transport group 2`	Displays transport information for the specified group or for all groups.
Step 5	**show redundancy application protocol** {*protocol-id* | **group** [*group-id*]} **Example:** `Device# show redundancy application protocol 2`	Displays protocol information for the specified group or for all groups.
Step 6	**show redundancy application faults group** [*group-id*] **Example:** `Device# show redundancy application faults group 2`	Displays information about faults for the specified group or for all groups.
Step 7	**show redundancy application if-mgr group** [*group-id*] **Example:** `Device# show redundancy application if-mgr group 2`	Displays information about the interface manager (if-mgr) for the specified group or for all groups.
Step 8	**show redundancy application control-interface group** [*group-id*] **Example:** `Device# show redundancy application control-interface group IF-2`	Displays interface information associated with redundancy groups for the specified control interface.
Step 9	**show redundancy application data-interface group** [*group-id*] **Example:** `Device# show redundancy application data-interface group IF-2`	Displays interface information associated with redundancy groups for the specified data interface.
Step 10	**show monitor event-trace rg_infra all** **Example:**	Displays event trace information associated with all redundancy groups.

	Command or Action	Purpose
	Device# show monitor event-trace rg_infra all	

Configuration Examples for Stateful Interchassis Redundancy

Example: Configuring the Control Interface Protocol

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# mode none
Device(config-red)# application redundancy
Device(config-red-app)# protocol 4
Device(config-red-app-prot)# name rg1
Device(config-red-app-prot)# timers hellotime 3 holdtime 10
Device(config-red-app-prot)# authentication text password
Device(config-red-app-prot)# bfd
```

Example: Configuring a Redundancy Group

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name rg1
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# priority 120 failover-threshold 80
Device(config-red-app-grp)# track 44 decrement 20
Device(config-red-app-grp)# timers delay 10 reload 20
Device(config-red-app-grp)# control GigabitEthernet0/1/0 protocol 1
Device(config-red-app-grp)# data GigabitEthernet0/1/2
Device(config-red-app-grp)# end
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# redundancy group 1 ip 10.10.1.1 exclusive decrement 20
Device(config-if)# redundancy rii 40
```

Example: Configuring a Redundant Traffic Interface

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/5
Device(config-if)# ip address 10.1.1.2 255.0.0.0
Device(config-if)# ip nat outside
Device(config-if)# ip virtual-reassembly
Device(config-if)# negotiation auto
Device(config-if)# redundancy rii 200
Device(config-if)# redundancy group 1 ip 10.1.1.200 exclusive decrement 10
```

Example: Configuring NAT with Stateful Interchassis Redundancy

```
Device# configure terminal
Device(config)# ip nat pool VPN-18 10.10.0.0 10.10.255.255 netmask 255.255.0.0
Device(config)# ip nat inside source list acl-18 pool VPN-18 redundancy 2 mapping-id 152
```

Additional References for Stateful Interchassis Redundancy

Related Documents

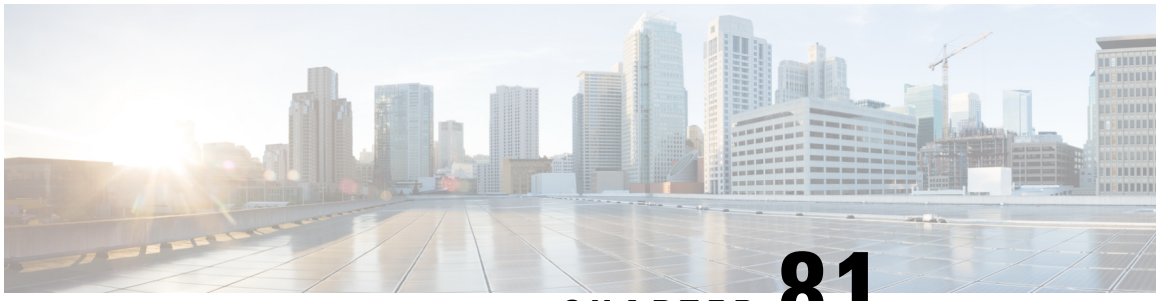
Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IP addressing commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
Fundamental principles of IP addressing and IP routing	*IP Routing Primer*

Standards and RFCs

Standards/RFCs	Title
RFC 791	Internet Protocol
RFC 1338	Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
RFC 1466	Guidelines for Management of IP Address Space
RFC 1716	Towards Requirements for IP Routers
RFC 1918	Address Allocation for Private Internets
RFC 3330	Special-Use IP Addresses

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 81

Mapping of Address and Port Using Encapsulation

The MAP-E feature provides rules to define the mapping between an IPv6 prefix and an IPv4 address or between a shared IPv4 address and an IPv6 prefix/address. The MAP-E feature is supported by the Stateless NAT64 feature and does not change the system flow of the NAT64 client.

Feature Information for Mapping of Address and Port Using Encapsulation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 121: Feature Information

Feature Name	Releases	Feature Information
MAP-E	Cisco IOS XE Amsterdam 17.2.1	The MAP-E feature provides support for configurable rules used to define the mapping between an IPv6 prefix and an IPv4 address or between a shared IPv4 address and an IPv6 prefix/address. The following commands were introduced or modified: **basic-mapping-rule**, **default-mapping-rule**, **nat64 map-e**, **port-parameters**, **show nat64 map-e**.

Restrictions for Mapping of Address and Port Using Encapsulation

- The MAP-E feature supports only a single basic mapping rule (BMR) per IPv6 prefix. This requires you to configure different mapping rules for every address and port translation.
- Default mapping rule (DMR) with 128 prefix must be configured before starting the MAP-E BMR configuration.
- This feature does not support BMR prefix length of 64, fragmentation, and local packet generation.

Information About Mapping of Address and Port Using Encapsulation

Mapping of Address and Port Using Encapsulation

MAP-E refers to Mapping of Address and Port Encapsulation (MAP-E). The MAP-E feature enables you to configure mapping rules for translation between IPv4 and IPv6 addresses. Each mapping of address and port using MAP-E domain uses a different mapping rule. A MAP-E configuration comprises of one basic mapping rule (BMR), one default mapping rule (DMR), and one or more forwarding mapping rules (FMRs) for each MAP-E domain.

A BMR configures the MAP IPv6 address or prefix. You can configure only one BMR per IPv6 prefix. The MAP-E CE uses the BMR to configure itself with an IPv4 address, an IPv4 prefix, or a shared IPv4 address from an IPv6 prefix. A BMR can also be used for forwarding packets in such scenarios where an IPv4 source address and source port are mapped into an IPv6 address/prefix. Every MAP-E node (CE device is a MAP-E node) must be provisioned with a BMR. The BMR prefix along with the port parameter is used as tunnel source address. You can use the **port-parameters** command to configure port parameters for the MAP-E BMR.

A DMR prefix which matches with the interface address is recognized as hosts and a DMR prefix with a prefix length of 128 is recognized as the tunnel source address. A border relay IPv6 address is used as the tunnel destination address.

How to Configure Mapping of Address Port Using Encapsulation

Configuring Mapping of Address and Port Using Encapsulation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **nat64 map-e domain** *number*

4. **border-relay-address***br-ipv6-address*
5. **basic-mapping-rule**
6. **ipv4-prefix** *ipv4-prefix/length*
7. **ipv6-prefix** *ipv6-prefix/length*
8. **port-parameters share-ratio** *number* **port-offet-bits***number***start-port***port-number***no-eabits***number*
9. **port-set-id***number*
10. **exit**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**nat64 map-e domain** *number* **Example:** `Device(config)# nat64 map-e domain 01`	Specifies the nat64 MAP-E domain and enters the MAP-E configuration mode. • The range is from 1 to 128.
Step 4	**border-relay-address***br-ipv6-address* **Example:** `Device(config)# border-relay-address 2002:DB8::9`	Specifies the IPv6 address of the border relay router.
Step 5	**basic-mapping-rule** **Example:** `Device(config-nat64-mape)# basic-mapping-rule`	Specifies the MAP-E mapping rule and enters the basic mapping rule configuration mode.
Step 6	**ipv4-prefix** *ipv4-prefix/length* **Example:** `Device(config-nat64-mape-bmr)# ipv4-prefix 10.1.1.0/24`	Specifies the IPv4 prefix and length for translation.
Step 7	**ipv6-prefix** *ipv6-prefix/length* **Example:** `Device(config-nat64-mape-bmr)# ipv6-prefix 2001:100::0/64`	Specifies the IPv6 prefix and length for translation.
Step 8	**port-parameters share-ratio** *number* **port-offet-bits***number***start-port***port-number***no-eabits***number* **Example:**	Specifies the values for port-parameters share-ratio, contiguous ports and start-port for MAP-E Basic Mapping Rule (BMR).

	Command or Action	Purpose
	`Device(config-nat64-mape-bmr)# port-parameters share-ratio 2 port-offset-bits 5 start-port 1024`	• If the share ratio is greater than 1, the configuration throws an error if the startport value is incorrect. The calculation is based on the share-ratio and port-offset bits. The configuration throws error and displays the value to be configured. • If the share ratio is 1, there are no port-offset bits as the values is automatically set to 6 and the start port is set to 1024.
Step 9	**port-set-id***number* **Example:** `Device(config-nat64-mape-bmr)# port-set-id 1`	Specifies the port-set identifier.
Step 10	**exit** **Example:** `Device(config-nat64-mape-bmr)# exit`	Exits basic mapping rule configuration mode and returns to MAP-E configuration mode.
Step 11	**end** **Example:** `Device(config)# end`	Exits MAP-E configuration mode and returns to privileged EXEC mode.

Verifying Mapping of Address and Port Using Encapsulation Configuration

SUMMARY STEPS

1. **enable**
2. **show nat64 MAP-E** [*domain number*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show nat64 MAP-E** [*domain number*]

Example:

```
Device# show nat64 MAP-E domain 1
  MAP-E Domain 1
   Mode MAP-E
   Default-mapping-rule
      Ip-v6-prefix 2001:22::/128
```

```
Basic-mapping-rule
   Ip-v6-prefix 2001:100::/64
   Ip-v4-prefix 10.1.1.0/24
   Port-parameters
      Share-ratio 2   Contiguous-ports 1024   Start-port 1024
      Share-ratio-bits 1   Contiguous-ports-bits 10   Port-offset-bits 5
```

Displays MAP-E configuration.

Configuration Examples for Mapping of Address and Port Using Encapsulation

Example: Mapping of Address and Port Using Encapsulation

The following example shows how to configure MAP-E:

```
!
ipv6 unicast-routing
!
interface GigabitEthernet0/1/0
switchport access vlan 10
!
interface GigabitEthernet0/1/1
switchport access vlan 10
!
interface GigabitEthernet0/1/2
switchport access vlan 11
!
interface GigabitEthernet0/1/3
switchport access vlan 11
!
!
interface Vlan10
ip address 10.0.0.1 255.255.255.0
nat64 enable
ip virtual-reassembly
!
interface Vlan11
no ip address
ipv6 address 2001:DB8:0001:80:0:CBC8:34:1/64
ipv6 enable
ipv6 virtual-reassembly in
nat64 enable
!
ip nat pool p3 209.165.200.225 209.165.200.225 netmask 255.255.255.224
ip nat inside source route-map rm1 pool p3 overload
!
!
ip access-list extended inside-local
20 permit ip 10.10.0.0 255.255.0.0 any
!
ipv6 route ::/0 2001:DB8:0001:80::9
!
!
route-map rm1 permit 10
```

```
match ip address inside-local
!
!
nat64 settings fragmentation header disable
nat64 route 0.0.0.0/0 Vlan11
nat64 map-e domain 1
border-relay-address 2001:DB8::9
basic-mapping-rule
ipv6-prefix 2001:DB8:0001::/56
ipv4-prefix 209.165.200.225/32
port-parameters share-ratio 2 start-port 4096
port-set-id 1
local-ipv4-prefix 192.168.0.0/16
!
!
!
!
!
end
```

Additional References for Mapping of Address and Port Using Encapsulation

Related Documents

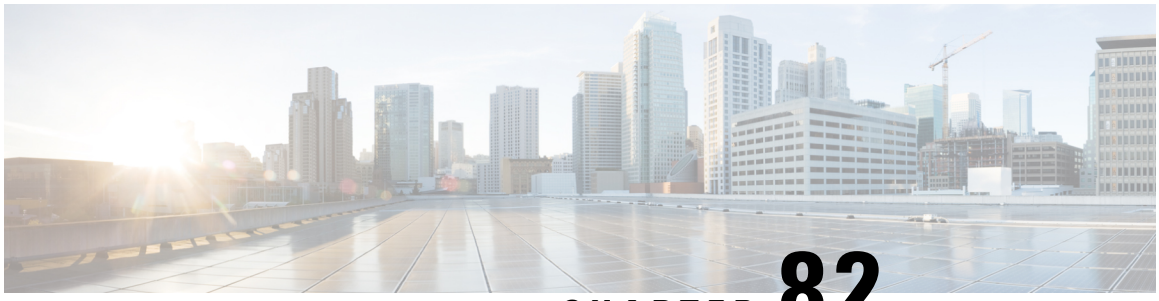
Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title
MAP	Mapping of Address and Port (MAP)
MAP Encapsulation	MAP Encapsulation (MAP-E) - specification
RFC 6052	IPv6 Addressing of IPv4/IPv6 Translators
RFC 6144	Framework for IPv4/IPv6 Translation
RFC 6145	IP/ICMP Translation Algorithm

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 82

Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT feature supports the forwarding of packets from a standby redundancy group to the active redundancy group for packet handling. If this feature is not enabled, the return TCP packets forwarded to the router that did not receive the initial synchronization (SYN) message are dropped because they do not belong to any known existing session.

This module provides an overview of asymmetric routing and describes how to configure asymmetric routing

Restrictions for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The following restrictions apply to the Interchassis Asymmetric Routing Support feature:

- LANs that use virtual IP addresses and virtual MAC (VMAC) addresses do not support asymmetric routing.
- In Service Software Upgrade (ISSU) is not supported.

The following features are not supported by the VRF-Aware Asymmetric Routing Support feature:

- Cisco Trustsec
- Edge switching services
- Header compression

- IPsec
- Policy Based Routing (PBR)
- Port bundle
- Lawful intercept
- Layer 2 Tunneling Protocol (L2TP)
- Locator/ID Separation Protocol (LISP) inner packet inspection
- Secure Shell (SSL) VPN
- Session Border Controller (SBC)

Information About Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

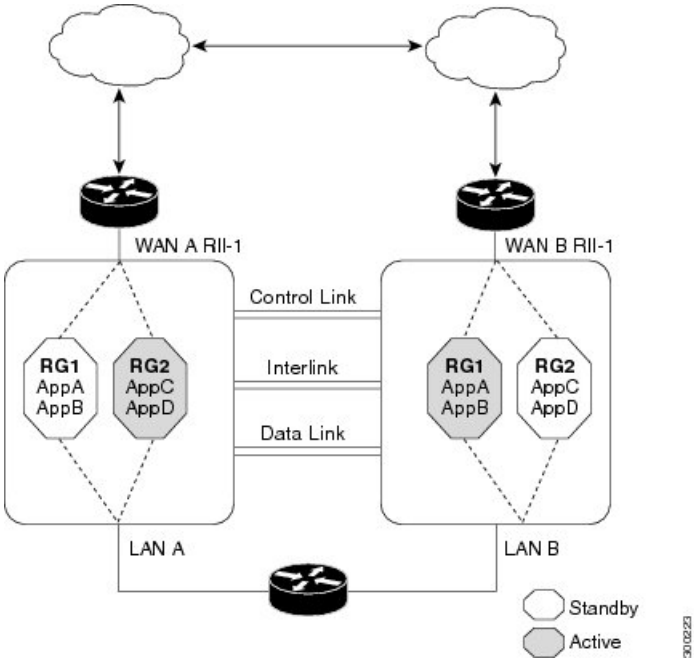
Asymmetric Routing Overview

Asymmetric routing occurs when packets from TCP or UDP connections flow in different directions through different routes. In asymmetric routing, packets that belong to a single TCP or UDP connection are forwarded through one interface in a redundancy group (RG), but returned through another interface in the same RG. In asymmetric routing, the packet flow remains in the same RG. When you configure asymmetric routing, packets received on the standby RG are redirected to the active RG for processing. If asymmetric routing is not configured, the packets received on the standby RG may be dropped.

Asymmetric routing determines the RG for a particular traffic flow. The state of the RG is critical in determining the handling of packets. If an RG is active, normal packet processing is performed. In case the RG is in a standby state and you have configured asymmetric routing and the **asymmetric-routing always-divert enable** command, packets are diverted to the active RG. Use the **asymmetric-routing always-divert enable** command to always divert packets received from the standby RG to the active RG.

The figure below shows an asymmetric routing scenario with a separate asymmetric-routing interlink interface to divert packets to the active RG.

Figure 87: Asymmetric Routing Scenario



The following rules apply to asymmetric routing:

- 1:1 mapping exists between the redundancy interface identifier (RII) and the interface.
- 1:*n* mapping exists between the interface and an RG. (An asymmetric routing interface can receive traffic from and send traffic to multiple RGs. For a non asymmetric-routing interface (normal LAN interface), a 1:1 mapping exists between the interface and the RG.)
- 1:*n* mapping exists between an RG and applications that use it. (Multiple applications can use the same RG).
- 1:1 mapping exists between an RG and the traffic flow. The traffic flow must map only to a single RG. If a traffic flow maps to multiple RGs, an error occurs.
- 1:1 or 1:*n* mapping can exist between an RG and an asymmetric-routing interlink as long as the interlink has sufficient bandwidth to support all the RG interlink traffic.

Asymmetric routing consists of an interlink interface that handles all traffic that is to be diverted. The bandwidth of the asymmetric-routing interlink interface must be large enough to handle all expected traffic that is to be diverted. An IPv4 address must be configured on the asymmetric-routing interlink interface, and the IP address of the asymmetric routing interface must be reachable from this interface.

Note We recommend that the asymmetric-routing interlink interface be used for interlink traffic only and not be shared with high availability control or data interfaces because the amount of traffic on the asymmetric-routing interlink interface could be quite high.

Asymmetric Routing Support in Firewalls

For intrabox asymmetric routing support, the firewall does a stateful Layer 3 and Layer 4 inspection of Internet Control Message Protocol (ICMP), TCP, and UDP packets. The firewall does a stateful inspection of TCP packets by verifying the window size and order of packets. The firewall also requires the state information from both directions of the traffic for stateful inspection. The firewall does a limited inspection of ICMP information flows. It verifies the sequence number associated with the ICMP echo request and response. The firewall does not synchronize any packet flows to the standby redundancy group (RG) until a session is established for that packet. An established session is a three-way handshake for TCP, the second packet for UDP, and informational messages for ICMP. All ICMP flows are sent to the active RG.

The firewall does a stateless verification of policies for packets that do not belong to the ICMP, TCP, and UDP protocols.

The firewall depends on bidirectional traffic to determine when a packet flow should be aged out and diverts all inspected packet flows to the active RG. Packet flows that have a pass policy and that include the same zone with no policy or a drop policy are not diverted.

Note The firewall does not support the **asymmetric-routing always-divert enable** command that diverts packets received on the standby RG to the active RG. By default, the firewall forces all packet flows to be diverted to the active RG.

Asymmetric Routing in NAT

By default, when asymmetric routing is configured, Network Address Translation (NAT) processes non-ALG packets on the standby RG, instead of forwarding them to the active. The NAT-only configuration (that is when the firewall is not configured) can use both the active and standby RGs for processing packets. If you have a NAT-only configuration and you have configured asymmetric routing, the default asymmetric routing rule is that NAT will selectively process packets on the standby RG. You can configure the **asymmetric-routing always-divert enable** command to divert packets received on the standby RG to the active RG. Alternatively, if you have configured the firewall along with NAT, the default asymmetric routing rule is to always divert the packets to the active RG.

When NAT receives a packet on the standby RG and if you have not configured the diverting of packets, NAT does a lookup to see if a session exists for that packet. If a session exists and there is no ALG associated for that session, NAT processes the packet on the standby RG. The processing of packets on the standby RG when a session exists significantly increases the bandwidth of the NAT traffic.

ALGs are used by NAT to identify and translate payload and to create child flows. ALGs require a two-way traffic to function correctly. NAT must divert all traffic to the active RG for any packet flow that is associated with an ALG. This is accomplished by checking if ALG data that is associated with the session is found on the standby RG. If ALG data exits, the packet is diverted for asymmetric routing.

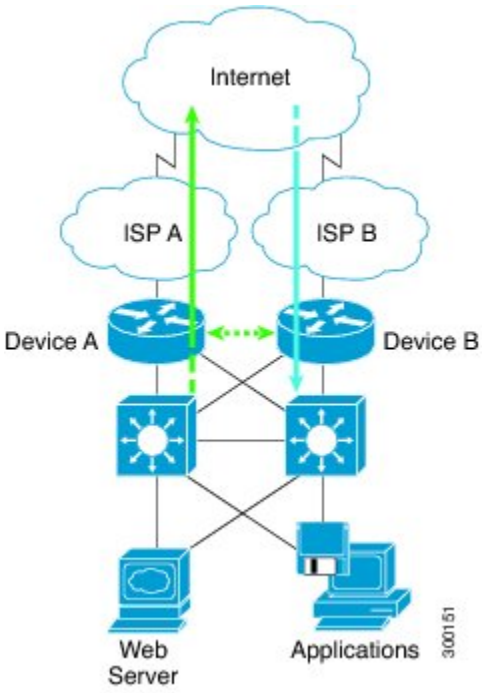
VRF-Aware Software Infrastructure (VASI) support was added in Cisco IOS XE Release 3.16S. Multiprotocol Label Switching (MPLS) asymmetric routing is also supported.

In Cisco IOS XE Release 3.16S, NAT supports asymmetric routing with ALGs, Carrier Grade NAT (CGN), and virtual routing and forwarding (VRF) instances. No configuration changes are required to enable asymmetric routing with ALGs, CGN, or VRF. For more information, see the section, “Example: Configuring Asymmetric Routing with VRF”.

Asymmetric Routing in a WAN-LAN Topology

Asymmetric routing supports only a WAN-LAN topology. In a WAN-LAN topology, devices are connected through LAN interfaces on the inside and WAN interfaces on the outside. There is no control on the routing of return traffic received through WAN links. Asymmetric routing controls the routing of return traffic received through WAN links in a WAN-LAN topology. The figure below shows a WAN-LAN topology.

Figure 88: Asymmetric Routing in a WAN-LAN Topology



VRF-Aware Asymmetric Routing in Zone-Based Firewalls

In Cisco IOS XE Release 3.14S, zone-based firewalls support the VRF-Aware Interchassis Asymmetric Routing feature. The feature supports Multiprotocol Label Switching (MPLS).

During asymmetric routing diversion, the VPN routing and forwarding (VRF) name hash value is sent with diverted packets. The VRF name hash value is converted to the local VRF ID and table ID at the active device after the diversion.

When diverted packets reach the active device on which Network Address Translation (NAT) and the zone-based firewall are configured, the firewall retrieves the VRF ID from NAT or NAT64 and saves the VRF ID in the firewall session key.

The following section describes the asymmetric routing packet flow when only the zone-based firewall is configured on a device:

- When MPLS is configured on a device, the VRF ID handling for diverted packets is the same as the handling of non-asymmetric routing diverted packets. An MPLS packet is diverted to the active device, even though the MPLS label is removed at the standby device. The zone-based firewall inspects the packet at the egress interface, and the egress VRF ID is set to zero, if MPLS is detected at this interface. The firewall sets the ingress VRF ID to zero if MPLS is configured at the ingress interface.

- When a Multiprotocol Label Switching (MPLS) packet is diverted to the active device from the standby device, the MPLS label is removed before the asymmetric routing diversion happens.
- When MPLS is not configured on a device, an IP packet is diverted to the active device and the VRF ID is set. The firewall gets the local VRF ID, when it inspects the packet at the egress interface.

VRF mapping between active and standby devices require no configuration changes.

VRF-Aware Asymmetric Routing in NAT

In Cisco IOS XE Release 3.14S, Network Address Translation supports VRF-aware interchassis asymmetric routing. VRF-aware interchassis asymmetric routing uses message digest (MD) 5 hash of the VPN routing and forwarding (VRF) name to identify the VRF and datapath in the active and standby devices to retrieve the local VRF ID from the VRF name hash and viceversa.

For VRF-aware interchassis asymmetric routing, the VRFs on active and standby devices must have the same VRF name. However, the VRF ID need not be identical on both devices because the VRF ID is mapped based on the VRF name on the standby and active devices during asymmetric routing diversion or box-to-box high availability synchronization.

In case of MD5 hash collision for VRF names, the firewall and NAT sessions that belong to the VRF are not synced to the standby device.

VRF mapping between active and standby devices require no configuration changes.

How to Configure Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Configuring a Redundancy Application Group and a Redundancy Group Protocol

Redundancy groups consist of the following configuration elements:

- The amount by which the priority will be decremented for each object.
- Faults (objects) that decrement the priority
- Failover priority
- Failover threshold
- Group instance
- Group name
- Initialization delay timer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**

4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **priority** *value* [**failover threshold** *value*]
8. **preempt**
9. **track** *object-number* **decrement** *number*
10. **exit**
11. **protocol** *id*
12. **timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*}
13. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*}
14. **bfd**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group and enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	Specifies an optional alias for the protocol instance.
Step 7	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	Specifies the initial priority and failover threshold for a redundancy group.

	Command or Action	Purpose
Step 8	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the redundancy group and enables the standby device to preempt the active device. • The standby device preempts only when its priority is higher than that of the active device.
Step 9	**track** *object-number* **decrement** *number* **Example:** `Device(config-red-app-grp)# track 50 decrement 50`	Specifies the priority value of a redundancy group that will be decremented if an event occurs on the tracked object.
Step 10	**exit** **Example:** `Device(config-red-app-grp)# exit`	Exits redundancy application group configuration mode and enters redundancy application configuration mode.
Step 11	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Specifies the protocol instance that will be attached to a control interface and enters redundancy application protocol configuration mode.
Step 12	**timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10`	Specifies the interval between hello messages sent and the time period before which a device is declared to be down. • Holdtime should be at least three times the hellotime.
Step 13	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies authentication information.
Step 14	**bfd** **Example:** `Device(config-red-app-prtcl)# bfd`	Enables the integration of the failover protocol running on the control interface with the Bidirectional Forwarding Detection (BFD) protocol to achieve failure detection in milliseconds. • BFD is enabled by default.
Step 15	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring Data, Control, and Asymmetric Routing Interfaces

In this task, you configure the following redundancy group (RG) elements:

- The interface that is used as the control interface.

- The interface that is used as the data interface.
- The interface that is used for asymmetric routing. This is an optional task. Perform this task only if you are configuring asymmetric routing for Network Address Translation (NAT).

Note Asymmetric routing, data, and control must be configured on separate interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **data** *interface-type interface-number*
7. **control** *interface-type interface-number* **protocol** *id*
8. **timers delay** *seconds* [**reload** *seconds*]
9. **asymmetric-routing interface** *type number*
10. **asymmetric-routing always-divert enable**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group (RG) and enters redundancy application group configuration mode.

	Command or Action	Purpose
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/1`	Specifies the data interface that is used by the RG.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control GigabitEthernet 1/0/0 protocol 1`	Specifies the control interface that is used by the RG. • The control interface is also associated with an instance of the control interface protocol.
Step 8	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 100 reload 400`	Specifies the time required for an RG to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**asymmetric-routing interface** *type number* **Example:** `Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1`	Specifies the asymmetric routing interface that is used by the RG.
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Always diverts packets received from the standby RG to the active RG.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

Note
- You must not configure a redundant interface identifier (RII) on an interface that is configured either as a data interface or as a control interface.
- You must configure the RII and asymmetric routing on both active and standby devices.
- You cannot enable asymmetric routing on the interface that has a virtual IP address configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **redundancy rii** *id*
5. **redundancy group** *id* [**decrement** *number*]
6. **redundancy asymmetric-routing enable**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/3`	Selects an interface to be associated with the redundancy group (RG) and enters interface configuration mode.
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 600`	Configures the redundancy interface identifier (RII).
Step 5	**redundancy group** *id* [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 decrement 20`	Enables the RG redundancy traffic interface configuration and specifies the amount to be decremented from the priority when the interface goes down. **Note** You need not configure an RG on the traffic interface on which asymmetric routing is enabled.
Step 6	**redundancy asymmetric-routing enable** **Example:** `Device(config-if)# redundancy asymmetric-routing enable`	Establishes an asymmetric flow diversion tunnel for each RG.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring Dynamic Inside Source Translation with Asymmetric Routing

The following configuration is a sample dynamic inside source translation with asymmetric routing. You can configure asymmetric routing with the following types of NAT configurations—dynamic outside source, static inside and outside source, and Port Address Translation (PAT) inside and outside source translations. For more information on different types of NAT configurations, see the "Configuring NAT for IP Address Conservation" chapter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat outside**
6. **exit**
7. **redundancy**
8. **application redundancy**
9. **group** *id*
10. **asymmetric-routing always-divert enable**
11. **end**
12. **configure terminal**
13. **ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*}
14. **exit**
15. **ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id*
16. **access-list** *standard-acl-number* **permit** *source-address wildcard-bits*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/3`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for an interface.
Step 5	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 6	**exit** **Example:**	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 7	**redundancy** **Example:** `Device(config)# redundancy`	Configures redundancy and enters redundancy configuration mode.
Step 8	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 9	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group and enters redundancy application group configuration mode.
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Diverts the traffic to the active device.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.
Step 12	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 13	**ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool pool1 prefix-length 24`	Defines a pool of global addresses. • Enters IP NAT pool configuration mode.
Step 14	**exit** **Example:** `Device(config-ipnat-pool)# exit`	Exits IP NAT pool configuration mode and enters global configuration mode.
Step 15	**ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id* **Example:** `Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100`	Enables NAT of the inside source address and associates NAT with a redundancy group by using the mapping ID.
Step 16	**access-list** *standard-acl-number* **permit** *source-address wildcard-bits* **Example:**	Defines a standard access list for the inside addresses that are to be translated.

	Command or Action	Purpose
	`Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0`	
Step 17	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Example: Configuring a Redundancy Application Group and a Redundancy Group Protocol

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 50 decrement 50
Device(config-red-app-grp)# exit
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end
```

Example: Configuring Data, Control, and Asymmetric Routing Interfaces

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/1
Device(config-red-app-grp)# control GigabitEthernet 1/0/0 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
```

Example: Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/3
Device(config-if)# redundancy rii 600
Device(config-if)# redundancy group 1 decrement 20
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# end
```

Example: Configuring Dynamic Inside Source Translation with Asymmetric Routing

```
Device(config)# interface gigabitethernet 0/1/3
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
Device# configure terminal
Device(config)# ip nat pool pool1 prefix-length 24
Device(config-ipnat-pool)# exit
Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100
Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0
```

Example: Configuring VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

The following is a sample WAN-to-WAN symmetric routing configuration:

```
vrf definition Mgmt-intf
 address-family ipv4
  exit-address-family
!
address-family ipv6
 exit-address-family
!
!
vrf definition VRFA
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 address-family ipv4
  exit-address-family
!
!
no logging console
no aaa new-model
```

```
!
multilink bundle-name authenticated
!
redundancy
 mode sso
 application redundancy
  group 1
   preempt
   priority 120
   control GigabitEthernet 0/0/1 protocol 1
   data GigabitEthernet 0/0/2
!
!
!
!
ip tftp source-interface GigabitEthernet0
ip tftp blocksize 8192
!
track 1 interface GigabitEthernet 0/0/4 line-protocol
!
interface Loopback 0
 ip address 209.165.201.1 255.255.255.224
!
interface GigabitEthernet 0/0/0
 vrf forwarding VRFA
 ip address 192.168.0.1 255.255.255.248
 ip nat inside
 negotiation auto
 bfd interval 50 min_rx 50 multiplier 3
 redundancy rii 2
!
interface GigabitEthernet 0/0/1
 ip address 209.165.202.129 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/2
 ip address 192.0.2.1 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/3
 ip address 198.51.100.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0
 vrf forwarding Mgmt-intf
 ip address 172.16.0.1 255.255.0.0
 negotiation auto
!
interface vasileft 1
 vrf forwarding VRFA
 ip address 10.4.4.1 255.255.0.0
 ip nat outside
 no keepalive
!
interface vasiright 1
ip address 10.4.4.2 255.255.0.0
no keepalive
!
router mobile
!
```

```
router bgp 577
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 neighbor 203.0.113.1 remote-as 223
 neighbor 203.0.113.1 description PEERING to PTNR neighbor 10.4.4.1 remote-as 577
 neighbor 10.4.4.1 description PEEERING to VASI VRFA interface
!
address-family ipv4
 network 203.0.113.1 mask 255.255.255.240
 network 10.4.0.0 mask 255.255.0.0
 network 209.165.200.224 mask 255.255.255.224
 neighbor 203.0.113.1 activate
 neighbor 10.4.4.1 activate
 neighbor 10.4.4.1 next-hop-self
 exit-address-family
!
address-family ipv4 vrf VRFA
 bgp router-id 4.4.4.4
 network 192.168.0.0 mask 255.255.255.248
 network 10.4.0.0 mask 255.255.0.0
 redistribute connected
 redistribute static
 neighbor 192.168.0.2 remote-as 65004
 neighbor 192.168.0.2 fall-over bfd
 neighbor 192.168.0.2 activate
 neighbor 10.4.4.2 remote-as 577
 neighbor 10.4.4.2 description PEERING to VASI Global intf
 neighbor 10.4.4.2 activate
 exit-address-family
!
ip nat switchover replication http
ip nat pool att_pool 209.165.200.225 209.165.200.225 prefix-length 16
ip nat inside source list 4 pool att_pool redundancy 1 mapping-id 100 vrf VRFA overload
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip route 203.0.113.1 255.255.255.224 10.4.4.1
ip route 192.168.0.0 255.255.0.0 10.4.4.1
ip route 209.165.200.224 255.255.255.224 10.4.4.1
ip route vrf Mgmt-intf 209.165.200.1 255.255.255.224 172.16.0.0
!
ip prefix-list VRF_Pool seq 5 permit 209.165.200.0/27
ip prefix-list p1-adv-1 seq 5 permit 209.165.200.0/27
ip prefix-list p1-exist-1 seq 5 permit 203.0.113.193/27
logging esm config
access-list 4 permit 203.0.113.193 255.255.255.224
!
control-plane
line console 0
 stopbits 1
!
line vty 0 3
 login
!
line vty 4
 password lab
 login
!
end
```

Example: Configuring Asymmetric Routing with VRF

The following example shows how to configure asymmetric routing with virtual routing and forwarding (VRF) instances:

```
Device(config)# redundancy
Device(config-red)# mode sso
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name RG1
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# priority 100 failover threshold 40
Device(config-red-app-grp)# control GigabitEthernet 1/0/3 protocol 1
Device(config-red-app-grp)# data GigabitEthernet 1/0/3
Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 1/0/4
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# exit
Device(config-red-app)# exit
Device(config-red)# exit
!
Device(config)# interface TenGigabitEthernet 2/0/0
Device(config-if)# ip vrf forwarding vrf001
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# exit
!
Device(config)# interface TenGigabitEthernet 3/0/0
Device(config-if)# ip vrf forwarding vrf001
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# exit
!
Device(config-if)# ip nat pool pool-vrf001 209.165.201.1 209.165.201.30 prefix-length 24
Device(config-if)# ip nat inside source list 1 pool pool-vrf001 redundancy 1 mapping-id 1
vrf vrf001 match-in-vrf overload
Device(config-if)# end
```

Additional References for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Firewall inter-chassis redundancy	"Configuring Firewall Stateful Inter-Chassis Redundancy" module
NAT inter-chassis redundancy	"Configuring Stateful Inter-Chassis Redundancy" module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

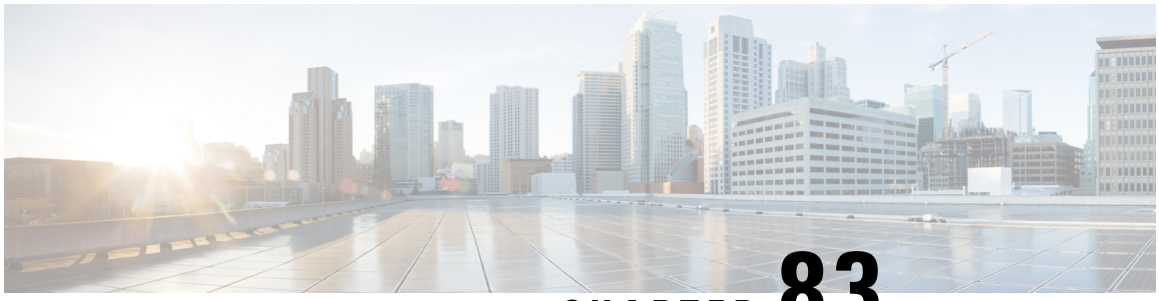
Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 122: Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Feature Name	Releases	Feature Information
Asymmetric Routing Enhancements for NAT44	Cisco IOS XE Release 3.16S	The Asymmetric Routing Enhancements for NAT44 feature supports asymmetric routing with CGN, ALGs, VRF, VASI and MPLS. No commands were introduced or modified.
Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT	Cisco IOS XE Release 3.5S	The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT feature supports the forwarding of packets from a standby redundancy group to the active redundancy group for packet handling. The following commands were introduced or modified: **asymmetric-routing**, **redundancy asymmetric-routing enable**.
VRF-Aware Interchassis Asymmetric Routing Support for Zone-Based Firewalls	Cisco IOS XE Release 3.14S	Zone-based firewalls support the VRF-Aware Interchassis Asymmetric Routing feature. This feature supports MPLS. There are no configuration changes for this feature. No commands were introduced or modified.
VRF-Aware Interchassis Asymmetric Routing Support for NAT	Cisco IOS XE Release 3.14S	NAT supports the VRF-Aware Interchassis Asymmetric Routing feature. This feature supports MPLS. There are no configuration changes for this feature. No commands were introduced or modified.



CHAPTER 83

VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

In Cisco IOS XE Release 3.14S, Network Address Translation (NAT) supports the VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy feature. VRF-Aware NAT for WAN-to-LAN topology is already supported in NAT.

This module describes this feature.

Restrictions for VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

The following features are not supported:

- Asymmetric routing
- Cisco TrustSec
- Edge switching services
- Header Compression
- IPsec
- Lawful intercept (Intercept twice, once at active and once at standby)
- Layer 2 Tunneling Protocol (L2TP)
- Locator-ID Separation Protocol (LISP) inner packet inspection
- Port Bundle

- Stile and Ceasr
- Secure Sockets Layer (SSL) VPN
- Session Border Controller (SBC)

Information About VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

VRF-Aware Box-to-Box High Availability Support

In Cisco IOS XE Release 3.14S, Network Address Translation (NAT) supports VRF-aware box-to-box high availability in a WAN-to-WAN topology.

To support VRF-aware box-to-box high availability, NAT ties the NAT mapping with a mandatorily configured mapping ID when a redundancy group (RG) is configured. The standby device retrieves the correct locally significant VRF ID from the mapping ID after synchronization. The VRF ID is set before NAT processes or translates a packet on the active device.

The VRF-aware box-to-box high availability configuration must be the same on both active and standby devices. The VRF configuration must use the same VRF name at active and standby devices. NAT provides a hashed VRF name value in the high availability message, and sends it to active and standby devices, so that the corresponding local VRF ID is converted at the peer device by using the VRF name hash value-to-VRF ID mapping.

Note In a High Availability configuration with HSRP or box-to-box redundancy, only the inside source NAT mappings are supported. The outside source NAT mappings are not supported with this configuration.

Note In some cases you might experience FTP disconnection after failover in a NAT B2B scenario. To resolve this issue, quit the existing FTP connection and start a new FTP connection.

Stateful Interchassis Redundancy Overview

You can configure the Stateful Interchassis Redundancy feature to determine the active device from a group of devices, based on a number of failover conditions. When a failover occurs, the standby device seamlessly takes over, starts performing traffic forwarding services, and maintains a dynamic routing table.

Note Manually shutting down the control or data interface link on an active NAT router results in traffic outage as the NAT router never transitions to active state.

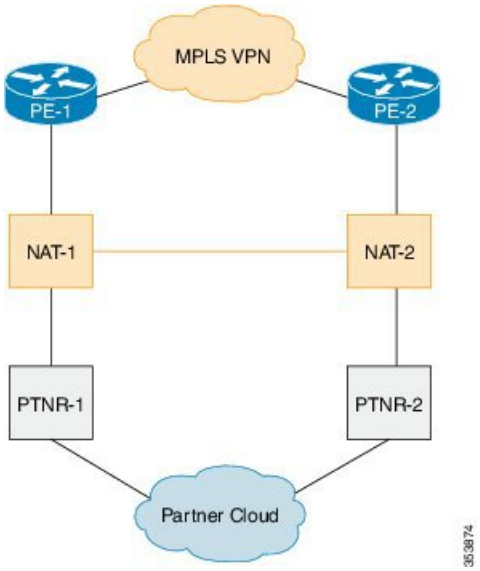
Stateful Interchassis Redundancy Operation in NAT

You can configure pairs of devices to act as hot standbys for each other. Redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups (RGs). Redundancy occurs at

an application level and does not require a complete physical failure of the interface or device for a switchover of the application to occur. When a switchover occurs, the application activity continues to run seamlessly on the redundant interface.

The figure below depicts an active/standby load-sharing scenario. The figure shows how an RG is configured for a pair of devices that has one outgoing interface. Group A on Router 1 is the active RG and Group A on Router 2 is the standby RG.

Figure 89: Stateful Interchassis Redundancy Operation in a WAN-WAN Topology



Redundant devices are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of devices. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and synchronize the stateful database. The pairs of redundant interfaces are configured with the same unique ID number known as the redundant interface identifier (RII).

The status of redundancy group members is determined through the use of hello messages sent over the control link. The software considers either device not responding to a hello message within a configurable amount of time to be a failure and initiates a switchover. For the software to detect a failure in milliseconds, control links run the failover protocol that is integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for hello messages:

- Hello time—Interval at which hello messages are sent.
- Hold time—Amount of time before which the active or standby device is declared to be down.

The hello time defaults to 3 seconds to align with the Hot Standby Router Protocol (HSRP), and the hold time defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine the pairs of interfaces that are affected by the switchover, you must configure a unique ID for each pair of redundant interfaces. This ID is known as the RII that is associated with the interface.

A switchover to the standby device can occur when the priority setting that is configured on each device changes. The device with the highest priority value acts as the active device. If a fault occurs on either the active or standby device, the priority of the device is decremented by a configurable amount known as the weight. If the priority of the active device falls below the priority of the standby device, a switchover occurs and the standby device becomes the active device. This default behavior can be overridden by disabling the

preemption attribute for the RG. You can also configure each interface to decrease the priority when the Layer 1 state of the interface goes down. The priority that is configured overrides the default priority of an RG.

Each failure event that causes a modification of an RG priority generates a syslog entry that contains a time stamp, the RG that was affected, the previous priority, the new priority, and a description of the failure event cause.

A switchover also can occur when the priority of a device or interface falls below a configurable threshold level.

A switchover to the standby device occurs under the following circumstances:

- Power loss or a reload occurs on the active device (including reloads).
- The run-time priority of the active device goes below that of the standby device (with preempt configured).
- The run-time priority of the active device goes below that of the configured threshold.
- The redundancy group on the active device is reloaded manually. Use the **redundancy application reload group** *rg-number* command for a manual reload.

How to Configure VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

The configuration for VRF-aware box-to-box redundancy is same as the configuration for stateful interchassis redundancy. For more information, see the "Configuring Stateful Interchassis Redundancy" module in the *IP Addressing: NAT Configuration Guide*.

Configuration Examples for VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

Example: Configuring VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

The following is a sample WAN-to-WAN symmetric routing configuration:

```
vrf definition Mgmt-intf
 address-family ipv4
  exit-address-family
!
address-family ipv6
 exit-address-family
!
!
vrf definition VRFA
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 address-family ipv4
  exit-address-family
```

```
!
!
no logging console
no aaa new-model
!
multilink bundle-name authenticated
!
redundancy
 mode sso
 application redundancy
  group 1
   preempt
   priority 120
   control GigabitEthernet 0/0/1 protocol 1
   data GigabitEthernet 0/0/2
!
!
!
!
ip tftp source-interface GigabitEthernet0
ip tftp blocksize 8192
!
track 1 interface GigabitEthernet 0/0/4 line-protocol
!
interface Loopback 0
 ip address 209.165.201.1 255.255.255.224
!
interface GigabitEthernet 0/0/0
 vrf forwarding VRFA
 ip address 192.168.0.1 255.255.255.248
 ip nat inside
 negotiation auto
 bfd interval 50 min_rx 50 multiplier 3
 redundancy rii 2
!
interface GigabitEthernet 0/0/1
 ip address 209.165.202.129 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/2
 ip address 192.0.2.1 255.255.255.224
 negotiation auto
!
interface GigabitEthernet 0/0/3
 ip address 198.51.100.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.1 255.255.255.240
 negotiation auto
!
interface GigabitEthernet 0
 vrf forwarding Mgmt-intf
 ip address 172.16.0.1 255.255.0.0
 negotiation auto
!
interface vasileft 1
 vrf forwarding VRFA
 ip address 10.4.4.1 255.255.0.0
 ip nat outside
 no keepalive
!
interface vasiright 1
ip address 10.4.4.2 255.255.0.0
```

```
no keepalive
!
router mobile
!
router bgp 577
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 neighbor 203.0.113.1 remote-as 223
 neighbor 203.0.113.1 description PEERING to PTNR neighbor 10.4.4.1 remote-as 577
 neighbor 10.4.4.1 description PEEERING to VASI VRFA interface
!
address-family ipv4
 network 203.0.113.1 mask 255.255.255.240
 network 10.4.0.0 mask 255.255.0.0
 network 209.165.200.224 mask 255.255.255.224
 neighbor 203.0.113.1 activate
 neighbor 10.4.4.1 activate
 neighbor 10.4.4.1 next-hop-self
 exit-address-family
!
address-family ipv4 vrf VRFA
 bgp router-id 4.4.4.4
 network 192.168.0.0 mask 255.255.255.248
 network 10.4.0.0 mask 255.255.0.0
 redistribute connected
 redistribute static
 neighbor 192.168.0.2 remote-as 65004
 neighbor 192.168.0.2 fall-over bfd
 neighbor 192.168.0.2 activate
 neighbor 10.4.4.2 remote-as 577
 neighbor 10.4.4.2 description PEERING to VASI Global intf
 neighbor 10.4.4.2 activate
 exit-address-family
!
ip nat switchover replication http
ip nat pool att_pool 209.165.200.225 209.165.200.225 prefix-length 16
ip nat inside source list 4 pool att_pool redundancy 1 mapping-id 100 vrf VRFA overload
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip route 203.0.113.1 255.255.255.224 10.4.4.1
ip route 192.168.0.0 255.255.0.0 10.4.4.1
ip route 209.165.200.224 255.255.255.224 10.4.4.1
ip route vrf Mgmt-intf 209.165.200.1 255.255.255.224 172.16.0.0
!
ip prefix-list VRF_Pool seq 5 permit 209.165.200.0/27
ip prefix-list p1-adv-1 seq 5 permit 209.165.200.0/27
ip prefix-list p1-exist-1 seq 5 permit 203.0.113.193/27
logging esm config
access-list 4 permit 203.0.113.193 255.255.255.224
!
control-plane
line console 0
 stopbits 1
!
line vty 0 3
 login
!
line vty 4
 password lab
 login
!
```

```
end
```

Additional References for VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference
NAT stateful interchassis redundancy	Configuring Stateful Interchassis Redundancy

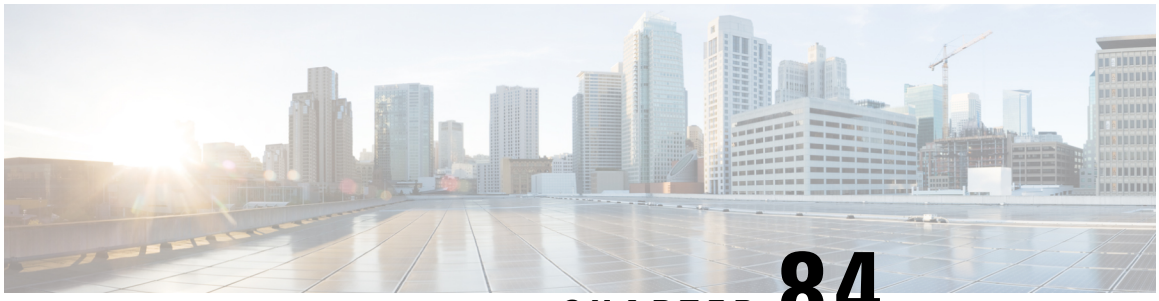
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

Table 123: Feature Information for VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy

Feature Name	Releases	Feature Information
VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy	Cisco IOS XE Release 3.14S	In Cisco IOS XE Release 3.14S, Network Address Translation (NAT) supports the VRF-Aware NAT for WAN-WAN Topology with Symmetric Routing Box-to-Box Redundancy feature. This feature contains the following two features: VRF-aware stateful interchassis redundancy and VRF-aware interchassis symmetric routing. No commands were introduced or modified by this feature.



CHAPTER 84

Integrating NAT with MPLS VPNs

Network Address Translation (NAT) Integration with MPLS VPNs feature allows multiple Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) to be configured on a single device to work together. NAT can differentiate which MPLS VPN it receives IP traffic from even if the MPLS VPNs are all using the same IP addressing scheme. This enhancement enables multiple MPLS VPN customers to share services while ensuring that each MPLS VPN is completely separate from the other.

Prerequisites for Integrating NAT with MPLS VPNs

- Before performing the tasks in this module, you should be familiar with the concepts described in the "Configuring NAT for IP Address Conservation" module.
- All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the *IP Access List Sequence Numbering* document at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fsaclseq.htm

Note If you specify an access list to use with a NAT command, NAT does not support the commonly used **permit ip any any** command in the access list.

Restrictions for Integrating NAT with MPLS VPNs

Inside VPN to VPN with NAT is not supported.

Information About Integrating NAT with MPLS VPNs

Benefits of NAT Integration with MPLS VPNs

MPLS service providers would like to provide value-added services such as Internet connectivity, domain name servers (DNS), and voice over IP (VoIP) service to their customers. The providers require that their customers; IP addresses be different when reaching the services. Because MPLS VPN allows customers to use overlapped IP addresses in their networks, NAT must be implemented to make the services possible.

Implementation Options for Integrating Nat with MPLS VPNs

There are two approaches to implementing NAT in the MPLS VPN network. NAT can be implemented on the customer edge (CE) router, which is already supported by NAT, or it can be implemented on a provider edge (PE) router. The NAT Integration with MPLS VPNs feature enables the implementation of NAT on a PE router in an MPLS cloud.

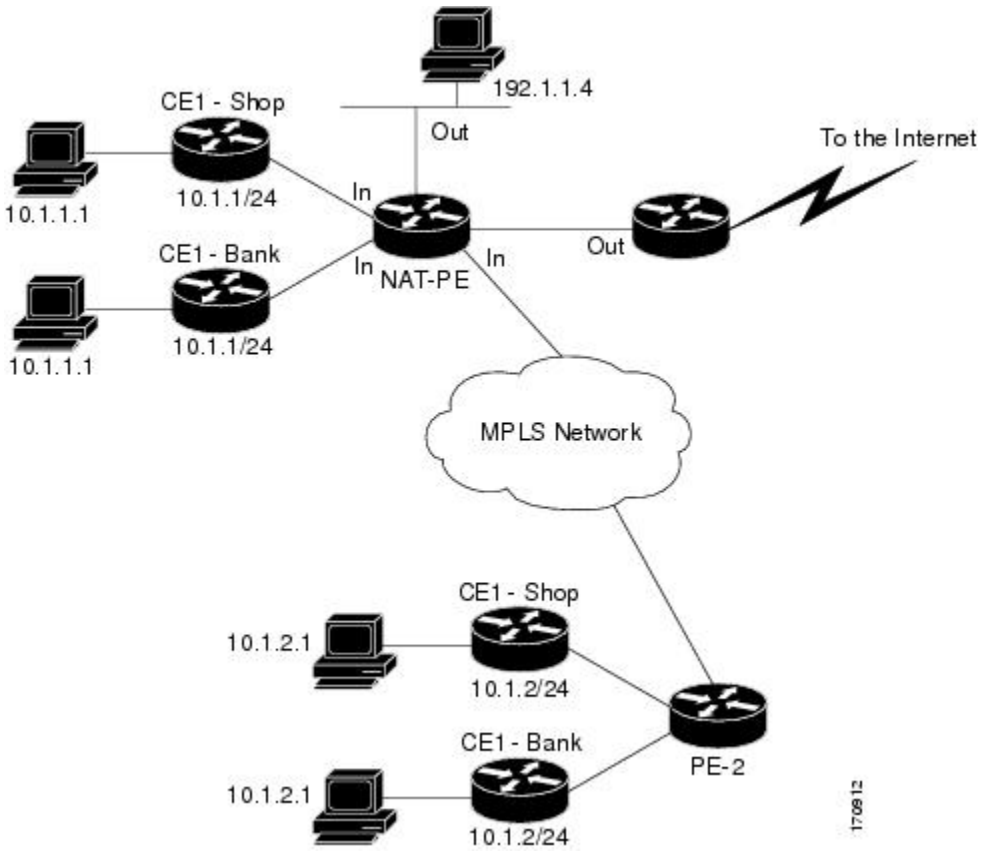
Scenarios for Implementing NAT on the PE Router

NAT could be implemented on the PE router in the following scenarios:

- Service point--Shared access can be from a generic interface or from a VPN interface.
- NAT point--NAT can be configured on the PE router that is directly connected to the shared access gateway, or on the PE router that is not directly connected to the shared access gateway.
- NAT interface--The shared access gateway interface most often is configured as the outside interface of NAT. The inside interface of NAT can be either the PE-CE interface of a VPN, the interface to the MPLS backbone, or both. The shared access gateway interface can also be configured as the inside interface.
- Routing type--Common service can be Internet connectivity or a common server. For Internet connectivity, a default route should be propagated to all the VPN customers that use the service. For common server access, a static or dynamically learned route should be propagated to the VPN customers.
- NAT configuration--NAT can have different configurations: static, dynamic, pool/interface overloading, and route-map.

The figure below shows a typical NAT integration with MPLS VPNs. The PE router connected to the internet and centralized mail service is employed to do the address translation.

Figure 90: Typical NAT Integration with MPLS VPNs



How to Integrate NAT with MPLS VPNs

Perform one or more of the following tasks depending on the type of translation you wish to configure for your network:

Configuring Inside Dynamic NAT with MPLS VPNs

Perform this task to configure your NAT PE router for dynamic translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
4. **ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number* | *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name*[**overload**]
5. Repeat Step 4 for each VPN being configured
6. **ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address*
7. Repeat Step 6 for each VPN being configured.

8. **exit**
9. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool inside 2.2.2.10 2.2.2.10 netmask 255.255.255.0`	Defines a pool of IP addresses for NAT.
Step 4	**ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number* | *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name*[**overload**] **Example:** `Router(config)# ip nat inside source list 1 pool mypool vrf shop overload`	Allows NAT to be configured on a particular VPN.
Step 5	Repeat Step 4 for each VPN being configured	--
Step 6	**ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address* **Example:** `Router(config)# ip route vrf shop 0.0.0.0 0.0.0.0 ethernet 0 168.58.88.2`	Allows NAT to be configured on a particular VPN.
Step 7	Repeat Step 6 for each VPN being configured.	--
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 9	**show ip nat translations vrf** *vrf-name* **Example:**	(Optional) Displays the settings used by virtual routing/forwarding (VRF) table translations.

	Command or Action	Purpose
	`Router# show ip nat translations vrf shop`	

Configuring Inside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**static** {**esp** *local-ip* **interface** *type number* | *local-ip global-ip*}} [**extendable** | **mapping-id** *map-id*| **no-alias** | **no-payload** | **redundancy** *group-name* | **route-map** | **vrf** *name*]
4. Repeat Step 3 for each VPN being configured.
5. **ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global**
6. Repeat Step 5 for each VPN being configured.
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**static** {**esp** *local-ip* **interface** *type number*	*local-ip global-ip*}} [**extendable**	**mapping-id** *map-id*| **no-alias**	**no-payload**	**redundancy** *group-name*	**route-map**	**vrf** *name*] **Example:** `Router(config)# ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop`	Enables inside static translation on the VRF.
Step 4	Repeat Step 3 for each VPN being configured.	--
Step 5	**ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global** **Example:**	Allows the route to be shared by several customers.

	Command or Action	Purpose
	`Router(config)# ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global`	
Step 6	Repeat Step 5 for each VPN being configured.	--
Step 7	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuring Outside Dynamic NAT with MPLS VPNs

Perform this step to configure your NAT PE router for dynamic outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool outside** *global-ip local-ip* **netmask** *netmask*
4. **ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name*
5. Repeat Step 4 for each VRF being configured.
6. **ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat pool outside** *global-ip local-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.00`	Allows the configured VRF to be associated with the NAT translation rule.
Step 4	**ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name* **Example:** `Router(config)# ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop`	Allows the route to be shared by several customers.
Step 5	Repeat Step 4 for each VRF being configured.	Allows the route to be shared by several customers.
Step 6	**ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name* **Example:** `Router(config)# ip nat outside source static 168.58.88.2 4.4.4.1 vrf shop`	Enables NAT translation of the outside source address.
Step 7	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuring Outside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **ip nat pool inside** *global-ip local-ip* **netmask** *netmask*
4. Repeat Step 3 for each pool being configured.
5. **ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name*
6. Repeat Step 5 for each pool being configured.

7. **ip nat outside source static** *global-ip local-ip* vrf *vrf-name*
8. Repeat Step 7 for all VPNs being configured.
9. **exit**
10. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure** {**terminal** | **memory** | **network**} **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool inside** *global-ip local-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0`	Allows the configured VRF to be associated with the NAT translation rule.
Step 4	Repeat Step 3 for each pool being configured.	--
Step 5	**ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* **Example:** `Router(config)# ip nat inside source list 1 pool inside2 vrf shop`	Allows the route to be shared by several customers.
Step 6	Repeat Step 5 for each pool being configured.	Defines the access list.
Step 7	**ip nat outside source static** *global-ip local-ip* vrf *vrf-name* **Example:** `Router(config)# ip nat outside source static 168.58.88.2 4.4.4.1 vrf shop`	Allows the route to be shared by several customers.
Step 8	Repeat Step 7 for all VPNs being configured.	--
Step 9	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 10	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuration Examples for Integrating NAT with MPLS VPNs

Configuring Inside Dynamic NAT with MPLS VPNs Example

The following example shows configuring inside Dynamic NAT with MPLS VPNs.

```
!
ip nat pool inside 2.2.2.10 2.2.2.10 netmask 255.255.255.0
ip nat inside source list 1 pool inside vrf bank overload
ip nat inside source list 1 pool inside vrf park overload
ip nat inside source list 1 pool inside vrf shop overload
!
ip route vrf shop 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
ip route vrf bank 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
ip route vrf park 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Configuring Inside Static NAT with MPLS VPNs Example

The following example shows configuring inside static NAT with MPLS VPNs.

```
!
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat inside source static 192.168.11.1 2.2.2.11 vrf shop
ip nat inside source static 192.168.11.3 2.2.2.12 vrf shop
ip nat inside source static 140.48.5.20 2.2.2.13 vrf shop
!
ip route 2.2.2.1 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.2 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.3 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.4 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.5 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.6 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.11 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.12 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.13 255.255.255.255 Ethernet1/0 192.168.121.113
```

Configuring Outside Dynamic NAT with MPLS VPNs Example

The following example shows configuring outside dynamic NAT with MPLS VPNs.

```
!
ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.0
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat outside source list 1 pool outside
!
```

Configuring Outside Static NAT with MPLS VPNs Example

The following example shows configuring outside static NAT with MPLS VPNs.

```
!
ip default-gateway 10.1.15.1
ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0
ip nat pool inside2 2.2.2.1 2.2.2.254 netmask 255.255.255.0
ip nat pool inside3 2.2.3.1 2.2.3.254 netmask 255.255.255.0
ip nat inside source list 1 pool inside2 vrf bank
ip nat inside source list 1 pool inside3 vrf park
ip nat inside source list 1 pool inside1 vrf shop
ip nat outside source static 168.58.88.2 4.4.4.1 vrf bank
ip nat outside source static 18.68.58.1 4.4.4.2 vrf park
ip nat outside source static 168.58.88.1 4.4.4.3 vrf shop
ip classless
ip route 192.170.10.0 255.255.255.0 Ethernet1/0 192.168.121.113
ip route 192.170.11.0 255.255.255.0 Serial2/1.1 192.168.121.113
ip route 192.170.12.0 255.255.255.0 FastEthernet0/0 192.168.121.113
ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf bank 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf park 0.0.0.0 0.0.0.0 168.58.88.2 global
no ip http server
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Where to Go Next

- To learn about Network Address Translation and configure NAT for IP address conservation, see the “Configuring NAT for IP Address Conservation” module.
- To verify, monitor, and maintain NAT, see the “Monitoring and Maintaining NAT” module.
- To use NAT with application level gateways, see the “Using Application Level Gateways with NAT” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References for Integrating NAT with MPLS VPNs

Related Documents

Related Topic	Document Title
IOS Commands	Cisco IOS Master Command List
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard & RFC	Title
RFC 2547	*BGP/MPLS VPNs*

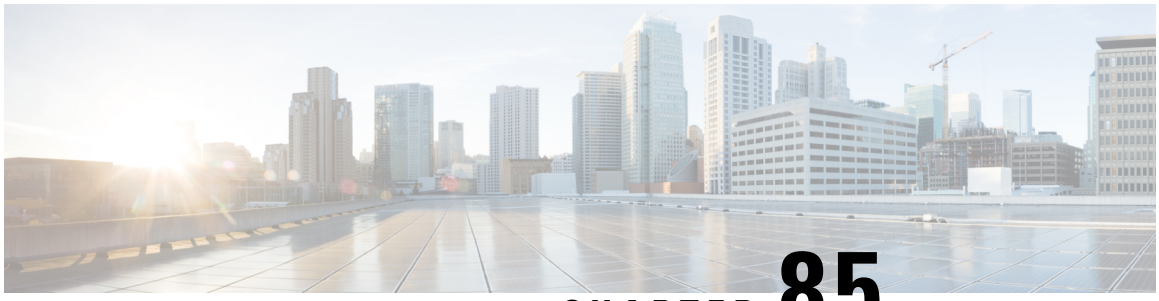
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Integrating NAT with MPLS VPNs

Table 124: Feature Information for Integrating NAT with MPLS VPNs

Feature Name	Releases	Feature Configuration Information
Integrating NAT with MPLS VPNs	12.1(13)T 15.1(1)SY	The Integrating NAT with MPLS VPNs feature allows multiple Multiprotocol Label Switching (MPLS) VPNs to be configured on a single device to work together.



CHAPTER 85

Monitoring and Maintaining NAT

The Monitoring and Maintaining NAT feature enables the monitoring of Network Address Translation (NAT) by using translation information and statistics displays. It enables the logging of NAT translation to log and track system error messages and exceptions. The Monitoring and Maintaining NAT feature helps maintain NAT by clearing NAT translations before the timeout is expired.

This modules the Monitoring and Maintaining NAT feature.

Prerequisites for Monitoring and Maintaining NAT

Before performing the tasks in this module, you must be familiar with the concepts described in the "Configuring NAT for IP Address Conservation" module and have NAT configured in your network.

Restrictions for Monitoring and Maintaining NAT

- Syslog for Network Address Translation (NAT) is not supported.
- On the Cisco Catalyst 8300 Series, Bidirectional Forwarding Detection (BFD) flaps occur when the clear IP NAT translation privileged EXEC command is executed, particularly when NAT is set up with a high volume of NAT sessions or translations. Although BFD is set up on a different interface from NAT, BFD sessions tend to immediately flap due to an echo failure. This happens because NAT briefly locks the database for a few seconds to finalize the clear operation, which can cause a momentary disruption. An echo failure is a situation in which a network device does not receive a reply to an echo request within a designated time. This echo request is part of the control messages used in protocols like BFD to check the availability and operational status of other network devices.

Information About Monitoring and Maintaining NAT

NAT Display Contents

There are two basic types of IP Network Address Translation (NAT) translation information:

Translation Entries

Translation entry information includes the following:

- The protocol of the port identifying the address.
- The legitimate IP address that represents one or more inside local IP addresses to the outside world.
- The IP address assigned to a host on the inside network; probably not a legitimate address assigned by the NIC or service provider.
- The IP address of an outside host as it appears to the inside network; probably not a legitimate address assigned by the NIC or service provider.
- The IP address assigned to a host on the outside network by its owner.
- The time since the entry was created (in hours:minutes:seconds).
- The time since the entry was last used (in hours:minutes:seconds).
- Flags indicating the type of translation. Possible flags are:
  - extended—Extended translation.
  - static—Static translation.
  - destination—Rotary translation.
  - outside—Outside translation.
  - timing out—Translation will no longer be used, due to a TCP finish (FIN) or reset (RST) flag.

Statistical Information

Statistical information includes the following:

- The total number of translations active in the system. This number is incremented each time a translation is created and is decremented each time a translation is cleared or times out.
- A list of interfaces marked as outside with the **ip nat outside** command.
- A list of interfaces marked as inside with the **ip nat inside** command.
- The number of times the software does a translations table lookup and finds an entry.
- The number of times the software does a translations table lookup, fails to find an entry, and must try to create one.
- A cumulative count of translations that have expired since the router was booted.
- Information about dynamic mappings.
- Information about an inside source translation.

- The access list number being used for the translation.
- The name of the pool.
- The number of translations using this pool.
- The IP network mask being used in the pool.
- The starting IP address in the pool range.
- The ending IP address in the pool range.
- The type of pool. Possible types are generic or rotary.
- The number of addresses in the pool available for translation.
- The number of addresses being used.
- The number of failed allocations from the pool.

NAT does not support access control lists (ACLs) with the log option. The same functionality can be achieved by using one of the following options:

- By having a physical interface or virtual LAN (VLAN) with the logging option
- By using NetFlow

NAT-Forced Clear of Dynamic NAT Half-Entries

The NAT-Forced Clear of Dynamic NAT Half-Entries feature filters the display of the translation table by specifying an inside or outside address. This feature introduces the **clear ip nat translation forced** command that forcefully clears active dynamic Network Address Translation (NAT) half-entries that have child translations.

How to Monitor and Maintain NAT

Displaying NAT Translation Information

SUMMARY STEPS

1. **enable**
2. **show ip nat translations** [**verbose**]
3. **show ip nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT translations.
Step 3	**show ip nat statistics** **Example:** `Device# show ip nat statistics`	(Optional) Displays active NAT translation statistics.

Example:

The following is sample output from the **show ip nat translations** command:

```
Device# show ip nat translations

Pro Inside global         Inside local       Outside local        Outside global
tcp 192.168.1.1:514      192.168.2.3:53     192.168.2.22:256     192.168.2.22:256
tcp 192.168.1.1:513      192.168.2.2:53     192.168.2.22:256     192.168.2.22:256
tcp 192.168.1.1:512      192.168.2.4:53     192.168.2.22:256     192.168.2.22:256
Total number of translations: 3
```

The following is sample output from the **show ip nat translations verbose** command:

```
Device# show ip nat translations verbose

Pro Inside global        Inside local       Outside local      Outside global
tcp 192.168.1.1:514      192.168.2.3:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef80350, use_count:1
tcp 192.168.1.1:513      192.168.2.2:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef801b0, use_count:1
tcp 192.168.1.1:512      192.168.2.4:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef80280, use_count:1
Total number of translations: 3
```

The following is sample output from the **show ip nat statistics** command:

```
Device# show ip nat statistics

Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Outside interfaces:
GigabitEthernet0/3/0
Inside interfaces:
GigabitEthernet0/3/1
Hits: 3228980 Misses: 3
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool pool1 refcount 3
  pool pool1: netmask 255.255.255.0
  start 198.168.1.1 end 198.168.254.254
  type generic, total addresses 254, allocated 0 (0%), misses 0
  longest chain in pool: pool1's addr-hash: 0, average len 0,chains 0/256
```

```
Pool stats drop: 0 Mapping stats drop: 0
Port block alloc fail: 0
IP alias add fail: 0
Limit entry add fail: 0
```

Clearing NAT Entries Before the Timeout

By default, dynamic address translations will time out from the NAT translation table at some point. Perform this task to clear the entries before the timeout.

SUMMARY STEPS

1. **enable**
2. **clear ip nat translation inside** *global-ip local-ip* **outside** *local-ip global-ip*
3. **clear ip nat translation outside** *global-ip local-ip*
4. **clear ip nat translation** *protocol* **inside** *global-ip global-port local-ip local-port* **outside** *local-ip local-port global-ip global-port*
5. **clear ip nat translation** {* | [**forced**] | [**inside** *global-ip local-ip*] [**outside** *local-ip global-ip*]}
6. **clear ip nat translation inside** *global-ip local-ip* [**forced**]
7. **clear ip nat translation outside** *local-ip global-ip* [**forced**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip nat translation inside** *global-ip local-ip* **outside** *local-ip global-ip* **Example:** `Device# clear ip nat translation inside 192.168.2.209 192.168.2.95 outside 192.168.2.100 192.168.2.101`	(Optional) Clears a single dynamic half-entry containing an inside translation or both an inside and outside translation created in a dynamic configuration. • A dynamic half-entry is cleared only if it does not have any child translations.
Step 3	**clear ip nat translation outside** *global-ip local-ip* **Example:** `Device# clear ip nat translation outside 192.168.2.100 192.168.2.80`	(Optional) Clears a single dynamic half-entry containing an outside translation created in a dynamic configuration. • A dynamic half-entry is cleared only if it does not have any child translations.
Step 4	**clear ip nat translation** *protocol* **inside** *global-ip global-port local-ip local-port* **outside** *local-ip local-port global-ip global-port* **Example:** `Device # clear ip nat translation udp inside 192.168.2.209 1220 192.168.2.195 1220 outside 192.168.2.13 53 192.168.2.132 53`	(Optional) Clears a UDP translation entry.

	Command or Action	Purpose
Step 5	**clear ip nat translation** {* | [**forced**] | [**inside** *global-ip local-ip*] [**outside** *local-ip global-ip*]} **Example:** `Device# clear ip nat translation *`	(Optional) Clears either all dynamic translations (with the ***** or **forced** keyword), a single dynamic half-entry containing an inside translation, or a single dynamic half-entry containing an outside translation. • A single dynamic half-entry is cleared only if it does not have any child translations.
Step 6	**clear ip nat translation inside** *global-ip local-ip* [**forced**] **Example:** `Device# clear ip nat translation inside 192.168.2.209 192.168.2.195 forced`	(Optional) Forces the clearing of a single dynamic half-entry and its child translations containing an inside translation created in a dynamic configuration, with or without its corresponding outside translation. • A dynamic half-entry is always cleared, regardless of whether it has any child translations.
Step 7	**clear ip nat translation outside** *local-ip global-ip* [**forced**] **Example:** `Device# clear ip nat translation outside 192.168.2.100 192.168.2.80 forced`	(Optional) Forces the clearing of a single dynamic half-entry and its child translations containing an outside translation created in a dynamic configuration. • A dynamic half-entry is always cleared, regardless of whether it has any child translations.

Examples for Monitoring and Maintaining NAT

Example: Clearing UDP NAT Translations

The following example shows the Network Address Translation (NAT) entries before and after the UDP entry is cleared:

```
Device# show ip nat translation
Pro Inside global        Inside local         Outside local      Outside global
udp 192.168.2.20:1220    192.168.2.95:1220    192.168.2.22:53    192.168.2.20:53
tcp 192.168.2.20:11012   192.168.2.209:11012  171.69.1.220:23    192.168.2.20:23
tcp 192.168.2.20:1067    192.168.2.20:1067    192.168.2.20:23    192.168.2.20:23

Device# clear ip nat translation udp inside 192.168.2.20:1067 192.168.2.20:1067 outside
192.168.2.20:23 192.168.2.20:23
Device# show ip nat translation

Pro Inside global      Inside local        Outside local      Outside global
udp 192.168.2.20:1220  192.168.2.95:1220   192.168.2.22:53    192.168.2.20:53
tcp 192.168.2.20:11012 192.168.2.209:11012 171.69.1.220:23    192.168.2.20:23
```

Additional References for Monitoring and Maintaining NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
NAT for IP address conservation	"Configuring NAT for IP Address Conservation" module

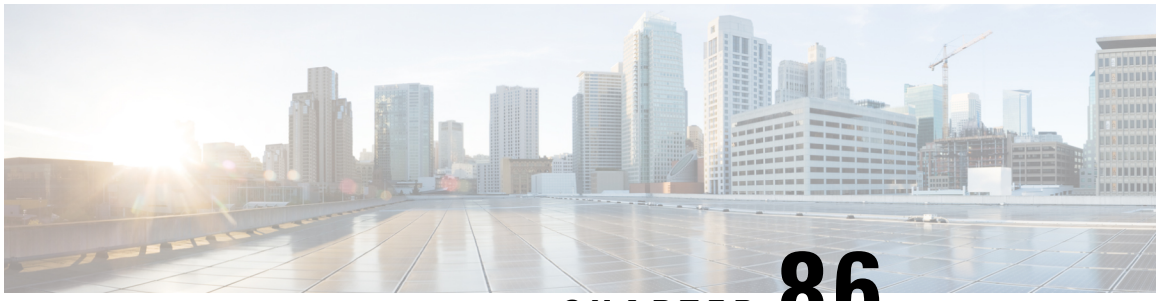
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Monitoring and Maintaining NAT

Table 125: Feature Information for Monitoring and Maintaining NAT

Feature Name	Releases	Feature Information
NAT—Forced Clear of Dynamic NAT Half-Entries	Cisco IOS XE Release 2.4	The NAT-Forced Clear of Dynamic NAT Half-Entries feature filters the display of the translation table by specifying an inside or outside address. The following commands were introduced or modified: **clear ip nat translations forced**, **show ip nat translations**.



CHAPTER 86

Information About NAT 44 Pool Exhaustion Alerts

The NAT 44 pool exhaustion alert feature enables generation of alerts before addresses in an address pool are exhausted. The alerts are generated for TCP and UDP ports and separate Syslog entries are generated for each protocol. This feature can help the administrator take action before the address pool is exhausted.

Define Thresholds for Address Pool

You can define high and low thresholds for the address pool. These thresholds are set in terms of percentage. You can use ip nat settings command to configure the threshold limits.

Thresholds Applicable for Different Address Pools

When you specify a threshold, the usage of the address pools are as follows:

Address Pool Type	Pool Usage Based On
Pool for port-address translation (PAT)	Pool usage for such pools will be based total ports allocated from the pool
Pool for Address translation	Pool usage for such pools will be based on address allocated from the pool
Pool with BPA configured	Pool usage will be based on the total number of port-sets allocated from the pool

Prerequisites for NAT 44 Pool Exhaustion Alerts

Before performing the tasks in this module, you must be familiar with the concepts described in the "Monitoring and Maintaining NAT" module and have NAT configured in your network.

Restrictions for NAT 44 Pool Exhaustion Alerts

The NAT 44 Pool Exhaustion Alert feature does not support setting alerts for ICMP ports.

Use Case on How NAT 44 Pool Exhaustion Alerts Work

Let us assume you have defined thresholds for the address pool using the following command:

ip nat settings pool watermark high 80 low 50

This means that the higher threshold and lower threshold for the address pool is set at 80 and 50 percent respectively.

Pool Usage	Syslog Status
Pool usage from 78-80	Syslogs are generated
Pool usage falls to 60	No Syslogs are generated
Pool usage increased again to more than or equal to 80	No Syslogs are generated
Pool usage decreased to less than or equal to 50	No Syslogs are generated
Pool usage increased to more than or equal to 80	Syslogs are generated

Additional References for NAT 44 Pool Exhaustion Alerts

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference

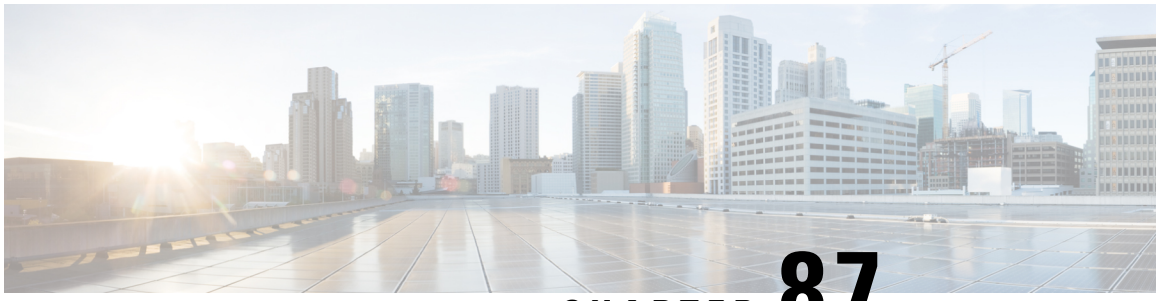
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NAT 44 Pool Exhaustion Alerts

Table 126: Feature Information for NAT 44 Pool Exhaustion Alerts

Feature Name	Releases	Feature Information
NAT 44 Pool Exhaustion Alerts	Cisco IOS XE Fuji 16.8	The NAT 44 pool exhaustion alert feature enables generation of alerts before addresses in an address pool are exhausted. The alerts are generated for TCP and UDP ports and separate Syslog entries are generated for each protocol. This feature can help the administrator take action before the address pool is exhausted.. The following command is introduced : **ip nat settings pool watermark**, .



CHAPTER 87

Enabling NAT High-Speed Logging per VRF

The Enabling NAT High-Speed Logging Per VRF feature provides the ability to enable and disable Network Address Translation (NAT) high-speed logging (HAL) for virtual routing and forwarding (VRF) instances.

This module provides information about how to enable HSL for VRFs.

Information About Enabling NAT High-Speed Logging per VRF

High-Speed Logging for NAT

Network Address Translation (NAT) supports high-speed logging (HSL) for upto 4 destinations. When HSL is configured, NAT provides a log of the packets flowing through the routing devices (similar to the Version 9 NetFlow-like records) to an external collector. Records are sent for each binding (binding is the address binding between the local address and the global address to which the local address is translated) and when sessions are created and destroyed. Session records contain the full 5-tuple of information (the source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements. NAT also sends an HSL message when a NAT pool runs out of addresses (also called *pool exhaustion*). Because the pool exhaustion messages are rate limited, each packet that hits the pool exhaustion condition does not trigger an HSL message.

The table below describes the templates for HSL bind and session create or destroy.

Table 127: Template for HSL Bind and Session Create or Destroy

Field	Format	ID	Value
Source IP address	IPv4 address	8	varies
Translated source IP address	IPv4 address	225	varies

Field	Format	ID	Value
Destination IP address	IPv4 address	12	varies
Translated destination IP address	IPv4 address	226	varies
Original source port	16-bit port	7	varies
Translated source port	16-bit port	227	varies
Original destination port	16-bit port	11	varies
Translated destination port	16-bit port	228	varies
Virtual routing and forwarding (VRF) ID	32-bit ID	234	varies
Protocol	8-bit value	4	varies
Event	8-bit value	230	0-Invalid 1-Adds event 2-Deletes event
Unix timestamp in milliseconds	64-bit value	323	varies **Note** Based on your release version, this field will be available.

The table below describes the HSL pool exhaustion templates.

Table 128: Template for HSL Pool Exhaustion

Field	Format	ID	Values
NAT pool ID	32-bit value	283	varies
NAT event	8-bit value	230	3-Pool exhaust

How to Configure Enabling NAT High-Speed Logging per VRF

Enabling High-Speed Logging of NAT Translations

You can enable or disable high-speed logging (HSL) of all Network Address Translation (NAT) translations or only translations for specific VPNs.

You must first use the **ip nat log translations flow-export v9 udp destination** command to enable HSL for all VPN and non-VPN translations. . VPN translations are also known as Virtual Routing and Forwarding (VRF) translations.

After you enable HSL for all NAT translations, you can then use the **ip nat log translations flow-export v9** *vrf-name* command to enable or disable translations for specific VPNs. When you use this command, HSL is disabled for all VPNs, except for the ones the command is explicitly enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat log translations flow-export v9 udp destination source** *interface type interface-number*
4. **ip nat log translations flow-export v9** {*vrf-name* | **global-on**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat log translations flow-export v9 udp destination source** *interface type interface-number* **Example:** This example shows how to enable high-speed logging using an IPv4 address `Device(config)# ip nat log translations flow-export v9 udp destination 10.10.0.1 1020 source GigabitEthernet 0/0/0`	
Step 4	**ip nat log translations flow-export v9** {*vrf-name* | **global-on**} **Example:** `Device(config)# ip nat log translations flow-export v9 VPN-18`	Enables or disables the high-speed logging of specific NAT VPN translations.
Step 5	**exit** **Example:** `Device(config)# exit`	(Optional) Exits global configuration mode and enters privileged EXEC mode.

Disabling High-Speed Logging of NAT Translations

You can disable high-speed logging (HSL) of all Network Address Translation (NAT) translations or only translations for specific VPNs.

To disable NAT Logging follow the below steps. The commands have to be in the order specified if vrf-logging is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip nat log translations flow-export v9 vrf***vrf-name***on**
4. **no ip nat log translations flow-export v9udp destination** *ip-address vrf-source-name* **vrf** *vrf-destination-name*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip nat log translations flow-export v9 vrf***vrf-name***on** **Example:** This example shows how to disable high-speed logging using an IPv4 address `Device(config)# no ip nat log translations flow-export v9 vrf 1020 on`	
Step 4	**no ip nat log translations flow-export v9udp destination** *ip-address vrf-source-name* **vrf** *vrf-destination-name* **Example:** `Device(config)# no ip nat log translations flow-export v9 udp destination 10.10.0.1 1020 vrf 2030`	Disables the high-speed logging of specific NAT VPN translations.
Step 5	**exit** **Example:** `Device(config)# exit`	(Optional) Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Enabling NAT High-Speed Logging per VRF

Example: Enabling High-Speed Logging of NAT Translations

```
Device# configure terminal
Device(config)# ip nat log translations flow-export v9 udp destination 10.10.0.1 1020 source
 GigabitEthernet 0/0/0
Device(config)# ip nat log translations flow-export v9 VPN-18
Device(config)# exit
```

Additional References for Enabling NAT High-Speed Logging per VRF

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title

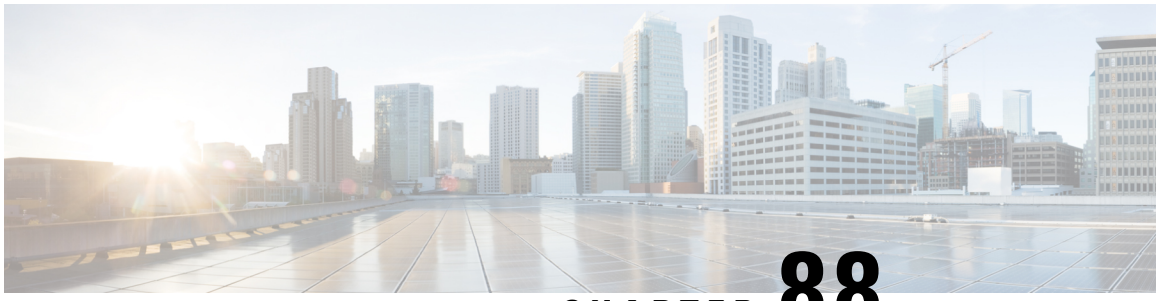
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Enabling NAT High-Speed Logging per VRF

Table 129: Feature Information for Enabling NAT HIgh-Speed Logging per VRF

Feature Name	Releases	Feature Information
Enabling NAT High-Speed Logging per VRF	Cisco IOS XE Release 3.1S	The Enabling NAT High-Speed Logging per VRF feature provides the ability to enable and disable Network Address Translation (NAT) high-speed logging (HAL) for virtual routing and forwarding (VRF) instances. The following commands were introduced or modified: **ip nat log translations flow-export**.



CHAPTER 88

Stateless Network Address Translation 64

The Stateless Network Address Translation 64 (NAT64) feature provides a translation mechanism that translates an IPv6 packet into an IPv4 packet and vice versa. The translation involves parsing the entire IPv6 header, including the extension headers, and extracting the relevant information and translating it into an IPv4 header.This processing happens on a per-packet basis on the interfaces that are configured for Stateless NAT64 translation.

The Stateless NAT64 translator enables native IPv6 or IPv4 communication and facilitates coexistence of IPv4 and IPv6 networks.

In Cisco IOS-XE release 17.4 release, support is introduced to map a VRF to an IPv4 to IPv6 prefix mapping. Multiple source and destination prefix can be mapped to a VRF.

The Stateless NAT64 translator does not maintain any state information in the datapath.

Restrictions for Stateless Network Address Translation 64

The following restrictions apply to the Stateless NAT64 feature:

- Only valid IPv4-translatable addresses can be used for stateless translation.
- Multicast is not supported.
- Applications without a corresponding application layer gateway (ALG) may not work properly with the Stateless NAT64 translator.
- The translation of IPv4 options, IPv6 routing headers, hop-by-hop extension headers, destination option headers, and source routing headers are not supported.
- Fragmented IPv4 UDP packets that do not contain a UDP checksum are not translated.

- IPv6 packets with zero UDP checksum are not translated.
- Both NAT44 (static, dynamic, and Port Address Translation [PAT]) configurations and Stateless NAT64 configuration are not supported on the same interface.

Restrictions for Stateless Network Address Translation 64

In addition to the restrictions indicated for Stateless Network Address Translation 64, the followingrestrictions are applicable to mapping a VRF to an IPv4 to IPv6 prefix mapping:

- High-Speed Logging is not supported.
- Multicast is not supported.
- This feature cannot be configured in a high availability scenario.
- This feature is only supported in autonomous mode.
- NAT64 static command is not supported for this feature.

Information About Stateless Network Address Translation 64

Fragmentation of IP Datagrams in IPv6 and IPv4 Networks

In IPv4 networks, any intermediate router can do the fragmentation of an IP datagram. However, in IPv6 networks, fragmentation can be done only by the originating IPv6 host. Because fragmentation in IPv6 networks is done by the IPv6 hosts, the path maximum transmission unit (PMTU) discovery should also be done by the IPv6 hosts. However, a PMTU discovery is not possible across an IPv4 network where the routers are allowed to fragment the packets. In IPv4 networks, a Stateless NAT64 translator is used to fragment the IPv6 datagram and set the Don't Fragment (DF) bits in the IPv4 header. Similarly, the translator can add the fragment header to the IPv6 packet if an IPv4 fragment is received.

Translation of ICMP for Stateless NAT64 Translation

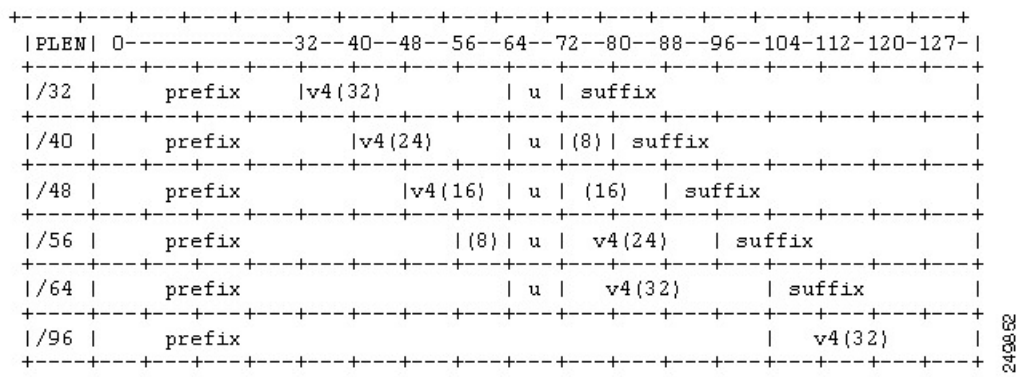
The IETF draft on the IP/ICMP translation algorithm describes the ICMP types or codes that should be translated between IPv4 and IPv6. ICMP errors embed the actual IP header and the transport header. Because the ICMP errors are embedded in the IP header, the IP header is not translated properly. For ICMP error packets, Stateless NAT64 translation should be applied twice: once for the outer header, and once again for the embedded header.

IPv4-Translatable IPv6 Address

IPv4-translatable IPv6 addresses are IPv6 addresses assigned to the IPv6 nodes for use with stateless translation. IPv4-translatable addresses consist of a variable-length prefix, an embedded IPv4 address, fixed universal bits (u-bits), and in some cases a suffix. IPv4-embedded IPv6 addresses are IPv6 addresses in which 32 bits contain an IPv4 address. This format is the same for both IPv4-converted and IPv4-translatable IPv6 addresses.

The figure below shows an IPv4-translatable IPv6 address format with several different prefixes and embedded IPv4 address positions.

Figure 91: IPv4-Translatable IPv6 Address Format



Prefixes Format

A set of bits at the start of an IPv6 address is called the format prefix. Prefix length is a decimal value that specifies how many of the leftmost contiguous bits of an address comprise the prefix.

An embedded IPv4 address is used to construct IPv4 addresses from the IPv6 packet. The Stateless NAT64 translator has to derive the IPv4 addresses that are embedded in the IPv6-translatable address by using the prefix length. The translator has to construct an IPv6-translatable address based on the prefix and prefix length and embed the IPv4 address based on the algorithm.

The prefix lengths of 32, 40, 48, 56, 64, or 96 are supported for Stateless NAT64 translation. The Well Known Prefix (WKP) is not supported. When traffic flows from the IPv4-to-IPv6 direction, either a WKP or a configured prefix can be added only in stateful translation.

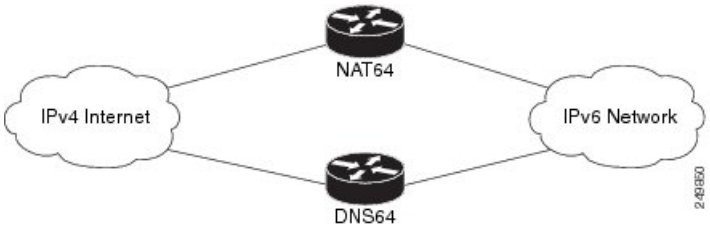
Supported Stateless NAT64 Scenarios

The following scenarios are supported by the Cisco IOS Stateless NAT64 feature and are described in this section:

- Scenario 1--an IPv6 network to the IPv4 Internet
- Scenario 2--the IPv4 Internet to an IPv6 network
- Scenario 5--an IPv6 network to an IPv4 network
- Scenario 6--an IPv4 network to an IPv6 network

The figure below shows stateless translation for scenarios 1 and 2. An IPv6-only network communicates with the IPv4 Internet.

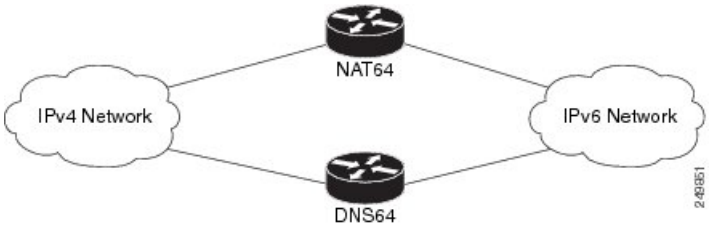
Figure 92: Stateless Translation for Scenarios 1 and 2



Scenario 1 is an IPv6 initiated connection and scenario 2 is an IPv4 initiated connection. Stateless NAT64 translates these two scenarios only if the IPv6 addresses are IPv4 translatable. In these two scenarios, the Stateless NAT64 feature does not help with IPv4 address depletion, because each IPv6 host that communicates with the IPv4 Internet is a globally routable IPv4 address. This consumption is similar to the IPv4 consumption rate as a dual-stack. The savings, however, is that the internal network is 100 percent IPv6, which eases management (Access Control Lists, routing tables), and IPv4 exists only at the edge where the Stateless translators live.

The figure below shows stateless translation for scenarios 5 and 6. The IPv4 network and IPv6 network are within the same organization.

Figure 93: Stateless Translation for Scenarios 5 and 6



The IPv4 addresses used are either public IPv4 addresses or RFC 1918 addresses. The IPv6 addresses used are either public IPv6 addresses or Unique Local Addresses (ULAs).

Both these scenarios consist of an IPv6 network that communicates with an IPv4 network. Scenario 5 is an IPv6 initiated connection and scenario 6 is an IPv4 initiated connection. The IPv4 and IPv6 addresses may not be public addresses. These scenarios are similar to the scenarios 1 and 2. The Stateless NAT64 feature supports these scenarios if the IPv6 addresses are IPv4 translatable.

Multiple Prefixes Support for Stateless NAT64 Translation

Network topologies that use the same IPv6 prefix for source and destination addresses may not handle routing correctly and may be difficult to troubleshoot. The Stateless NAT64 feature addresses these challenges in Cisco IOS XE Release 3.3S and later releases through the support of multiple prefixes for stateless translation. The entire IPv4 Internet is represented as using a different prefix from the one used for the IPv6 network.

Support to Map a VRF to an IPv4 to IPv6 Prefix Mapping

The stateless NAT64 IPv4 to IPv6 prefix mappings is now VRF-aware. Additionally multiple source and destination prefix can be mapped to a VRF.

How to Configure Stateless Network Address Translation 64

Configuring a Routing Network for Stateless NAT64 Communication

Perform this task to configure and verify a routing network for Stateless NAT64 communication. You can configure stateless NAT64 along with your NAT configuration: static, dynamic, or overload.

Before you begin

- An IPv6 address assigned to any host in the network should have a valid IPv4-translatable address and vice versa.
- You should enable the **ipv6 unicast-routing** command for this configuration to work.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateless** *ipv6-prefix*/*length*
16. **nat64 route** *ipv4-prefix*/*mask interface-type interface-number*
17. **ipv6 route** *ipv4-prefix*/*length interface-type interface-number*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 7	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8::1/128`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode.
Step 11	**description** *string* **Example:**	Adds a description to an interface configuration.

	Command or Action	Purpose
	`Device(config-if)# description interface facing ipv4`	
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 198.51.100.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 15	**nat64 prefix stateless** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateless 2001:0db8:0:1::/96`	Defines the Stateless NAT64 prefix to be added to the IPv4 hosts to translate the IPv4 address into an IPv6 address. • The command also identifies the prefix that must be used to create the IPv4-translatable addresses for the IPv6 hosts.
Step 16	**nat64 route** *ipv4-prefix/mask interface-type interface-number* **Example:** `Device(config)# nat64 route 203.0.113.0/24 gigabitethernet 0/0/0`	Routes the IPv4 traffic towards the correct IPv6 interface.
Step 17	**ipv6 route** *ipv4-prefix/length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0`	Routes the translated packets to the IPv4 address. • You must configure the **ipv6 route** command if your network is not running IPv6 routing protocols.
Step 18	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Multiple Prefixes for Stateless NAT64 Translation

Perform this task to configure multiple prefixes for Stateless NAT64 translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
6. **ipv6 enable**
7. **nat64 enable**
8. **nat64 prefix stateless v6v4** *ipv6-prefix*/*length*
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **negotiation auto**
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateless v4v6** *ipv6-prefix*/*length*
16. **nat64 route** *ipv4-prefix*/*mask interface-type interface-number*
17. **ipv6 route** *ipv6-prefix*/*length interface-type interface-number*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:**	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

	Command or Action	Purpose
	`Device(config-if)# ipv6 address 2001:DB8::1/128`	
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 7	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv6 interface.
Step 8	**nat64 prefix stateless v6v4** *ipv6-prefix/length* **Example:** `Device(config-if)# nat64 prefix stateless v6v4 2001:0db8:0:1::/96`	Maps an IPv6 address to an IPv4 host for Stateless NAT 64 translation. • The NAT64 prefix in the command is the same as the prefix of the source packet that is coming from the IPv6-to-IPv4 direction.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 203.0.113.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 12	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control on an interface.
Step 13	**nat64 enable** **Example:** `Router(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	
Step 15	**nat64 prefix stateless v4v6** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateless v4v6 2001:DB8:2::/96`	Maps an IPv4 address to an IPv6 host for Stateless NAT 64 translation. • This command identifies the prefix that creates the IPv4-translatable addresses for the IPv6 hosts.
Step 16	**nat64 route** *ipv4-prefix/mask interface-type interface-number* **Example:** `Device(config)# nat64 route 203.0.113.0/24 gigabitethernet 0/0/0`	Routes the IPv4 traffic towards the correct IPv6 interface.
Step 17	**ipv6 route** *ipv6-prefix/length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0`	Routes the translated packets to the IPv4 address. • You must configure the **ipv6 route** command if your network is not running IPv6 routing protocols.
Step 18	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining the Stateless NAT64 Routing Network

Perform this task to verify and monitor the Stateless NAT64 routing network. In the privileged EXEC mode, you can enter the commands in any order.

SUMMARY STEPS

1. **show nat64 statistics**
2. **show ipv6 route**
3. **show ip route**
4. **debug nat64** {**all** | **ha** {**all** | **info** | **trace** | **warn**} | **id-manager** | **info** | **issu** {**all** | **message** | **trace**} | **memory** | **statistics** | **trace** | **warn**}
5. **ping** [*protocol* [**tag**]] {*host-name* | *system-address*}

DETAILED STEPS

Step 1 show nat64 statistics

This command displays the global and interface-specific statistics of the packets that are translated and dropped.

Example:

```
Device# show nat64 statistics

NAT64 Statistics
Global Stats:
   Packets translated (IPv4 -> IPv6): 21
   Packets translated (IPv6 -> IPv4): 15
GigabitEthernet0/0/1 (IPv4 configured, IPv6 configured):
   Packets translated (IPv4 -> IPv6): 5
   Packets translated (IPv6 -> IPv4): 0
   Packets dropped: 0
GigabitEthernet1/2/0 (IPv4 configured, IPv6 configured):
   Packets translated (IPv4 -> IPv6): 0
   Packets translated (IPv6 -> IPv4): 5
   Packets dropped: 0
```

Step 2 **show ipv6 route**

This command displays the configured stateless prefix and the specific route for the IPv4 embedded IPv6 address pointing toward the IPv6 side.

Example:

```
Device# show ipv6 route

IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
ND - Neighbor Discovery
O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
LC  2001::1/128 [0/0] via FastEthernet0/3/4, receive
S   2001::1B01:10A/128 [1/0] via FastEthernet0/3/4, directly connected
S   3001::/96 [1/0] via ::42, NVI0
S   3001::1E1E:2/128 [1/0] via FastEthernet0/3/0, directly connected
LC  3001::C0A8:64D5/128 [0/0] via FastEthernet0/3/0, receive
L   FF00::/8 [0/0] via Null0, receive
```

Step 3 **show ip route**

This command displays the IPv4 addresses in the Internet that have reached the IPv4 side.

Example:

```
Device# show ip route

Codes: R - RIP derived, O - OSPF derived,
       C - connected, S - static, B - BGP derived,
       * - candidate default route, IA - OSPF inter area route,
       i - IS-IS derived, ia - IS-IS, U - per-user static route,
       o - on-demand routing, M - mobile, P - periodic downloaded static route,
       D - EIGRP, EX - EIGRP external, E1 - OSPF external type 1 route,
       E2 - OSPF external type 2 route, N1 - OSPF NSSA external type 1 route,
       N2 - OSPF NSSA external type 2 route
Gateway of last resort is 10.119.254.240 to network 10.140.0.0
O E2 10.110.0.0 [160/5] via 10.119.254.6, 0:01:00, Ethernet2
E    10.67.10.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
O E2 10.68.132.0 [160/5] via 10.119.254.6, 0:00:59, Ethernet2
O E2 10.130.0.0 [160/5] via 10.119.254.6, 0:00:59, Ethernet2
```

```
E    10.128.0.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.129.0.0 [200/129] via 10.119.254.240, 0:02:22, Ethernet2
E    10.65.129.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.10.0.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.75.139.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
E    10.16.208.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.84.148.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
E    10.31.223.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.44.236.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
E    10.141.0.0 [200/129] via 10.119.254.240, 0:02:22, Ethernet2
E    10.140.0.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
IPv6 Routing Table - default - 6 entries
```

Step 4 **debug nat64** {**all** | **ha** {**all** | **info** | **trace** | **warn**} | **id-manager** | **info** | **issu** {**all** | **message** | **trace**} | **memory** | **statistics** | **trace** | **warn**}

This command enables Stateless NAT64 debugging.

Example:

```
Device# debug nat64 statistics
```

Step 5 **ping** [*protocol* [**tag**]] {*host-name* | *system-address*}

The following is a sample packet capture from the IPv6 side when you specify the **ping 198.168.0.2** command after you configure the **nat64 enable** command on both the IPv4 and IPv6 interfaces:

Example:

```
Device# ping 198.168.0.2

Time             Source            Destination         Protocol       Info
1 0.000000       2001::c6a7:2      2001::c6a8:2        ICMPv6         Echo request
Frame 1: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
  Arrival Time: Oct 8, 2010 11:54:06.408354000 India Standard Time
  Epoch Time: 1286519046.408354000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.000000000 seconds]
  Frame Number: 1
  Frame Length: 118 bytes (944 bits)
  Capture Length: 118 bytes (944 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocol in frame: eth:1pv6:icmpv6: data]
Ethernet II, Src:Cisco_c3:64:94 (00:22:64:c3:64:94), Dst: Cisco_23:f2:30 (00:1f:6c:23:f2:30)
  Destination: Cisco_23:f2:30 (00:1f:6c:23:f2:30)
     Address: Cisco_23:f2:30 (00:1f:6c:23:f2:30)
     .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
     .... ...0 .... .... .... .... = LG bit: Globally unique address (factory default)
  Source: Cisco_c3:64:94 (00:22:64:c3:64:94)
    Address: Cisco_c3:64:94 (00:22:64:c3:64:94)
     .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
     .... ...0 .... .... .... .... = LG bit: Globally unique address (factory default)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6, src: 2001::c6a7:2 (2001::c6a7:2), Dst: 2001::c6a8:2 (2001::c6a8:2)
   0110 .... = Version: 6
     [0110 .... = This field makes the filter "ip.version ==6" possible:: 6]
  .... 0000 0000 ... .... .... .... .... = Traffic class: 0x00000000
    .... 0000 00.. .... .... .... .... .... = Differentiated Services Field: Default (0x00000000)
    .... .... ..0. .... .... .... ... .... = ECN-Capable Transport (ECT): Not set
```

```
  .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 64
  Next header: 64
  Hop limit: 64
  Source: 2001::c6a7:2 (2001::c6a7:2)
  [Source Teredo Server IPv4: 0.0.0.0 (0.0.0.0)]
  [Source Teredo Port: 6535]
  [Source Teredo Client IPv4: 198.51.100.1 (198.51.100.1)]
  Destination: 2001:c6a8:2 (2001::c6a8:2)
  [Destination Teredo Server IPv4: 0.0.0.0 (0.0.0.0)]
  [Destination Teredo Port: 65535]
  [Destination Teredo Client IPv4: 198.51.100.2 (198.51.100.2)]
Internet Control Message Protocol v6
  Type: 128 (Echo request)
  Code: 0 (Should always be zero)
  Checksum: 0xaed2 [correct]
  ID: 0x5018
  Sequence: 0x0000
  Data (56 bytes)
    Data: 069ae4c0d3b060008090a0b0c0d0e0f1011121314151617...
    [Length: 57]
```

Configuring a VRF for Stateless NAT64 Translation

Perform this task to configure a VRF for IPv4 to IPv6 prefix mapping.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
7. **ipv6 enable**
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **negotiation auto**
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateless v4 v6** *ipv6-prefix*/*length src-prefix* /*dst-prefix* **vrf***dst-prefix*
16. **nat64 route** *ipv4-prefix*/*mask interface-type interface-number*
17. **ipv6 route** *ipv6-prefix*/*length interface-type interface-number*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8::1/128`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 7	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 198.51.100.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 12	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control on an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 15	**nat64 prefix stateless v4 v6** *ipv6-prefix/length src-prefix /dst-prefix* **vrf***dst-prefix* **Example:** `Device(config)# nat64 prefix stateless 2001:0db8:0:1::/96`	Defines the Stateless NAT64 prefix to be added to the IPv4 hosts to translate the IPv4 address into an IPv6 address. • The command also identifies the prefix that must be used to create the IPv4-translatable addresses for the IPv6 hosts.
Step 16	**nat64 route** *ipv4-prefix/mask interface-type interface-number* **Example:** `Device(config)# nat64 route 203.0.113.0/24 gigabitethernet 0/0/0`	Routes the IPv4 traffic towards the correct IPv6 interface.
Step 17	**ipv6 route** *ipv6-prefix/length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0`	Routes the translated packets to the IPv4 address. • You must configure the **ipv6 route** command if your network is not running IPv6 routing protocols.
Step 18	**end** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configuration Examples for Stateless Network Address Translation 64

Example Configuring a Routing Network for Stateless NAT64 Translation

The following example shows how to configure a routing network for Stateless NAT64 translation:

```
ipv6 unicast-routing
!
interface gigabitethernet 0/0/0
 description interface facing ipv6
 ipv6 enable
 ipv6 address 2001:DB8::1/128
 nat64 enable
!

interface gigabitethernet 1/2/0
 description interface facing ipv4
 ip address 198.51.100.1 255.255.255.0
 nat64 enable
!

nat64 prefix stateless 2001:0db8:0:1::/96
nat64 route 203.0.113.0/24 gigabitethernet 0/0/0
ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0
```

Example: Configuring Multiple Prefixes for Stateless NAT64 Translation

```
ipv6 unicast-routing
!
interface gigabitethernet 0/0/0
 ipv6 address 2001:DB8::1/128
 ipv6 enable
 nat64 enable
 nat64 prefix stateless v6v4 2001:0db8:0:1::/96
!
interface gigabitethernet 1/2/0
 ip address 198.51.100.1 255.255.255.0
 negotiation auto
 nat64 enable
!
nat64 prefix stateless v4v6 2001:DB8:2::/96
nat64 route 203.0.113.0/24 gigabitethernet 0/0/0
ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0
```

Additional References for Stateless Network Address Translation 64

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Document Title
RFC 6052	IPv6 Addressing of IPv4/IPv6 Translators
RFC 6144	Framework for IPv4/IPv6 Translation
RFC 6145	IP/ICMP Translation Algorithm

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Glossary

ALG—application-layer gateway or application-level gateway.

FP—Forward Processor.

IPv4-converted address—IPv6 addresses used to represent the IPv4 hosts. These have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. Both stateless and stateful translators use IPv4-converted IPv6 addresses to represent the IPv4 hosts.

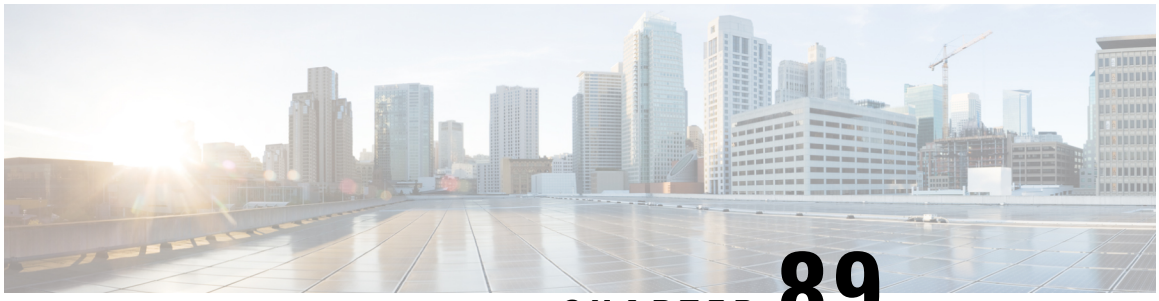
IPv6-converted address—IPv6 addresses that are assigned to the IPv6 hosts for the stateless translator. These IPv6-converted addresses have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. The stateless translator uses the corresponding IPv4 addresses to represent the IPv6 hosts. The stateful translator does not use IPv6-converted addresses, because the IPv6 hosts are represented by the IPv4 address pool in the translator via dynamic states.

NAT—Network Address Translation.

RP—Route Processor.

stateful translation—In stateful translation a per-flow state is created when the first packet in a flow is received. A translation algorithm is said to be stateful if the transmission or reception of a packet creates or modifies a data structure in the relevant network element. Stateful translation allows the use of multiple translators interchangeably and also some level of scalability. Stateful translation is defined to enable the IPv6 clients and peers without mapped IPv4 addresses to connect to the IPv4-only servers and peers.

stateless translation—A translation algorithm that is not stateful is called stateless. A stateless translation requires configuring a static translation table, or may derive information algorithmically from the messages it is translating. Stateless translation requires less computational overhead than stateful translation. It also requires less memory to maintain the state, because the translation tables and the associated methods and processes exist in a stateful algorithm and do not exist in a stateless one. Stateless translation enables the IPv4-only clients and peers to initiate connections to the IPv6-only servers or peers that are equipped with IPv4-embedded IPv6 addresses. It also enables scalable coordination of IPv4-only stub networks or ISP IPv6-only networks. Because the source port in an IPv6-to-IPv4 translation may have to be changed to provide adequate flow identification, the source port in the IPv4-to-IPv6 direction need not be changed.



CHAPTER 89

Stateful Network Address Translation 64

The Stateful Network Address Translation 64 feature provides a translation mechanism that translates IPv6 packets into IPv4 packets and vice versa. The stateful NAT64 translator algorithmically translates the IPv4 addresses of IPv4 hosts to and from IPv6 addresses by using the configured stateful prefix. In a similar manner, the IPv6 addresses of IPv6 hosts are translated to and from IPv4 addresses through Network Address Translation (NAT). Stateful Network Address Translation 64 (NAT64) also translates protocols and IP addresses. The Stateful NAT64 translator enables native IPv6 or IPv4 communication and facilitates coexistence of IPv4 and IPv6 networks.

This document explains how Stateful NAT64 works and how to configure your network for Stateful NAT64 translation.

Prerequisites for Configuring Stateful Network Address Translation 64

- For Domain Name System (DNS) traffic to work, you must have a separate working installation of DNS64.

Restrictions for Configuring Stateful Network Address Translation 64

- Applications without a corresponding application-level gateway (ALG) may not work properly with the Stateful NAT64 translator.
- IP Multicast is not supported.

- The translation of IPv4 options, IPv6 routing headers, hop-by-hop extension headers, destination option headers, and source routing headers is not supported.
- Virtual routing and forwarding (VRF)-aware NAT64 is not supported.
- When traffic flows from IPv6 to IPv4, the destination IP address that you have configured must match a stateful prefix to prevent hairpinning loops. However, the source IP address (source address of the IPv6 host) must not match the stateful prefix. If the source IP address matches the stateful prefix, packets are dropped.

  Hairpinning allows two endpoints inside Network Address Translation (NAT) to communicate with each other, even when the endpoints use only each other's external IP addresses and ports for communication.

- Only TCP and UDP Layer 4 protocols are supported for header translation.
- Routemaps are not supported.
- Application-level gateways (ALGs) FTP and ICMP are not supported.
- In the absence of a pre-existing state in NAT 64, stateful translation only supports IPv6-initiated sessions.
- If a static mapping host-binding entry exists for an IPv6 host, the IPv4 nodes can initiate communication. In dynamic mapping, IPv4 nodes can initiate communication only if a host-binding entry is created for the IPv6 host through a previously established connection to the same or a different IPv4 host.

  Dynamic mapping rules that use Port-Address Translation (PAT), host-binding entries cannot be created because IPv4-initiated communication not possible through PAT.

- Both NAT44 (static, dynamic and PAT) configuration and stateful NAT64 configuration are not supported on the same interface.

Information About Stateful Network Address Translation 64

Stateful Network Address Translation 64

The Stateful NAT64 feature provides a translation mechanism that translates IPv6 packets into IPv4 packets and vice versa.

Stateful NAT64 supports TCP, and UDP traffic. Packets that are generated in an IPv6 network and are destined for an IPv4 network are routed within the IPv6 network towards the Stateful NAT64 translator. Stateful NAT64 translates the packets and forwards them as IPv4 packets through the IPv4 network. The process is reversed for traffic that is generated by hosts connected to the IPv4 network and destined for an IPv6 receiver.

The Stateful NAT64 translation is not symmetric, because the IPv6 address space is larger than the IPv4 address space and a one-to-one address mapping is not possible. Before it can perform an IPv6 to an IPv4 translation, Stateful NAT64 requires a state that binds the IPv6 address and the TCP/UDP port to the IPv4 address. The binding state is either statically configured or dynamically created when the first packet that flows from the IPv6 network to the IPv4 network is translated. After the binding state is created, packets flowing in both directions are translated. In dynamic binding, Stateful NAT64 supports communication initiated by the IPv6-only node toward an IPv4-only node. Static binding supports communication initiated by an IPv4-only node to an IPv6-only node and vice versa. Stateful NAT64 with NAT overload or Port Address Translation (PAT) provides a 1:*n* mapping between IPv4 and IPv6 addresses.

When an IPv6 node initiates traffic through Stateful NAT64, and the incoming packet does not have an existing state and the following events happen:

- The source IPv6 address (and the source port) is associated with an IPv4 configured pool address (and port, based on the configuration).

- The destination IPv6 address is translated mechanically based on the BEHAVE translation draft using either the configured NAT64 stateful prefix or the Well Known Prefix (WKP).
- The packet is translated from IPv6 to IPv4 and forwarded to the IPv4 network.

When an incoming packet is stateful (if a state exists for an incoming packet), NAT64 identifies the state and uses the state to translate the packet.

Prefixes Format for Stateful Network Address Translation 64

A set of bits at the start of an IPv6 address is called the format prefix. Prefix length is a decimal value that specifies how many of the leftmost contiguous bits of an address comprise the prefix.

When packets flow from the IPv6 to the IPv4 direction, the IPv4 host address is derived from the destination IP address of the IPv6 packet that uses the prefix length. When packets flow from the IPv4 to the IPv6 direction, the IPv4 host address is constructed using the stateful prefix.

According to the IETF address format BEHAVE draft, a u-bit (bit 70) defined in the IPv6 architecture should be set to zero. For more information on the u-bit usage, see RFC 2464. The reserved octet, also called u-octet, is reserved for compatibility with the host identifier format defined in the IPv6 addressing architecture. When constructing an IPv6 packet, the translator has to make sure that the u-bits are not tampered with and are set to the value suggested by RFC 2373. The suffix will be set to all zeros by the translator. IETF recommends that the 8 bits of the u-octet (bit range 64–71) be set to zero.

Well Known Prefix

The Well Known Prefix 64:FF9B::/96 is supported for Stateful NAT64. During a stateful translation, if no stateful prefix is configured (either on the interface or globally), the WKP prefix is used to translate the IPv4 host addresses.

Stateful IPv4-to-IPv6 Packet Flow

The packet flow of IPv4-initiated packets for Stateful NAT64 is as follows:

- The destination address is routed to a NAT Virtual Interface (NVI).

  A virtual interface is created when Stateful NAT64 is configured. For Stateful NAT64 translation to work, all packets must get routed to the NVI. When you configure an address pool, a route is automatically added to all IPv4 addresses in the pool. This route automatically points to the NVI.

- The IPv4-initiated packet hits static or dynamic binding.

  Dynamic address bindings are created by the Stateful NAT64 translator when you configure dynamic Stateful NAT64. A binding is dynamically created between an IPv6 and an IPv4 address pool. Dynamic binding is triggered by the IPv6-to-IPv4 traffic and the address is dynamically allocated. Based on your configuration, you can have static or dynamic binding.

- The IPv4-initiated packet is protocol-translated and the destination IP address of the packet is set to IPv6 based on static or dynamic binding. The Stateful NAT64 translator translates the source IP address to IPv6 by using the Stateful NAT64 prefix (if a stateful prefix is configured) or the Well Known Prefix (WKP) (if a stateful prefix is not configured).
- A session is created based on the translation information.

All subsequent IPv4-initiated packets are translated based on the previously created session.

Stateful IPv6-to-IPv4 Packet Flow

The stateful IPv6-initiated packet flow is as follows:

- The first IPv6 packet is routed to the NAT Virtual Interface (NVI) based on the automatic routing setup that is configured for the stateful prefix. Stateful NAT64 performs a series of lookups to determine whether the IPv6 packet matches any of the configured mappings based on an access control list (ACL) lookup. Based on the mapping, an IPv4 address (and port) is associated with the IPv6 destination address. The IPv6 packet is translated and the IPv4 packet is formed by using the following methods:
  - Extracting the destination IPv4 address by stripping the prefix from the IPv6 address. The source address is replaced by the allocated IPv4 address (and port).
  - The rest of the fields are translated from IPv6-to-IPv4 to form a valid IPv4 packet.

Note This protocol translation is the same for stateless NAT64.

- A new NAT64 translation is created in the session database and in the bind database. The pool and port databases are updated depending on the configuration. The return traffic and the subsequent traffic of the IPv6 packet flow will use this session database entry for translation.

IP Packet Filtering

Stateful Network Address Translation 64 (NAT64) filters IPv6 and IPv4 packets. All IPv6 packets that are transmitted into the stateful translator are filtered because statefully translated IPv6 packets consume resources in the translator. These packets consume processor resources for packet processing, memory resources (always session memory) for static configuration, IPv4 address resources for dynamic configuration, and IPv4 address and port resources for Port Address Translation (PAT).

Stateful NAT64 utilizes configured access control lists (ACLs) and prefix lists to filter IPv6-initiated traffic flows that are allowed to create the NAT64 state. Filtering of IPv6 packets is done in the IPv6-to-IPv4 direction because dynamic allocation of mapping between an IPv6 host and an IPv4 address can be done only in this direction.

Stateful NAT64 supports endpoint-dependent filtering for the IPv4-to-IPv6 packet flow with PAT configuration. In a Stateful NAT64 PAT configuration, the packet flow must have originated from the IPv6 realm and created the state information in NAT64 state tables. Packets from the IPv4 side that do not have a previously created state are dropped. Endpoint-independent filtering is supported with static Network Address Translation (NAT) and non-PAT configurations.

Differences Between Stateful NAT64 and Stateless NAT64

The table below displays the differences between Stateful NAT64 and Stateless NAT64.

Table 130: Differences Between Stateful NAT64 and Stateless NAT64

Supported Features	Stateful NAT64	Stateless NAT64
Address savings	*N*:1 mapping for PAT or overload configuration that saves IPv4 addresses.	One-to-one mapping—one IPv4 address is used for each IPv6 host).
Address space	IPv6 systems may use any type of IPv6 addresses.	IPv6 systems must have IPv4-translatable addresses (based on RFC 6052).
ALGs supported	FTP64	None
Protocols supported	ICMP, TCP, UDP	All
Standards	Draft-ieft-behave-v6v4-xlate-stateful-12	Draft-ietf-behave-v6v4-xlate-05
State creation	Each traffic flow creates a state in the NAT64 translator. The maximum number of states depends on the number of supported translations.	Traffic flow does not create any state in the NAT64 translator. Algorithmic operation is performed on the packet headers.

High-Speed Logging for NAT64

When HSL is configured, NAT64 provides a log of packets that flow through routing devices (similar to the Version 9 NetFlow-like records) to an external collector. Records are sent for each binding (binding is the address binding between the local address and the global address to which the local address is translated) and when sessions are created and destroyed. Session records contain the full 5-tuple of information (the source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements. NAT64 also sends an HSL message when a NAT64 pool runs out of addresses (also called pool exhaustion). Because the pool exhaustion messages are rate limited, each packet that hits the pool exhaustion condition does not trigger an HSL message. Depending on your release, Stateful NAT64 supports high-speed logging (HSL) for upto 4 destinations.

Configure the **nat64 logging translations flow-export v9 udp destination** command to enable NAT64 HSL logging. The **vrf** keyword can be used to enable NAT64 HSL for a specific VRF

The table below describes the templates for HSL bind and session create or destroy. These fields (in the order they are displayed in the log) describe how the log collector must interpret the bytes in HSL records. The value for some of the fields varies based on whether the session is being created, destroyed, or modified.

Table 131: Templates for HSL Bind and Session Create or Destroy

Field	Format	ID	Value
Original IPv6 address	IPv6 address	27	Varies
Translated IPv4 address	IPv6 address	282	Varies
Translated IPv6 address	IPv4 address	225	Varies
Original IPv4 address	IPv4 address	12	Varies
Original IPv6 port	16-bit port	7	Varies

Field	Format	ID	Value
Translated IPv6 port	16-bit port	227	Varies
Translated IPv4 port	16-bit port	11	Varies
Original IPv4 port	16-bit port	228	Varies
Timestamp for an event	64 bits - milliseconds (This is a 64-bit field that holds the UNIX time, in milliseconds, when the event for the record occurred.)	323	Varies
VRF ID	32-bit ID	234	Zero
Protocol	8-bit value	4	Varies
Event	8-bit value	230	0–Invalid 1–Add event 2–Delete event

The table below describes the HSL pool exhaustion templates (in the order they are available in the template).

Table 132: Templates for HSL Pool Exhaustion

Field	Format	ID	Values
NAT pool ID	32-bit value	283	Varies
NAT event	8-bit value	230	3–Pool exhaust

How to Configure Enabling NAT64 High-Speed Logging per VRF

Enabling High-Speed Logging of NAT64 Translations

You can enable or disable high-speed logging (HSL) of all NAT64 translations or only translations for specific VPNs.

You must first use the **nat64 logging translations flow-export v9 udp destination** command to enable HSL for all VPN and non-VPN translations. The **vrf** keyword can be used to specify HSL destination address on a specific VRF. VPN translations are also known as Virtual Routing and Forwarding (VRF) translations.

After you enable HSL for all NAT translations, you can then use the **nat64 logging translations flow-export v9** *vrf-name* command to enable or disable translations for specific VPNs. When you use this command, HSL is disabled for all VPNs, except for the ones the command is explicitly enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **nat64 logging translations flow-export v9 udp destination** *addr*|**ipv6-destination** *IPv6 address* **vrf***vrf name* **source** *interface type interface-number*
4. **nat64 logging translations flow-export v9** {*vrf-name* | **global-on**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**nat64 logging translations flow-export v9 udp destination** *addr*|**ipv6-destination** *IPv6 address* **vrf***vrf name* **source** *interface type interface-number*
	Example:
	This example shows how to enable high-speed logging using an IPv4 address
	`Device(config)# nat64 logging translations flow-export v9 udp destination 10.10.0.1 1020 source GigabitEthernet 0/0/0`
	Example:
	This example shows how to enable high-speed logging using an IPv6 address
	`Device(config)# nat64 logging translations flow-export v9 udp ipv6-destination 2001::06 5050 source GigabitEthernet 0/0/0`
	Example:
	This example shows how to enable high-speed logging using an IPv6 address for a destination VRF
	`Device(config)# nat64 logging translations flow-export v9 udp ipv6-destination 2001::06 5050 vrf hslvrf source GigabitEthernet 0/0/0`
		Enables the high-speed logging of all VPN and non-VPN translations for up to four destinations. You can enable logging for a specific destination VRF using the **vrf** keyword. To specify an IPv6 address for the UDP destination, use the **ipv6-destination** keyword followed by the IPv6 address.

Step 4	**nat64 logging translations flow-export v9** {*vrf-name*	**global-on**} **Example:** `Device(config)# nat64 logging translations flow-export v9 VPN-18`	Enables or disables the high-speed logging of specific NAT VPN translations.
Step 5	**exit** **Example:** `Device(config)# exit`	(Optional) Exits global configuration mode and enters privileged EXEC mode.

FTP64 Application-Level Gateway Support

The FTP64 (or service FTP) application-level gateway (ALG) helps stateful Network Address Translation 64 (NAT64) to operate on Layer 7 data. FTP64 ALG translates IP addresses and the TCP port information embedded in the payload of an FTP control session.

NAT translates any TCP/UDP traffic that does not carry source and destination IP addresses in the application data stream. Protocols that embed the IP address information within the payload (or in the application data stream) require the support of an ALG. ALGs handle application data stream (Layer 7) protocol-specific services, such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection or session information from control channels.

FTP64 is automatically enabled when Stateful NAT64 is enabled. Use the **no nat64 service ftp** command to disable the NAT64 FTP service.

Note The FTP64 ALG is not supported in Stateless NAT64 translation.

Note The FTP64 ALG does not support IPv4-compatible IPv6 addresses.

Based on *IPv6-to-IPv4 translation FTP considerations draft-ietf-behave-ftp64-02* and RFC 2228, the FTP64 ALG must switch to transparent mode (a device in a transparent mode is invisible in the network; however, this device can act as a bridge and inspect or filter packets), when commands and responses flow between the FTP client and the FTP server. When a client issues the FTP AUTH command, the FTP64 ALG transparently forwards all data on the control channel in both (ingress and egress) directions, until the end of the control channel session. Similarly, during an AUTH negotiation, the ALG must be in transparent mode, whether the negotiation is successful or not.

Based on RFC 6384, the behavior of the FTP64 ALG during a client-server communication is different. During an IPv6-to-IPv4 translation, the FTP64 ALG must transparently copy data transmitted over the control channel so that the transport layer security (TLS) session works correctly. However, the client commands and server responses are hidden from the FTP64 ALG. To ensure a consistent behavior, as soon as the initial FTP AUTH command is issued by a client, the FTP64 ALG must stop translating commands and responses and start transparently copying TCP data that is sent by the server to the client and vice versa. The FTP64 ALG must ignore the AUTH command and not go into transparent mode if the server response is in the 4*xx* or 5*xx* ranges, which comprise FTP error/warning messages.

Prior to CSCtu37975, when an IPv6 FTP client issues an FTP AUTH command, irrespective of whether the IPv4 FTP server accepts or rejects that authorization negotiation, the FTP64 ALG moves the AUTH session to transparent mode (or bypass mode). When a session is in transparent mode, NAT cannot perform translation on the packets within the session. With CSCtu37975, during a client-server communication, the FTP64 ALG's behavior is compliant with RFC 6384.

FTP64 NAT ALG Intrabox High Availability Support

Depending on your release, the FTP64 application-level gateway (ALG) adds high availability (HA) support for Stateful NAT64. The FTP64 NAT ALG Intrabox HA Support feature supports the stateful switchover between redundant Forward Processors (FPs) within a single chassis. The HA support provided by the FTP64 ALG is applicable to both intrabox HA and In-Service Software Upgrade (ISSU).

Use the **no nat64 service ftp** command to disable the NAT64 ALG service.

The FTP64 ALG synchronizes data when it receives the following messages:

- User authentication flag after 230 replies.
- ALG disable/enable flag after ALG ENABLE and ALG DISABLE messages are received.
- Fragment detection information after the first segmented packet is detected.
- Fragment detection information after the end of the segmentation is detected.

Note

- Stateful NAT64 supports only intrabox HA in some releases.
- FTP64 ALG statistics and FTP64 debug logs are not synchronized to the standby device by the FTP64 ALG.

Stateful NAT64—Intrachassis Redundancy

Depending on your release, support for the Stateful NAT64—Intrachassis Redundancy feature is available. When a second Forward Processor (FP) is available inside a single chassis, the Stateful NAT64—Intrachassis Redundancy feature enables you to configure the second FP as a standby entity. When you plug in the second FP, redundancy starts automatically with no explicit configuration. There is a short delay before the standby FP becomes the “hot standby” (which means that all sessions have been synchronized). The standby FP maintains a backup of the Stateful NAT64 session information, and when the active (first) FP fails, there is very little disruption of NAT64 sessions.

NAT64 redundancy information is sent to the standby FP in the following instances:

- When a session or a dynamic bind is created.
- When a session or a dynamic bind is deleted.
- During periodic updates. Based on the time elapsed, the active FP periodically updates the state information to the standby. Not all changes in the replicated objects are sent immediately to the standby at the time of change. The most critical updates are sent immediately, and other changes are communicated by periodic updates.

When a standby FP is inserted or when a standby FP recovers from a reload, the active FP performs a bulk synchronization to synchronize the standby FP with the active FP. NAT does an aggressive synchronization by which the active FP pushes all the state information forcefully to the standby FP.

In addition to NAT64 session information, application-specific information (application-level gateway [ALG] information) also has to be communicated to the standby FP. Each ALG has a per-session state that needs to be synchronized in the standby. The ALG triggers the sending of all ALG state information to the standby FP. NAT provides the mechanism for actually sending the ALG state and associates the state to a particular session.

HTTP sessions are not backed up on the standby FP. To replicate HTTP sessions on the standby FP during a switchover, you must configure the **nat64 switchover replicate http enable** command.

Note The Stateful NAT64—Intrachassis Redundancy feature does not support box-to-box (B2B) redundancy or asymmetric routing.

Asymmetric Routing Support for NAT64

In Cisco IOS XE Release and later releases, Network Address Translation 64 (NAT64) supports asymmetric routing and asymmetric routing with Multiprotocol Label Switching (MPLS). In NAT 64, MPLS is enabled on the IPv4 interface. Packets coming from the IPv6 interface are switched to the IPv4 interface. No configuration changes are required to enable asymmetric routing or asymmetric routing with MPLS.

For more information, see the section "Example: Configuring Asymmetric Routing Support for NAT64".

How to Configure Stateful Network Address Translation 64

Based on your network configuration, you can configure static, dynamic, or dynamic Port Address Translation (PAT) Stateful NAT64.

Note You need to configure at least one of the configurations described in the following tasks for Stateful NAT64 to work.

Configuring Static Stateful Network Address Translation 64

You can configure a static IPv6 address to an IPv4 address and vice versa. Optionally, you can configure static Stateful NAT64 with or without ports. Perform this task to configure static Stateful NAT64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateful** *ipv6-prefix/length*

16. **nat64 v6v4 static** *ipv6-address ipv4-address*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 7	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:**	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface gigabitethernet 1/2/0`	
Step 11	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**nat64 prefix stateful** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateful 2001:DB8:1::1/96`	Defines the Stateful NAT64 prefix to be added to IPv4 hosts to translate the IPv4 address into an IPv6 address. • The Stateful NAT64 prefix can be configured at the global configuration level or at the interface level.
Step 16	**nat64 v6v4 static** *ipv6-address ipv4-address* **Example:** `Device(config)# nat64 v6v4 static 2001:DB8:1::FFFE 209.165.201.1`	Enables NAT64 IPv6-to-IPv4 static address mapping.
Step 17	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring Dynamic Stateful Network Address Translation 64

A dynamic Stateful NAT64 configuration provides a one-to-one mapping of IPv6 addresses to IPv4 addresses in the address pool. You can use the dynamic Stateful NAT64 configuration when the number of active IPv6 hosts is less than the number of IPv4 addresses in the pool. Perform this task to configure dynamic Stateful NAT64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**

4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **ipv6 access-list** *access-list-name*
16. **permit ipv6** *ipv6-address* **any**
17. **exit**
18. **nat64 prefix stateful** *ipv6-prefix/length*
19. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
20. **nat64 v6v4 list** *access-list-name* **pool** *pool-name*
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:**	Enables IPv6 processing on an interface.

	Command or Action	Purpose
	`Device(config-if)# ipv6 enable`	
Step 7	**ipv6** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Device(config-if)# ipv6 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode
Step 11	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.24 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list nat64-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 16	**permit ipv6** *ipv6-address* **any** **Example:** `Device(config-ipv6-acl)# permit ipv6 2001:DB8:2::/96 any`	Sets permit conditions for an IPv6 access list.

	Command or Action	Purpose
Step 17	**exit** **Example:** `Device(config-ipv6-acl# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 18	**nat64 prefix stateful** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateful 2001:DB8:1::1/96`	Enables NAT64 IPv6-to-IPv4 address mapping.
Step 19	**nat64 v4 pool** *pool-name start-ip-address end-ip-address* **Example:** `Device(config)# nat64 v4 pool pool1 209.165.201.1 209.165.201.254`	Defines the Stateful NAT64 IPv4 address pool.
Step 20	**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **Example:** `Device(config)# nat64 v6v4 list nat64-acl pool pool1`	Dynamically translates an IPv6 source address to an IPv6 source address and an IPv6 destination address to an IPv4 destination address for NAT64.
Step 21	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring Dynamic Port Address Translation Stateful NAT64

A Port Address Translation (PAT) or overload configuration is used to multiplex (mapping IPv6 addresses to a single IPv4 pool address) multiple IPv6 hosts to a pool of available IPv4 addresses on a first-come first-served basis. The dynamic PAT configuration conserves the IPv4 address space while providing connectivity to the IPv4 Internet. Configure the **nat64 v6v4 list** command with the **overload** keyword to configure PAT address translation. Perform this task to configure dynamic PAT Stateful NAT64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*

13. **nat64 enable**
14. **exit**
15. **ipv6 access-list** *access-list-name*
16. **permit ipv6** *ipv6-address* **any**
17. **exit**
18. **nat64 prefix stateful** *ipv6-prefix/length*
19. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
20. **nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 7	**ipv6** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Device(config-if)# ipv6 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode
Step 11	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.24 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list nat64-acl`	Defines an IPv6 access list and places the device in IPv6 access list configuration mode.
Step 16	**permit ipv6** *ipv6-address* **any** **Example:** `Device(config-ipv6-acl)# permit ipv6 2001:db8:2::/96 any`	Sets permit conditions for an IPv6 access list.
Step 17	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 18	**nat64 prefix stateful** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateful 2001:db8:1::1/96`	Enables NAT64 IPv6-to-IPv4 address mapping.

	Command or Action	Purpose
Step 19	**nat64 v4 pool** *pool-name start-ip-address end-ip-address* **Example:** `Device(config)# nat64 v4 pool pool1 209.165.201.1 209.165.201.254`	Defines the Stateful NAT64 IPv4 address pool.
Step 20	**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload** **Example:** `Device(config)# nat64 v6v4 list nat64-acl pool pool1 overload`	Enables NAT64 PAT or overload address translation.
Step 21	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Restrictions for Enabling Stateful Network Address Conversion using VRF

- One to one static IPv6 and IPv4 mapping is not supported.
- Inter VRF translations are not supported with NAT64.
- NAT64 pools and NAT44 pools do not support IP address sharing.

Configuring VRF Aware Stateful NAT64 with Carrier Grade NAT

A Port Address Translation (PAT) or overload configuration is used to multiplex multiple IPv6 hosts to a pool of available IPv4 addresses on a first come first served basis. The dynamic PAT configuration conserves the IPv4 address space while providing connectivity to the IPv4 Internet. Configuring the **nat64 v6v4** list command with the **overload** keyword, configures the PAT address translation and the **vrf** keyword segregates the incoming and outgoing traffic in the network.

Perform this task to enable stateful NAT64 conversion within a Virtual routing and forwarding (VRF) network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **vrf forwarding***vrf-name*
7. **ipv6 enable**
8. **ipv6** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
9. **nat64 enable**
10. **exit**
11. **interface** *type number*

12. **description** *string*
13. **vrf forwarding***vrf-name*
14. **ip address** *ip-address mask*
15. **nat64 enable**
16. **exit**
17. **ipv6 access-list** *access-list-name*
18. **permit ipv6** *ipv6-address* **any**
19. **exit**
20. **nat64 settings mode cgn**
21. **nat64 prefix stateful** *ipv6-prefix/length/vrf*
22. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
23. **nat64 v6v4 list** *access-list-name* **pool** *pool-name* **vrf***vrf-name***overloadmatch-in-vrf**
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast data packets.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**vrf forwarding***vrf-name* **Example:** `Device(config-if)# vrf forwarding test`	Enables IPv6 VRF routing.
Step 7	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.

	Command or Action	Purpose
Step 8	**ipv6** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Device(config-if)# ipv6 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 9	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode
Step 12	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 13	**vrf forwarding***vrf-name* **Example:** `Device(config-if)# vrf forwarding test`	Enables IPv4 VRF routing.
Step 14	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.24 255.255.255.0`	Configures an IPv4 address for an interface.
Step 15	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.
Step 16	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 17	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list nat64-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 18	**permit ipv6** *ipv6-address* **any** **Example:**	Sets permit conditions for an IPv6 access list.

	Command or Action	Purpose
	`Device(config-ipv6-acl)# permit ipv6 2001:DB8:2::/96 any`	
Step 19	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.

Step 20	**nat64 settings mode cgn**
	Example:
	`Device(config)# nat64 settings mode cgn`
	Example:
	`Device(config)# no nat64 settings mode cgn`
		Enables the CGN operating mode.
		• To disable the CGN operating mode, use the **no** form of this command.

Step 21	**nat64 prefix stateful** *ipv6-prefix/length/vrf* **Example:** `Device(config)# nat64 prefix stateful 2001:DB8:1::1/96 vrf 1`	Enables NAT64 IPv6-to-IPv4 address mapping.
Step 22	**nat64 v4 pool** *pool-name start-ip-address end-ip-address* **Example:** `Device(config)# nat64 v4 pool pool1 209.165.201.1 209.165.201.254`	Defines the stateful NAT64 IPv4 address pool.
Step 23	**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **vrf***vrf-name***overloadmatch-in-vrf** **Example:** `Device(config)# nat64 v6v4 list nat64-acl pool pool1 vrf 1 overload match-in-vrf`	Enables NAT64 conversion for a VRF network.
Step 24	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Verifying VRF Aware Stateful NAT64 with Carrier Grade NAT (CGN)

SUMMARY STEPS

1. **show nat64 translations**
2. **show ip route vrf***vrf*
3. **show ipv6 route vrf***vrf*
4. **show platform hardware qfp active feature nat64 data statistics**
5. **show platform hardware qfp active feature nat64 data sess-dump**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show nat64 translations** **Example:** `Device# show nat64 translations`	Displays all NAT 64 protocol information.
Step 2	**show ip route vrf***vrf* **Example:** `Device# show ip route vrf VRF1`	Displays the IP routing table associated with a VRF.
Step 3	**show ipv6 route vrf***vrf* **Example:** `Device(config)# show ipv6 route vrf VRF1`	Displays IPv6 routing table information associated with a VPN routing and forwarding (VRF) instance.
Step 4	**show platform hardware qfp active feature nat64 data statistics** **Example:** `Device(config)# show platform hardware qfp active feature nat64 data statistics`	Displays all NAT64 translation global statistics.
Step 5	**show platform hardware qfp active feature nat64 data sess-dump** **Example:** `Device(config)# show platform hardware qfp active feature nat64 data sess-dump`	Displays all NAT64 session dump information.

Monitoring and Maintaining a Stateful NAT64 Routing Network

Use the following commands in any order to display the status of your Stateful Network Address Translation 64 (NAT64) configuration.

SUMMARY STEPS

1. **show nat64 aliases** [*lower-address-range upper-address-range*]
2. **show nat64 logging**
3. **show nat64 prefix stateful** {**global** | {**interfaces** | **static-routes**} [**prefix** *ipv6-address*/*prefix-length*]}
4. **show nat64 timeouts**

DETAILED STEPS

Step 1 **show nat64 aliases** [*lower-address-range upper-address-range*]

This command displays the IP aliases created by NAT64.

Example:

```
Device# show nat64 aliases

Aliases configured: 1
Address   Table ID  Inserted   Flags   Send ARP   Reconcilable   Stale   Ref-Count
10.1.1.1  0         FALSE      0x0030  FALSE      TRUE           FALSE   1
```

Step 2 **show nat64 logging**

This command displays NAT64 logging.

Example:

```
Device# show nat64 logging

NAT64 Logging Type

Method       Protocol   Dst. Address   Dst. Port   Src. Port
translation
flow export  UDP        10.1.1.1       5000        60087
```

Step 3 **show nat64 prefix stateful** {**global** | {**interfaces** | **static-routes**} [**prefix** *ipv6-address/prefix-length*]}

This command displays information about NAT64 stateful prefixes.

Example:

```
Device# show nat64 prefix stateful interfaces

Stateful Prefixes

Interface               NAT64   Enabled   Global Prefix
GigabitEthernet0/1/0    TRUE    TRUE      2001:DB8:1:1/96
GigabitEthernet0/1/3    TRUE    FALSE     2001:DB8:2:2/96
```

Step 4 **show nat64 timeouts**

This command displays statistics for NAT64 translation session timeout.

Example:

```
Device# show nat64 timeouts

NAT64 Timeout

Seconds    CLI Cfg    Uses 'All'    all flows
86400      FALSE      FALSE         udp
300        FALSE      TRUE          tcp
7200       FALSE      TRUE          tcp-transient
240        FALSE      FALSE         icmp
60         FALSE      TRUE
```

Configuration Examples for Stateful Network Address Translation 64

Example: Configuring Static Stateful Network Address Translation 64

```
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# description interface facing ipv6
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 address 2001:DB8:1::1/96
Device(confif-if)# nat64 enable
Device(config-fi)# exit
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# description interface facing ipv4
Device(config-if)# ip address 209.165.201.1 255.255.255.0
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# nat64 prefix stateful 2001:DB8:1::1/96
Device(config)# nat64 v6v4 static 2001:DB8:1::FFFE 209.165.201.1
Device(config)# end
```

Example: Configuring Dynamic Stateful Network Address Translation 64

```
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# description interface facing ipv6
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 2001:DB8:1::1/96
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# description interface facing ipv4
Device(config-if)# ip address 209.165.201.24 255.255.255.0
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# ipv6 access-list nat64-acl
Device(config-ipv6-acl)# permit ipv6 2001:db8:2::/96 any
Device(config-ipv6-acl)# exit
Device(config)# nat64 prefix stateful 2001:db8:1::1/96
Device(config)# nat64 v4 pool pool1 209.165.201.1 209.165.201.254
Device(config)# nat64 v6v4 list nat64-acl pool pool1
Device(config)# end
```

Example: Configuring Dynamic Port Address Translation Stateful NAT64

```
enable
 configure terminal
  ipv6 unicast-routing
```

```
interface gigabitethernet 0/0/0
 description interface facing ipv6
 ipv6 enable
 ipv6 2001:DB8:1::1/96
 nat64 enable
 exit
interface gigabitethernet 1/2/0
 description interface facing ipv4
 ip address 209.165.201.24 255.255.255.0
 nat64 enable
 exit
ipv6 access-list nat64-acl
 permit ipv6 2001:db8:2::/96 any
 exit
nat64 prefix stateful 2001:db8:1::1/96
nat64 v4 pool pool1 209.165.201.1 209.165.201.254
nat64 v6v4 list nat64-acl pool pool1 overload
end
```

Example: Configuring Asymmetric Routing Support for NAT64

The following example shows how to configure asymmetric routing for Network Address Translation 64 (NAT64):

```
!RouterA Configuration

Device(config)# ipv6 unicast-routing
Device(config)# nat64 prefix stateful 2001:db8:2::/96
Device(config)# nat64 v6v4 static 2001:db8:1::5 150.0.0.1 redundancy 1 mapping-id 150
Device(config)# nat64 v6v4 static 2001:db8:1::6 150.0.0.2 redundancy 1 mapping-id 151
Device(config)# nat64 switchover replicate http enable port 80
!
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name RG1
Device(config-red-app-grp)# data gigabitethernet 1/1/0
Device(config-red-app-grp)# control gigabitethernet 1/1/1 protocol 1
Device(config-red-app-grp)# asymmetric-routing interface gigabitethernet 1/1/2
Device(config-red-app-grp)# priority 150 failover threshold 140
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# exit
Device(config-red-app)# exit
Device(config-red)# exit
!
Device(config)# interface gigabitethernet 1/1/0
Device(config-if)# ip address 10.10.10.1 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# exit
!
Device(config)# interface gigabitethernet 1/1/1
Device(config-if)# ip address 172.16.0.1 255.240.0.0
Device(config-if)# no shutdown
Device(config-if)# exit
!
Device(config)# interface gigabitethernet 1/1/2
Device(config-if)# ip address 192.168.0.1 255.255.0.0
Device(config-if)# no shutdown
Device(config-if)# exit
```

```
!
Device(config)# interface gigabitethernet 1/1/3
Device(config-if)# ipv6 enable
Device(config-if)# no shutdown
Device(config-if)# nat64 enable
Device(config-if)# ipv6 address 2001:db8:1::2/96
Device(config-if)# redundancy rii 100
Device(config-if)# redundancy group 1 ipv6 2001:db8:1::1/96 exclusive decrement 15
Device(config-if)# exit
!
Device(config)# interface gigabitethernet 1/1/4
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# nat64 enable
Device(config-if)# no shutdown
Device(config-if)# redundancy rii 101
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# exit
!
Device(config)# router ospf 90
Device(config-router)# network 192.0.2.0 255.255.255.0 area 0
Device(config-router)# end


! Router B Configuration

Device(config)# ipv6 unicast-routing
Device(config)# nat64 prefix stateful 2001:db8:2::/96
Device(config)# nat64 v6v4 static 2001:db8:1::5 150.0.0.1 redundancy 1 mapping-id 150
Device(config)# nat64 v6v4 static 2001:db8:1::6 150.0.0.2 redundancy 1 mapping-id 151
Device(config)# nat64 switchover replicate http enable port 80
!
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name RG1
Device(config-red-app-grp)# data gigabitethernet 1/2/0
Device(config-red-app-grp)# control gigabitethernet 1/2/1 protocol 1
Device(config-red-app-grp)# asymmetric-routing interface gigabitethernet 1/2/2
Device(config-red-app-grp)# priority 140 failover threshold 135
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# exit
Device(config-red-app)# exit
Device(config-red)# exit
!
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# ip address 10.10.10.2 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# exit
!
Device(config)# interface gigabitethernet 1/2/1
Device(config-if)# ip address 172.16.0.2 255.240.0.0
Device(config-if)# no shutdown
Device(config-if)# exit
!
Device(config)# interface gigabitethernet 1/2/2
Device(config-if)# ip address 192.168.0.2 255.255.0.0
Device(config-if)#no shutdown
Device(config-if)# exit
!
Device(config-if)# interface gigabitethernet 1/2/3
Device(config-if)# ipv6 enable
Device(config-if)# no shutdown
```

```
Device(config-if)# nat64 enable
Device(config-if)# ipv6 addr 2001:db8:1::3/96
Device(config-if)# redundancy rii 100
Device(config-if)# redundancy group 1 ipv6 2001:db8:1::1/96 exclusive decrement 15
Device(config-if)# exit
!
Device(config)# interface gigabitethernet 1/2/4
Device(config-if)# ip address 198.51.100.1 255.255.255.0
Device(config-if)# nat64 enable
Device(config-if)# no shutdown
Device(config-if)# redundancy rii 101
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# exit
!
Device(config)# router ospf 90
Device(config-router)# network 198.51.100.0 255.255.255.0 area 0
Device(config-router)# end
```

Additional References for Stateful Network Address Translation 64

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
NAT commands	IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title
Framework for IPv4/IPv6 Translation	Framework for IPv4/IPv6 Translation draft-ietf-behave-v6v4-framework-06
FTP ALG for IPv6-to-IPv4 translation	An FTP ALG for IPv6-to-IPv4 translation draft-ietf-behave-ftp64-06
IP/ICMP Translation Algorithm	IP/ICMP Translation Algorithm draft-ietf-behave-v6v4-xlate-10
IPv6 Addressing of IPv4/IPv6 Translators	IPv6 Addressing of IPv4/IPv6 Translators draft-ietf-behave-address-format-07
RFC 2228	FTP Security Extensions
RFC 2373	IP Version 6 Addressing Architecture
RFC 2464	Transmission of IPv6 Packets over Ethernet Networks
RFC 2765	Stateless IP/ICMP Translation Algorithm (SIIT)

Standard/RFC	Title
RFC 2766	Network Address Translation - Protocol Translation (NAT-PT)
RFC 4787	Network Address Translation (NAT) Behavioral Requirements for Unicast UDP
RFC 4966	Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status
RFC 6384	An FTP Application Layer Gateway (ALG) for IPv6-to-IPv4 Translation
Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers	Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers draft-ietf-behave-v6v4-xlate-stateful-12

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Stateful Network Address Translation 64

Table 133: Feature Information for Stateful Network Address Translation 64

Feature Name	Releases	Feature Information
Asymmetric Routing Support for NAT64	Cisco IOS XE Release 3.16S	In Cisco IOS XE Release and later releases, Network Address Translation 64 (NAT64) supports asymmetric routing and asymmetric routing with Multiprotocol Label Switching (MPLS).

Feature Name	Releases	Feature Information
FTP64 NAT ALG Intrabox HA Support	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, the FTP64 ALG adds HA support for Stateful NAT64. The FTP64 NAT ALG Intrabox HA Support feature supports the stateful switchover between redundant FPs within a single chassis. The HA support provided by the FTP64 ALG is applicable to both intrabox and interbox HA and In-Service Software Upgrade (ISSU).
Stateful NAT64 ALG—Stateful FTP64 ALG Support	Cisco IOS XE Release 3.4S	Cisco IOS XE Release 3.4S and later releases support FTP64 (or service FTP) ALGs. The FTP64 ALG helps Stateful NAT64 operate on Layer 7 data. An FTP ALG translates IP addresses and the TCP port information embedded in the payload of an FTP control session. The following commands were introduced or modified: **nat64 service ftp**.
Stateful NAT64—Intra-Chassis Redundancy	Cisco IOS XE Release 3.5S Cisco IOS XE Release 3.10S	Cisco IOS XE Release 3.5S and later releases support the Stateful NAT64—Intra-Chassis Redundancy feature. When a second Forward Processor (FP) is available inside a single chassis, the Stateful NAT64 Intra-Chassis Redundancy feature enables you to configure the second FP as a standby entity. The standby FP maintains a backup of the stateful NAT64 session information and when the active (first) FP fails, there is no disruption of NAT64 sessions. The following commands were introduced or modified: **nat64 switchover replicate http port**.

Feature Name	Releases	Feature Information
Stateful Network Address Translation 64	Cisco IOS XE Release 3.4S	The Stateful Network Address Translation 64 feature provides a translation mechanism that translates IPv6 packets into IPv4 packets and vice versa. The Stateful NAT64 translator, algorithmically translates the IPv4 addresses of IPv4 hosts to and from IPv6 addresses by using the configured stateful prefix. In a similar manner, the IPv6 addresses of IPv6 hosts are translated to and from IPv4 addresses through NAT. The following commands were introduced or modified: **clear nat64 statistics**, **debug nat64**, **nat64 logging**, **nat64 prefix stateful**, **nat64 translation**, **nat64 v4**, **nat64 v4v6**, **nat64 v6v4**, **show nat64 aliases**, **show nat64 limits**, **show nat64 logging**, **show nat64 mappings dynamic**, **show nat64 mappings static**, **show nat64 services**, **show nat64 pools**, **show nat64 prefix stateful**, **show nat64 statistics**, **show nat64 timeouts**, and **show nat64 translations**.

Glossary

ALG—application-layer gateway or application-level gateway.

FP—Forward Processor.

IPv4-converted address—IPv6 addresses used to represent the IPv4 hosts. These have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. Both stateless and stateful translators use IPv4-converted IPv6 addresses to represent the IPv4 hosts.

IPv6-converted address—IPv6 addresses that are assigned to the IPv6 hosts for the stateless translator. These IPv6-converted addresses have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. The stateless translator uses the corresponding IPv4 addresses to represent the IPv6 hosts. The stateful translator does not use IPv6-converted addresses, because the IPv6 hosts are represented by the IPv4 address pool in the translator via dynamic states.

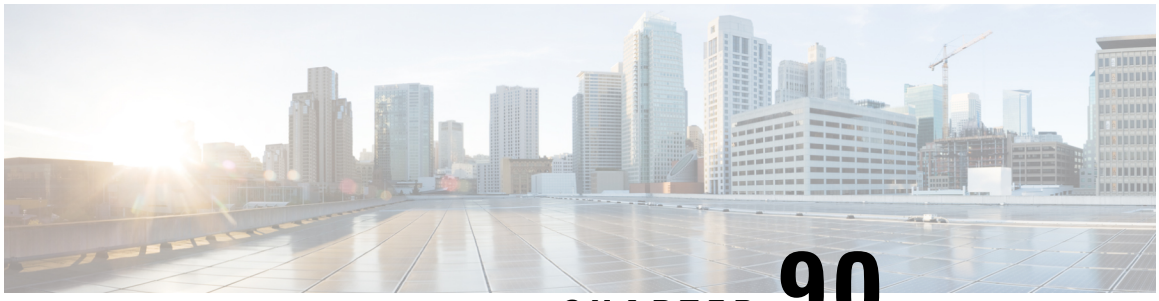
NAT—Network Address Translation.

RP—Route Processor.

stateful translation—In stateful translation a per-flow state is created when the first packet in a flow is received. A translation algorithm is said to be stateful if the transmission or reception of a packet creates or modifies a data structure in the relevant network element. Stateful translation allows the use of multiple translators interchangeably and also some level of scalability. Stateful translation is defined to enable the IPv6 clients and peers without mapped IPv4 addresses to connect to the IPv4-only servers and peers.

stateless translation—A translation algorithm that is not stateful is called stateless. A stateless translation requires configuring a static translation table, or may derive information algorithmically from the messages it is translating. Stateless translation requires less computational overhead than stateful translation. It also

requires less memory to maintain the state, because the translation tables and the associated methods and processes exist in a stateful algorithm and do not exist in a stateless one. Stateless translation enables the IPv4-only clients and peers to initiate connections to the IPv6-only servers or peers that are equipped with IPv4-embedded IPv6 addresses. It also enables scalable coordination of IPv4-only stub networks or ISP IPv6-only networks. Because the source port in an IPv6-to-IPv4 translation may have to be changed to provide adequate flow identification, the source port in the IPv4-to-IPv6 direction need not be changed.



CHAPTER 90

Stateful Network Address Translation 64 Interchassis Redundancy

The Stateful Network Address Translation 64 Interchassis Redundancy feature adds interchassis redundancy support to stateful Network Address Translation 64 (NAT64). The stateful interchassis redundancy enables you to configure pairs of devices to act as backups for each other.

This module describes how to configure stateful NAT64 interchassis redundancy.

Restrictions for Stateful Network Address Translation 64 Interchassis Redundancy

- Asymmetric routing is not supported.
- Box-to-box (B2B) redundancy along with intrachassis redundancy is not supported.
- NAT interface overload configuration is not supported.

Information About Stateful Network Address Translation 64 Interchassis Redundancy

Stateful Interchassis Redundancy Operation

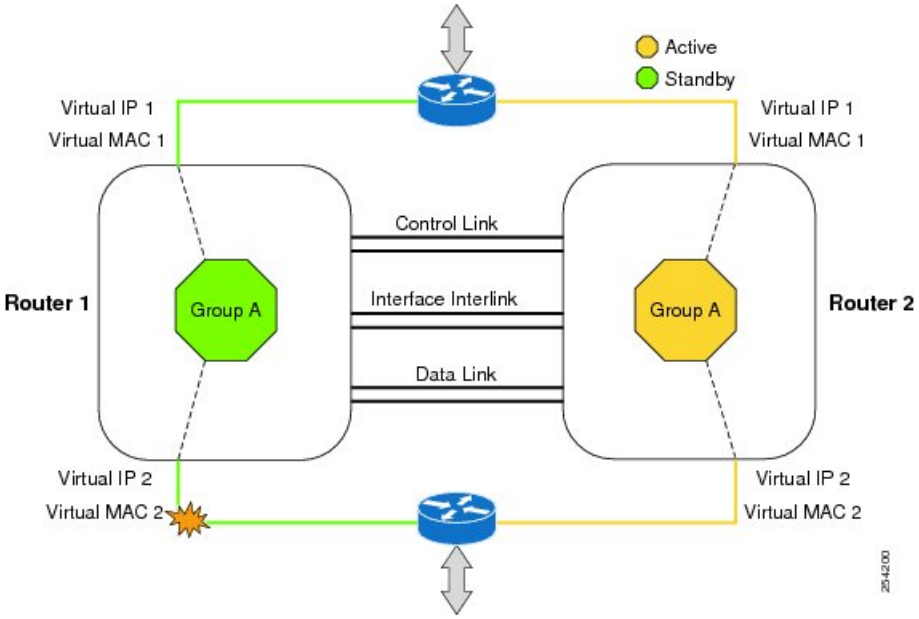
You can configure pairs of devices to act as hot standbys for each other. Redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups (RGs). Redundancy occurs at an application level and does not require a complete physical failure of the interface or device for a switchover

of the application to occur. When a switchover occurs, the application activity continues to run seamlessly on the redundant interface.

The figure below depicts an active/standby load-sharing scenario. The figure shows how an RG is configured for a pair of devices that has one outgoing interface. Group A on Router 1 is the active RG and Group A on Router 2 is the standby RG.

Redundant devices are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of devices. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and synchronize the stateful database. The pairs of redundant interfaces are configured with the same unique ID number known as the redundant interface identifier (RII).

Figure 94: Redundancy Group Configuration—One Outgoing Interface



The status of redundancy group members is determined through the use of hello messages sent over the control link. The software considers either device not responding to a hello message within a configurable amount of time to be a failure and initiates a switchover. For the software to detect a failure in milliseconds, control links run the failover protocol that is integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for hello messages:

- Hello time—Interval at which hello messages are sent.
- Hold time—Amount of time before which the active or standby device is declared to be down.

The hello time defaults to 3 seconds to align with the Hot Standby Router Protocol (HSRP), and the hold time defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine the pairs of interfaces that are affected by the switchover, you must configure a unique ID for each pair of redundant interfaces. This ID is known as the RII that is associated with the interface.

A switchover to the standby device can occur when the priority setting that is configured on each device changes. The device with the highest priority value acts as the active device. If a fault occurs on either the active or standby device, the priority of the device is decremented by a configurable amount known as the weight. If the priority of the active device falls below the priority of the standby device, a switchover occurs

and the standby device becomes the active device. This default behavior can be overridden by disabling the preemption attribute for the RG. You can also configure each interface to decrease the priority when the Layer 1 state of the interface goes down. The priority that is configured overrides the default priority of an RG.

Each failure event that causes a modification of an RG priority generates a syslog entry that contains a time stamp, the RG that was affected, the previous priority, the new priority, and a description of the failure event cause.

A switchover also can occur when the priority of a device or interface falls below a configurable threshold level.

A switchover to the standby device occurs under the following circumstances:

- Power loss or a reload occurs on the active device (including reloads).
- The run-time priority of the active device goes below that of the standby device (with preempt configured).
- The run-time priority of the active device goes below that of the configured threshold.
- The redundancy group on the active device is reloaded manually. Use the **redundancy application reload group** *rg-number* command for a manual reload.

Active/Active Failover

In an active/active failover configuration, both devices can process network traffic. Active/active failover generates virtual MAC (VMAC) addresses for interfaces in each redundancy group (RG).

One device in an active/active failover pair is designated as the primary (active) device, and the other is designated as the secondary (standby) device. Unlike with active/standby failover, this designation does not indicate which device becomes active when both devices start simultaneously. Instead, the primary/secondary designation determines the following:

- The device that provides the running configuration to the failover pair when they start simultaneously.
- The device on which the failover RG appears in the active state when devices start simultaneously. Each failover RG in the configuration is configured with a primary or secondary device preference. You can configure both failover RGs to be in the active state on a single device and the standby failover RGs to be on the other device. You can also configure one failover RG to be in the active state and the other RG to be in the standby state on a single device.

Active/Standby Failover

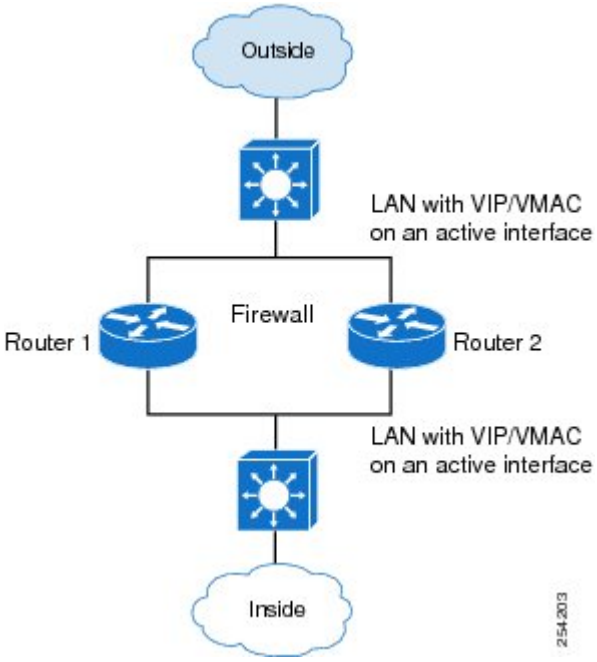
Active/standby failover enables you to use a standby device to take over the functionality of a failed device. A failed active device changes to the standby state, and the standby device changes to the active state. The device that is now in the active state takes over IP addresses and MAC addresses of the failed device and starts processing traffic. The device that is now in the standby state takes over standby IP addresses and MAC addresses. Because network devices do not see any change in the MAC-to-IP address pairing, Address Resolution Protocol (ARP) entries do not change or time out anywhere on the network.

In an active/standby scenario, the main difference between two devices in a failover pair depends on which device is active and which device is a standby, namely which IP addresses to use and which device actively passes the traffic. The active device always becomes the active device if both devices start up at the same time (and are of equal operational health). MAC addresses of the active device are always paired with active IP addresses.

LAN-LAN Topology

In a LAN-LAN topology, all participating devices are connected to each other through LAN interfaces on both the inside and the outside. In this scenario, the traffic is often directed to the correct firewall if static routing is configured on the upstream or downstream devices to an appropriate virtual IP address. The dynamic routing configuration supported on LAN-facing interfaces must not introduce a dependency on routing protocol convergence; otherwise, fast failover requirements will not be met. The figure below shows a LAN-LAN topology.

Figure 95: LAN-LAN Scenario



Redundancy Groups for Stateful NAT64

To support stateful Network Address Translation 64 (NAT64) box-to-box (B2B) redundancy, all stateful NAT64 mappings must be associated with a redundancy group (RG). You can associate multiple stateful NAT64 mappings with one RG. Any session or bind that is created from a stateful NAT64 mapping is associated with the RG to which the stateful NAT64 is mapped. In B2B redundancy, stateful NAT64 checks the state of the created, changed, or destroyed session or bind in the RG to determine whether the stateful NAT64 high availability (HA) message should be sent to the standby device.

NAT binding is a one-to-one association between a local IP address and a global IP address. Sessions are identified by the 5-tuple (the source IP address, the destination IP address, the protocol, the source port, and the destination port) information. Sessions are normally created and destroyed at a much faster rate than bindings.

Translation Filtering

RFC 4787 provides translation filtering behaviors for Network Address Translation (NAT). The following options are used by NAT to filter packets that originate from specific external endpoints:

- Endpoint-independent filtering—Filters out packets that are not destined to an internal IP address and port regardless of the external IP address and port source.
- Address-dependent filtering—Filters out packets that are not destined to an internal IP address. NAT also filters out packets that are destined for an internal endpoint.
- Address- and port-dependent filtering—Filters out packets that are not destined to an internal IP address. NAT also filters out packets that are destined for an internal endpoint if packets were not sent to the endpoint previously.

FTP64 Application-Level Gateway Support

The FTP64 (or service FTP) application-level gateway (ALG) helps stateful Network Address Translation 64 (NAT64) to operate on Layer 7 data. FTP64 ALG translates IP addresses and the TCP port information embedded in the payload of an FTP control session.

NAT translates any TCP/UDP traffic that does not carry source and destination IP addresses in the application data stream. Protocols that embed the IP address information within the payload (or in the application data stream) require the support of an ALG. ALGs handle application data stream (Layer 7) protocol-specific services, such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection or session information from control channels.

FTP64 is automatically enabled when Stateful NAT64 is enabled. Use the **no nat64 service ftp** command to disable the NAT64 FTP service.

Note The FTP64 ALG is not supported in Stateless NAT64 translation.

Note The FTP64 ALG does not support IPv4-compatible IPv6 addresses.

Based on *IPv6-to-IPv4 translation FTP considerations draft-ietf-behave-ftp64-02* and RFC 2228, the FTP64 ALG must switch to transparent mode (a device in a transparent mode is invisible in the network; however, this device can act as a bridge and inspect or filter packets), when commands and responses flow between the FTP client and the FTP server. When a client issues the FTP AUTH command, the FTP64 ALG transparently forwards all data on the control channel in both (ingress and egress) directions, until the end of the control channel session. Similarly, during an AUTH negotiation, the ALG must be in transparent mode, whether the negotiation is successful or not.

Based on RFC 6384, the behavior of the FTP64 ALG during a client-server communication is different. During an IPv6-to-IPv4 translation, the FTP64 ALG must transparently copy data transmitted over the control channel so that the transport layer security (TLS) session works correctly. However, the client commands and server responses are hidden from the FTP64 ALG. To ensure a consistent behavior, as soon as the initial FTP AUTH command is issued by a client, the FTP64 ALG must stop translating commands and responses and start transparently copying TCP data that is sent by the server to the client and vice versa. The FTP64 ALG must ignore the AUTH command and not go into transparent mode if the server response is in the 4*xx* or 5*xx* ranges, which comprise FTP error/warning messages.

Prior to CSCtu37975, when an IPv6 FTP client issues an FTP AUTH command, irrespective of whether the IPv4 FTP server accepts or rejects that authorization negotiation, the FTP64 ALG moves the AUTH session to transparent mode (or bypass mode). When a session is in transparent mode, NAT cannot perform translation

on the packets within the session. With CSCtu37975, during a client-server communication, the FTP64 ALG's behavior is compliant with RFC 6384.

How to Configure Stateful Network Translation 64 Interchassis Redundancy

Configuring Redundancy Group Protocols

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **protocol** *id*
6. **name** *group-name*
7. Repeat Steps 3 to 6 to configure a redundancy group protocol on another device.
8. **timers hellotime** *seconds* **holdtime** *seconds*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Defines a protocol instance for a redundancy group and enters redundancy application protocol configuration mode.

	Command or Action	Purpose
Step 6	**name** *group-name* **Example:** `Device(config-red-app-prtcl)# name RG1`	Configures a name for the redundancy group.
Step 7	Repeat Steps 3 to 6 to configure a redundancy group protocol on another device.	—
Step 8	**timers hellotime** *seconds* **holdtime** *seconds* **Example:** `Device(config-red-app-prtcl)# timers hellotime 1 holdtime 3`	Configures timers for hellotime and holdtime messages for a redundancy group.
Step 9	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring Redundancy Groups for Active/Standby Load Sharing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **control** *interface-type interface-number* **protocol** *id*
8. **data** *interface-type interface-number*
9. Repeat Steps 3 to 8 to configure another redundancy group.
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:**	Enters redundancy configuration mode.

	Command or Action	Purpose
	`Device(config)# redundancy`	
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy application group and enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name RG1`	Configures a name for the redundancy application group.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1`	Configures a control interface type and number for the redundancy application group.
Step 8	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data gigabitethernet 0/2/2`	Configures a data interface type and number for the redundancy application group.
Step 9	Repeat Steps 3 to 8 to configure another redundancy group.	—
Step 10	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring Redundancy Groups for Active/Active Load Sharing

Perform this task to configure two redundancy groups (RGs) on the same device for active/active load sharing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **priority** *value* [**failover-threshold** *value*]
8. **control** *interface-type interface-number* **protocol** *id*
9. **data** *interface-type interface-number*
10. **end**

11. **configure terminal**
12. **redundancy**
13. **application redundancy**
14. **group** *id*
15. **name** *group-name*
16. **priority** *value* [**failover-threshold** *value*]
17. **control** *interface-type interface-number* **protocol** *id*
18. **data** *interface-type interface-number*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy application group and enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name RG1`	Configures a name for the redundancy application group.
Step 7	**priority** *value* [**failover-threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 195 failover-threshold 190`	Specifies a group priority and failover threshold value for the redundancy group.
Step 8	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1`	Configures a control interface type and number for the redundancy application group.

	Command or Action	Purpose
Step 9	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data gigabitethernet 0/2/2`	Configures a data interface type and number for the redundancy application group.
Step 10	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.
Step 11	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 12	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 13	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 14	**group** *id* **Example:** `Device(config-red-app)# group 2`	Configures a redundancy application group and enters redundancy application group configuration mode.
Step 15	**name** *group-name* **Example:** `Device(config-red-app-grp)# name RG2`	Configures a name for the redundancy application group.
Step 16	**priority** *value* [**failover-threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 205 failover-threshold 200`	Specifies a group priority and failover threshold value for the redundancy group.
Step 17	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 2`	Configures a control interface type and number for the redundancy application group.
Step 18	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data gigabitethernet 0/2/2`	Configures a data interface type and number for the redundancy application group.

	Command or Action	Purpose
Step 19	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Traffic Interface for Stateful NAT64 Interchassis Redundancy

This task applies to a LAN-LAN scenario.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **redundancy rii** *id*
5. **redundancy group** *group-id* **ipv6** *ipv6-prefix/prefix-length* **exclusive decrement** *value*
6. **exit**
7. **interface** *type number*
8. **redundancy rii** *id*
9. **redundancy group** *group-id* **ipv6** *ipv6-prefix/prefix-length* **exclusive decrement** *value*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 100`	Configures a redundancy interface identifier (RII) for a redundancy group-protected traffic interfaces.
Step 5	**redundancy group** *group-id* **ipv6** *ipv6-prefix/prefix-length* **exclusive decrement** *value* **Example:**	Enables IPv6 redundancy.

	Command or Action	Purpose
	`Device(config-if)# redundancy group 1 ipv6 2001:DB8:1::1:100/64 exclusive decrement 50`	
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 8	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 120`	Configures an RII for a redundancy group-protected traffic interfaces.
Step 9	**redundancy group** *group-id* **ipv6** *ipv6-prefix/prefix-length* **exclusive decrement** *value* **Example:** `Device(config-if)# redundancy group 1 ipv6 2001:DB8:2::1:100/64 exclusive decrement 50`	Enables IPv6 redundancy.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring Static Stateful NAT64 for Interchassis Redundancy

Perform this task to configure a static stateful NAT64 with interchassis redundancy. You can configure interchassis redundancy with the following types of NAT configurations: dynamic, static, and Port Address Translation (PAT) translations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 enable**
6. **ipv6 address** *ipv6-address/prefix-length*
7. **nat64 enable**
8. **exit**
9. Repeat Steps 3 to 8 to configure NAT64 on another interface.
10. **nat64 prefix stateful** *ipv6-prefix/length*
11. **nat64 v6v4 static** *ipv6-address ipv6-address* [**redundancy** *group-id* **mapping-id** *id*]
12. **nat64 v6v4 tcp** *ipv6-address ipv6-port ipv4-address ipv4-port* [**redundancy** *group-id* **mapping-id** *id*]

13. **end**
14. **show nat64 translations protocol tcp**
15. **show nat64 translations redundancy** *group-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 5	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 6	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 7	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 translation on an IPv6 interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	Repeat Steps 3 to 8 to configure NAT64 on another interface.	—

	Command or Action	Purpose
Step 10	**nat64 prefix stateful** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateful 2001:DB8:1::1/96`	Defines the stateful NAT64 prefix that is to be added to IPv4 hosts to translate the IPv4 address into an IPv6 address. • The stateful NAT64 prefix can be configured at the global configuration level or at the interface configuration level.
Step 11	**nat64 v6v4 static** *ipv6-address ipv6-address* [**redundancy** *group-id* **mapping-id** *id*] **Example:** `Device(config)# nat64 v6v4 static 2001:DB8:1::FFFE 209.165.201.1 redundancy 1 mapping-id 30`	Enables NAT64 IPv6-to-IPv4 static address mapping and interchassis redundancy.
Step 12	**nat64 v6v4 tcp** *ipv6-address ipv6-port ipv4-address ipv4-port* [**redundancy** *group-id* **mapping-id** *id*] **Example:** `Device(config)# nat64 v6v4 tcp 2001:DB8:1::1 redundancy 1 mapping-id 1`	Applies static mapping to TCP protocol packets and enables interchassis redundancy.
Step 13	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.
Step 14	**show nat64 translations protocol tcp** **Example:** `Device# show nat64 translations protocol tcp`	Displays information about NAT 64 protocol translations.
Step 15	**show nat64 translations redundancy** *group-id* **Example:** `Device# show nat64 translations redundancy 1`	Displays information about NAT64 redundancy translations.

Example:

The following is sample output from the **show nat64 translations protocol tcp** command:

```
Device# show nat64 translations protocol tcp

 Proto  Original IPv4         Translated IPv4
        Translated IPv6       Original IPv6
 ------------------------------------------------------------------------------

 tcp    209.165.201.2:21      [2001:DB8:1::103]:32847
        10.2.1.1:80           [2001::11]:80
 tcp    209.165.201.2:21      [2001:DB8:1::104]:32848
        10.2.1.1:80           [2001::11]:80

 Total number of translations: 2
```

The following is sample output from the **show nat64 translations redundancy** command:

```
Device# show nat64 translations redundancy 1

 Proto  Original IPv4         Translated IPv4
        Translated IPv6       Original IPv6
 ------------------------------------------------------------------------------

        209.165.201.2:21      [2001:DB8:1::103]:32847

 tcp    10.2.1.11:32863       [2001::3201:10b]:32863
        10.1.1.1:80           [2001::11]:80
 tcp    209.165.201.2:21      [2001:DB8:1::104]:32848
        10.1.1.1:80           [2001::11]:80

 Total number of translations: 3
```

Configuration Examples for Stateful Network Address Translation 64 Interchassis Redundancy

Example: Configuring Redundancy Group Protocols

```
Device# configure terminal
Device(config)# redundancy
Device(red)# application redundancy
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# name RG1
Device(config-red-app-prtcl)# timers hellotime 1 holdtime 3
Device(config-red-app-prtcl)# end
Device# configure terminal
Device(config)# redundancy
Device(red)# application redundancy
Device(config-red-app)# protocol 2
Device(config-red-app-prtcl)# name RG1
Device(config-red-app-prtcl)# end
```

Example: Configuring Redundancy Groups for Active/Standby Load Sharing

The following example shows how to configure redundancy groups (RGs) on two devices for active/standby load sharing:

```
Device1# configure terminal
Device1(config)# redundancy
Device1(config-red)# application redundancy
Device1(config-red-app)# group 1
Device1(config-red-app-grp)# name RG1
Device1(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1
Device1(config-red-app-grp)# data gigabitethernet 0/2/2
Device1(config-red-app-grp)# end

Device2# configure terminal
Device2(config)# redundancy
Device2(config-red)# application redundancy
Device2(config-red-app)# group 1
Device2(config-red-app-grp)# name RG1
```

```
Device2(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1
Device2(config-red-app-grp)# data gigabitethernet 0/2/2
Device2(config-red-app-grp)# end
```

Example: Configuring Redundancy Groups for Active/Active Load Sharing

The following example shows how to configure two redundancy groups (RGs) on the same device for active/active load sharing:

```
Device1# configure terminal
Device1(config)# redundancy
Device1(config-red)# application redundancy
Device1(config-red-app)# group 1
Device1(config-red-app-grp)# name RG1
Device1(config-red-app-grp)# priority 195 failover-threshold 190
Device1(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1
Device1(config-red-app-grp)# data gigabitethernet 0/2/2
Device1(config-red-app-grp)# end
Device1# configure terminal
Device1(config)# redundancy
Device1(config-red)# application redundancy
Device1(config-red-app)# group 2
Device1(config-red-app-grp)# name RG2
Device1(config-red-app-grp)# priority 205 failover-threshold 200
Device1(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 2
Device1(config-red-app-grp)# data gigabitethernet 0/2/2
Device1(config-red-app-grp)# end

Device2# configure terminal
Device2(config)# redundancy
Device2(config-red)# application redundancy
Device2(config-red-app)# group 1
Device2(config-red-app-grp)# name RG1
Device2(config-red-app-grp)# priority 195 failover-threshold 190
Device2(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 1
Device2(config-red-app-grp)# data gigabitethernet 0/2/2
Device2(config-red-app-grp)# end
Device2# configure terminal
Device2(config)# redundancy
Device2(config-red)# application redundancy
Device2(config-red-app)# group 2
Device2(config-red-app-grp)# name RG2
Device2(config-red-app-grp)# priority 205 failover-threshold 200
Device2(config-red-app-grp)# control gigabitethernet 0/0/1 protocol 2
Device2(config-red-app-grp)# data gigabitethernet 0/2/2
Device2(config-red-app-grp)# end
```

Example: Configuring a Traffic Interface for Stateful NAT64 Interchassis Redundancy

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# redundancy rii 100
Device(config-if)# redundancy group 1 ipv6 2001:DB8:1::1:100/64 exclusive decrement 50
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
```

```
Device(config-if)# redundancy rii 100
Device(config-if)# redundancy group 1 ipv6 2001:DB8::2:1:100/64 exclusive decrement 50
Device(config-if)# end
```

Additional References

Related Documents

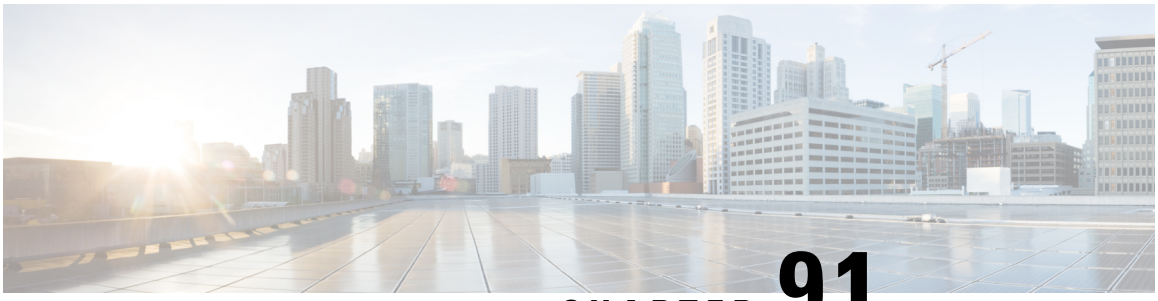
Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
NAT commands	IP Addressing Services Command Reference

Standards/RFCs

Standard/RFC	Title
RFC 4787	*Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 91

Connectivity Between IPv4 and IPv6 Hosts Using Stateless NAT 46

The Network Address Translation 46 (NAT 46) feature solves IPv4 to IPv6 connectivity by providing a mechanism for connectivity of IPv4 hosts to IPv6 internet when dual stack and IPv6 tunneling solutions cannot be used.

Feature Information for Connectivity Between IPv4 and IPv6 Hosts Using Stateless NAT 46

Table 134: Feature Information for Connectivity Between IPv4 and IPv6 Hosts Using Stateless NAT 46

Feature Name	Releases	Feature Information
Connectivity Between IPv4 and IPv6 Hosts Using Stateless NAT 46	Cisco IOS XE Gibraltar 16.10.1 Release	The Network Address Translation 46 (NAT 46) feature solves IPv4 to IPv6 connectivity by providing a mechanism for connectivity of IPv4 hosts to IPv6 internet when dual stack and IPv6 tunneling solutions cannot be used. **Note** NAT 46 is supported only on Cisco ISR 4000 platforms.

Restrictions for NAT 46

- Only Domain Name System (DNS) application layer gateway (ALG) is supported.

- Fragmented packet is not supported.
- Maximum Transmission Unit (MTU) discovery after converting to IPv6 packets is not supported.
- Virtual Routing and Forwarding-aware NAT 46 is not supported.
- Both NAT44 (static, dynamic, and PAT) configuration and stateful NAT46 configurations are not supported on the same interface.
- High-speed Logging (HSL) is not supported.
- Several IPv4 stateful features (PBR, ZBFW, WAAS, WCCP, NBAR, and so on) do not work after converting to IPv6 packets, and are not supported.
- High availability is not supported.

Information About NAT 46

Overview of NAT 46

The NAT46 solution solves IPv4 host to IPv6 internet connectivity. IPv4 hosts trying to reach a server, first initiate a DNS type A query packet. The NAT 46 router changes this to type AAAA query. When the query response is received, NAT 46 retrieves the IPv6 address from the response packet. An IPv4 address is allocated from the configured NAT 46 pool and an address binding is done for the retrieved IPv6 address and the allocated IPv4 address. An IPv4 address DNS response is sent to the IPv4 host. The source address of packets originating from IPv4 hosts is converted using a configured NAT 46 IPv6 prefix. The destination IPv4 address is translated to IPv6 address using pool address binding created during DNS packet flow.

Example:

Configured Prefix	IPv4 Address	IPv4-Embedded IPv6 Address
`2002:0DB8::/96`	`192.0.2.33`	`2002:0DB8::C000221`

Scalability on NAT 46

There is no limitation to the number of private IPv4 addresses that can be supported because no sessions are maintained. The number of IPv6 hosts that can be represented by the IPv4 pool address should be scalable up to 40,000.

NAT 46 Prefix

The NAT 46 prefix cannot be same as the interface prefix. Neighbor Discovery Neighbor/Router Solicitation messages for the addresses in the NAT 46 prefix are not answered by the NAT 46 router. Hence, NAT 46 prefix cannot be same as the interface prefix.

If a larger network (smaller prefix that is less than 96) is obtained from the service provider, the network can be subdivided into multiple smaller networks and NAT 46 prefix can be configured with a smaller network (prefix 96 bits). In addition, the NAT 46 router needs to be configured as a gateway or next hop router for the IPv6 hosts on an adjacent router of the service provider network.

Configuring Network Address Translation 46

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet 1/2/0
```

Configures an interface and enters interface configuration mode.

Step 4 **ip address** *ip-address mask*

Example:

```
Device(config-if)# ip address 209.165.201.1 255.255.255.0
```

Configures an IPv4 address for an interface.

Step 5 **nat64 enable**

Example:

```
Device(config-if)# nat64 enable
```

Enables NAT46 translation on an IPv4 interface.

Step 6 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode and enters global configuration mode.

Step 7 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet 0/0/0
```

Configures an interface and enters interface configuration mode.

Step 8 **ipv6 enable**

Example:

```
Device(config-if)# ipv6 enable
```

Enables IPv6 processing on an interface.

Step 9 **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/ prefix-lenth*

Example:

```
Device(config-if)# ipv6 address 2001:DB8:1::1/96
```

Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Step 10 **nat64 enable**

Example:

```
Device(config-if)# nat64 enable
```

Enables NAT46 translation on an IPv6 interface.

Step 11 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode and enters global configuration mode.

Step 12 **nat64 settings nat46 enable**

Example:

```
Device(config)# nat64 settings nat46 enable
```

Enables NAT46 in the NAT64 settings.

Step 13 **nat46 v6 prefix** *ipv6 prefix/prefix-length*

Example:

```
Device(config)# nat46 v6 prefix 2001::/96
```

Configures the NAT46 IPv6 prefix.

Step 14 **nat46 v4 pool** *pool-name pool-address-range*

Example:

```
Device(config)# nat46 v4 nat46_pool 13.0.0.1 13.0.0.200
```

Configures the NAT46 pool address range.

Step 15 **end**

Example:

```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Verifying the NAT 46 Configuration

Use the **show nat64 statistics** command to view the NAT 46 statistics. The following is sample output of the command.

SUMMARY STEPS

1. **show nat64 statistics**

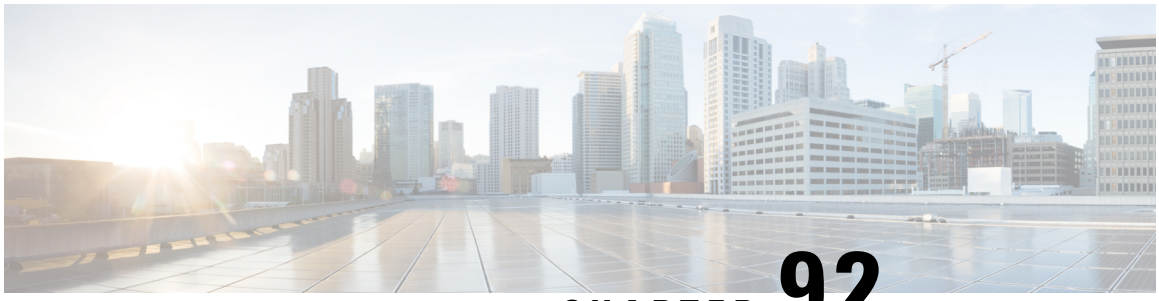
DETAILED STEPS

show nat64 statistics

Example:

```
Router# show nat64 statistics

NAT64 Statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Sessions found: 0
Sessions created: 0
Expired translations: 0
Global Stats:
   Packets translated (IPv4 -> IPv6)
   Stateless: 0
   Stateful: 0
   MAP-T: 0
   NAT46: 30
   Packets translated (IPv6 -> IPv4)
   Stateless: 0
   Stateful: 0
   MAP-T: 0
   NAT46: 30
```



CHAPTER 92

Mapping of Address and Port Using Translation

The Mapping of Address and Port Using Translation feature provides connectivity to IPv4 hosts across IPv6 domains. Mapping of address and port using translation (MAP-T) is a mechanism that performs double translation (IPv4 to IPv6 and vice versa) on customer edge (CE) devices and border routers.

This module provides an overview of MAP-T and explains how to configure this feature.

Restrictions for Mapping of Address and Port Using Translation

- The mapping of address and port using translation (MAP-T) customer edge (CE) functionality is not supported.
- In Cisco IOS XE Denali 16.2 release, the support for MAP-T domains were extended to 10000 domains. For releases prior to Cisco IOS XE Denali 16.2, a maximum of 128 MAP-T domains are supported.
- Forwarding mapping rule (FMR) is not supported.

Information About Mapping of Address and Port Using Translation

Mapping of Address and Port Using Translation Overview

The Mapping of Address and Port Using Translation feature provides connectivity to IPv4 hosts across IPv6 domains. Mapping of address and port using translation (MAP-T) builds on the existing stateless IPv4 and IPv6 address translation techniques that are specified in RFCs 6052, 6144, and 6145.

MAP-T is a mechanism that performs double translation (IPv4 to IPv6 and vice versa) on customer edge (CE) devices and border routers. The Mapping of Address and Port Using Translation feature supports only the MAP-T border router functionality. This feature does not support the MAP-T CE functionality.

The Mapping of Address and Port Using Translation feature leverages the Network Address Translation 64 (NAT64) translation engine and adds the MAP-T border router function to the NAT64 stateless function. MAP-T is enabled on IPv4 and IPv6 interfaces. MAP-T uses IPv4 and IPv6 forwarding, IPv4 and IPv6 fragmentation functions, and NAT64 translation functions. A MAP-T domain is one or more MAP CE devices and a border router, all connected to the same IPv6 network.

A MAP-T CE device connects a user's private IPv4 address and the native IPv6 network to the IPv6-only MAP-T domain. The MAP-T border router uses the stateless IPv4/IPv6 translation to connect external IPv4 networks to all devices available in the one or more MAP-T domains. MAP-T requires only one IPv6 prefix per network and supports the regular IPv6 prefix/address assignment mechanisms. The MAP-T domain contains regular IPv6-only hosts or servers that have an IPv4-translatable IPv6 address. MAP-T does not require the operation of an IPv4 overlay network or the introduction of a non-native-IPv6 network device or server functionality.

A MAP-T configuration provides the following features:

- Retains the ability for IPv4 end hosts to communicate across the IPv6 domain with other IPv4 hosts.
- Permits both individual IPv4 address assignment and IPv4 address sharing with a predefined port range.
- Allows communication between IPv4-only and IPv6-enabled end hosts and native IPv6-only servers in domains that use IPv4-translatable IPv6 addresses.
- Allows the use of IPv6 native network operations, including the ability to classify IP traffic and perform IP traffic routing optimization policies such as routing optimization based on peering policies for IPv4 destinations outside the domain.

MAP-T Mapping Rules

Mapping rules define the mapping between an IPv4 prefix and an IPv4 address or between a shared IPv4 address and an IPv6 prefix/address. Each mapping of address and port using translation (MAP-T) domain uses a different mapping rule.

A MAP-T configuration has one basic mapping rule (BMR), one default mapping rule (DMR), and one or more forwarding mapping rules (FMRs) for each MAP-T domain. You must configure the DMR before configuring the BMR for a MAP-T domain.

The three types of mapping rules are described below:

- A BMR configures the MAP IPv6 address or prefix. The basic mapping rule is configured for the source address prefix. You can configure only one basic mapping rule per IPv6 prefix. The basic mapping rule is used by the MAP-T CE to configure itself with an IPv4 address, an IPv4 prefix, or a shared IPv4 address from an IPv6 prefix. The basic mapping rule can also be used for forwarding packets, where an IPv4 destination address and a destination port are mapped into an IPv6 address/prefix. Every MAP-T node (a CE device is a MAP-T node) must be provisioned with a basic mapping rule. You can use the **port-parameters** command to configure port parameters for the MAP-T BMR.
- A DMR is a mandatory rule that is used for mapping IPv4 information to IPv6 addresses for destinations outside a MAP-T domain. A 0.0.0.0/0 entry is automatically configured in the MAP rule table (MRT) for this rule.

- An FMR is used for forwarding packets. Each FMR results in an entry in the MRT for the rule IPv4 prefix. FMR is an optional rule for mapping IPv4 and IPv6 destinations within a MAP-T domain.

Note FMR is not supported by the Mapping of Address and Port Using Translation feature.

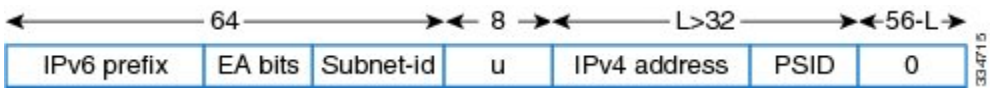
MAP-T Address Formats

The mapping of address and port using translation (MAP-T) customer edge (CE) device address format is defined by the IETF draft Mapping of Address and Port (MAP) . Address formats are used during mapping rule operations to construct the source and destination IPv6 addresses.

Note Forwarding mapping rule (FMR) is not supported by the Mapping of Address and Port Using Translation feature.

The figure below shows the mapped CE address format as defined in MAP-T configuration. This address format is used in basic mapping rule (BMR) and FMR operations.

Figure 96: IPv4-Translatable Address for BMR and FMR



The figure below shows the address format used by the MAP-T default mapping rule (DMR), an IPv4-translated address that is specific to MAP-T configuration.

Figure 97: IPv4-Translated Address for DMR

Packet Forwarding in MAP-T Customer Edge Devices

Note The Mapping of Address and Port Using Translation feature does not support the MAP-T customer edge (CE) functionality. The CE functionality is provided by third-party devices.

IPv4-to-IPv6 Packet Forwarding

A mapping of address and port using translation (MAP-T) CE device that receives IPv4 packets performs Network Address Translation (NAT) and creates appropriate NAT stateful bindings. The resulting IPv4 packets contain the source IPv4 address and the source transport number defined by MAP-T. This IPv4 packet is forwarded to the CE's MAP-T, which performs IPv4-to-IPv6 stateless translation. IPv6 source and destination addresses are then derived by the MAP-T translation, and IPv4 headers are replaced with IPv6 headers.

IPv6-to-IPv4 Packet Forwarding

A MAP-T CE device that receives an IPv6 packet performs its regular IPv6 operations. Only the packets that are addressed to the basic mapping rule (BMR) address are sent to the CE's MAP-T. All other IPv6 traffic is forwarded based on the IPv6 routing rules on the CE device. The CE device checks if the transport-layer destination port number of the packets received from MAP-T is in the range that was configured and forwards packets that confirm to the port number. The CE device drops all nonconforming packets and responds with an Internet Control Message Protocol Version 6 (ICMPv6) "Address Unreachable" message.

Packet Forwarding in Border Routers

IPv4-to-IPv6 Packet Forwarding

An incoming IPv4 packet is processed by the IPv4 input interface, and the destination route lookup routes the IPv4 packet to the mapping of address and port using translation (MAP-T) virtual interface. The border router compares the packet against the IPv4 prefix lookup unit (PLU) tree to obtain the corresponding basic mapping rule (BMR), the default mapping rule (DMR), and the forwarding mapping rule (FMR). Based on the BMR or FMR rules, the border router constructs the IPv6 destination address by encoding the embedded address (EA) bits and adding a suffix. The IPv6 source address is constructed from the DMR rule.

After the IPv6 source and destination addresses are constructed, the packet uses the Network Address Translation 64 (NAT64) IPv4-to-IPv6 translation to construct the IPv6 packet. A routing lookup is done on the IPv6 packet, and the packet is forwarded to the IPv6 egress interface for processing and transmission.

IPv6-to-IPv4 Packet Forwarding

An incoming IPv6 packet is processed by the IPv6 input interface, and the destination route lookup routes the IPv6 packet to the MAP-T virtual interface. The software compares the packet against the IPv6 PLU tree to obtain the corresponding BMR, DMR, and FMR rules. The border router checks whether the port-set ID (PSID) and the port set match. If the port-set ID and port set match, the DMR rule matches the packet destination of the IPv6 packet. Based on the BMR and FMR, the border router constructs the IPv4 source address and extracts the IPv4 destination address from the IPv6 destination address. The IPv6 packet uses the NAT64 IPv6-to-IPv4 translation engine to construct the IPv4 packet from the IPv6 packet. A routing lookup is done on the IPv4 packet, and the IPv4 packet is forwarded to the IPv4 egress interface for processing and transmission.

ICMP/ICMPv6 Header Translation for MAP-T

Mapping of address and port using translation (MAP-T) customer edge (CE) devices and border routers use the ICMP/ICMPv6 translation for address sharing of port ranges.

Unlike TCP and UDP, which provide two port fields to represent source and destination addresses, the Internet Control Message Protocol (ICMP) and ICMP Version 6 (ICMPv6) query message headers have only one ID field.

When an ICMP query message originates from an IPv4 host that exists beyond a MAP-T CE device, the ICMP ID field is exclusively used to identify the IPv4 host. The MAP-T CE device rewrites the ID field to a port-set value that is obtained through the basic mapping rule (BMR) during the IPv4-to-IPv6 translation, and the border router translates ICMPv6 packets to ICMP.

When a MAP-T border router receives an ICMP packet that contains an ID field that is bound for a shared address in the MAP-T domain, the MAP-T border router uses the ID field as a substitute for the destination port to determine the IPv6 destination address. The border router derives the destination IPv6 address by

mapping the destination IPv4 address without the port information for packets that do not contain the ID field, and the corresponding CE device translates the ICMPv6 packets to ICMP.

Path MTU Discovery and Fragmentation in MAP-T

Mapping of address and port using translation (MAP-T) uses path maximum transmission unit (MTU) discovery and fragmentation for IPv4-to-IPv6 translation because the size of IPv4 (more than 20 octets) and IPv6 (40 octets) headers is different. The MTU defines the largest size of a packet that an interface can transmit without the need to fragment the packet. IP packets larger than the MTU must go through IP fragmentation procedures.

When an IPv4 node performs path MTU discovery by setting the Don't Fragment (DF) bit in the packet header, path MTU discovery operates end-to-end across the MAP-T border router and customer edge (CE) translators. During IPv4 path MTU discovery, either the IPv4 device or the IPv6 device can send ICMP “Packet Too Big” messages to the sender. When IPv6 devices send these messages as Internet Control Message Protocol Version 6 (ICMPv6) errors, the packets that follow the message pass through the translator and result in an appropriate ICMP error message sent to the IPv4 sender.

When the IPv4 sender does not set the DF bit, the translator fragments the IPv4 packet and includes the packet with fragment headers to fit the packet in the minimum MTU 1280-byte IPv6 packets. When packets are fragmented, either by the sender or by IPv4 devices, the low-order 16 bits of the fragment identification are carried end-to-end across the MAP-T domain to ensure that packets are reassembled correctly.

How to Configure Mapping of Address and Port Using Translation

Prerequisites for Configuring MAP-T

There are no prerequisites to configure MAP-T.

Restrictions for Configuring MAP-T

- Application-level Gateway (ALG) is not supported with this feature.
- The maximum number of MAP-T domains supported is 10000.
- Forwarding Mapping Rule (FMR) is not supported.

Information for Configuring MAP-T

To support the MAP-T Customer Edge (CE) functionality, the current IOS-XE NAT64 architecture is used. It performs the translation of IPv4 packets to IPv6 packets, and vice versa. As MAP-T CE needs to perform the NAT44 to translate the private IPv4 address, it also utilizes the existing IOS-XE NAT44 pool-based translation to perform the NAT44 translation before going through the NAT64 translation.

The difference between MAP-E and MAP-T is mainly the packet encapsulation format. While MAP-T translates the IPv4 header to the IPv6 header (and vice versa), MAP-E encapsulates the entire IPv4 packet into the IPv6 packet. When handling the Basic Mapping Rule (BMR) and Default Mapping Rule (DMR) parameters, MAP-E and MAP-T have similar behaviour.

Description of the Algorithms

When a MAP-T domain is defined via CLI, an IPv6 and IPv4 routing entry would also be installed on the router to point to the NVI (Nat Virtual Interface). The NVI interface is a virtual interface which is used by IOS-XE NAT64 to perform the translation between IPv4 and IPv6 packets. Depending on the mode of the router (whether it is a CE or Border Router (BR)), the routes installed are different. If the router is a BR, an IPv6 routing entry is created based on the DMR IPv6 prefix, and an IPv4 routing entry is created based on the BMR IPv4 prefix. Whereas, if the router is a CE, an IPv6 routing entry is created based on the BMR IPv6 prefix. The IPv4 routing entry on the CE would need to be defined via the "nat64 route" CLI. It is a default route which points to the NVI.

When the CE receives an IPv4 packet in the LAN, it would need to determine the MAP-T domain which contains all the mapping parameters required to translate the IPv4 packet to an IPv6 packet. This is done by a longest IPv4 prefix search on the source address against the local IPv4 prefix defined in the BMR of the MAP-T domain. If a match is found, it would use the BMR and DMR parameters defined in the domain to process the packet further. In addition, it would create a NAT44 session to translate the private IPv4 source address to a public IPv4 address.

The handling of the IPv6 packet on the CE is the opposite of the IPv4 packet handling. It would first perform a longest IPv6 prefix search on the destination address against the BMR IPv6 prefix defined in the domain. If a match is found, it would translate the IPv6 header to an IPv4 header, based on the BMR and DMR parameters defined in the domain. After that, the IPv4 packet would be handled by the NAT44 component to translate the public IPv4 address to a private IPv4 address.

Configuring MAP-T

SUMMARY STEPS

1. **nat64 settingsmap-tce**
2. **nat64 map-t domain** *number***vrf***vrf-name*
3. **nat64 routevrf** *vrf-nameipv4-prefixinterface-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**nat64 settingsmap-tce**	Sets the router to be in the CE mode, instead of BR (which is the default). It must be configured before any map-t domain is defined.
Step 2	**nat64 map-t domain** *number***vrf***vrf-name*	Defines a map-t domain by optionally specifying the vrf. • The port-set-id defines the port-set id used by the CE. This is a mandatory config on the CE. • The local-ipv4-prefix is used as the selector to identify the correct domain for the local traffic. Defines a map-t domain by optionally specifying the vrf. • The port-set-id defines the port-set id used by the CE. This is a mandatory config on the CE.

	Command or Action	Purpose
		• The local-ipv4-prefix is used as the selector to identify the correct domain for the local traffic.
Step 3	**nat64 routevrf** *vrf-nameipv4-prefixinterface-name*	This CLI defines the routing to route the local traffic in a vrf to the NVI (Nat Virtual Interface), for nat64 handling.

Sample Configurations

Sample configuration on CE (same vrf on IPv4 and IPv6):

```
vrf definition vrf2
address-family ipv4
exit-address-family
address-family ipv6
exit-address-family
ipv6 unicast-routing
interface GigabitEthernet2
vrf forwarding vrf2
no ip address
nat64 enable
ipv6 address 2701:D01:4:1000:0:A601:1:1/64
ipv6 address autoconfig default
ipv6 enable
ipv6 virtual-reassembly in
interface GigabitEthernet4
vrf forwarding vrf2
ip address 100.100.0.93 255.255.255.0
nat64 enable
no ip nat service all-algs
ip nat pool pool-mapt 166.1.0.1 166.1.0.1 prefix-length 30
ip nat inside source route-map rm1 pool pool-mapt vrf vrf2 match-in-vrf overload
ip access-list extended inside-local
10 permit ip 100.100.0.0 0.0.255.255 any
route-map rm1 permit 10
match ip address inside-local
nat64 settings map-t ce
nat64 route vrf vrf2 0.0.0.0/0 GigabitEthernet2
nat64 map-t domain 1001 vrf vrf2
default-mapping-rule 3601:D01:3344:5566::/64
basic-mapping-rule
ipv6-prefix 2701:D01::/32
ipv4-prefix 166.1.0.0/18
port-parameters share-ratio 64 start-port 512
port-set-id 1
local-ipv4-prefix 100.100.0.0/16
```

Sample configuration on CE (IPv6 on global vrf):

```
vrf definition vrf2
address-family ipv4
exit-address-family
address-family ipv6
exit-address-family
ipv6 unicast-routing
interface GigabitEthernet2
no ip address
nat64 enable
```

```
ipv6 address 2701:D01:4:1000:0:A601:1:1/64
ipv6 address autoconfig default
ipv6 enable
ipv6 virtual-reassembly in
interface GigabitEthernet4
vrf forwarding vrf2
ip address 100.100.0.93 255.255.255.0
nat64 enable
no ip nat service all-algs
ip nat pool pool-mapt 166.1.0.1 166.1.0.1 prefix-length 30
ip nat inside source route-map rm1 pool pool-mapt overload
ip access-list extended inside-local
10 permit ip 100.100.0.0 0.0.255.255 any
route-map rm1 permit 10
match ip address inside-local
nat64 settings map-t ce
nat64 route vrf vrf2 0.0.0.0/0 GigabitEthernet2
nat64 map-t domain 1001
default-mapping-rule 3601:D01:3344:5566::/64
basic-mapping-rule
ipv6-prefix 2701:D01::/32
ipv4-prefix 166.1.0.0/18
port-parameters share-ratio 64 start-port 512
port-set-id 1
local-ipv4-prefix 100.100.0.0/16
```

Sample configuration on BR:

```
ipv6 unicast-routing
interface GigabitEthernet2
nat64 enable
ipv6 address 2701:D01:4:1000::9/64
ipv6 enable
ipv6 virtual-reassembly in
interface GigabitEthernet3
ip address 192.0.2.1 255.255.255.0
nat64 enable
nat64 map-t domain 1000
default-mapping-rule 3601:D01:3344:5566::/64
basic-mapping-rule
ipv6-prefix 2701:D01::/32
ipv4-prefix 166.1.0.0/18
port-parameters share-ratio 64 start-port 512
```

Configuring Mapping of Address and Port Using Translation

Before you begin

Prerequisites:

- Configure the **ipv6 enable** command on interfaces on which you configure the Mapping of Address and Port Using Translation feature.
- Configure the default mapping rule before you configure the basic mapping rule.
- While configuring mapping of address and port using translation (MAP-T), the default mapping rule (DMR) prefix, the IPv6 user prefix, and the IPv6 prefix plus the embedded address (EA) bits must be less than or equal to 64 bits, and the share ratio plus the contiguous ports plus the start port must be 16 bits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **nat64 map-t domain** *number*
4. **default-mapping-rule** *ipv6-prefix/prefix-length*
5. **basic-mapping-rule**
6. **ipv6-prefix** *prefix/length*
7. **ipv4-prefix** *prefix/length*
8. **port-parameters share-ratio** *ratio* [**start-port** *port-number*]
9. **end**
10. **show nat64 map-t domain** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter you password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**nat64 map-t domain** *number* **Example:** `Device(config)# nat64 map-t domain 1`	Configures the Network Address Translation 64 (NAT64) mapping of address and port using translation (MAP-T) domain and enters NAT64 MAP-T configuration mode.
Step 4	**default-mapping-rule** *ipv6-prefix/prefix-length* **Example:** `Device(config-nat64-mapt)# default-mapping-rule 2001:DA8:B001:FFFF::/64`	Configures the default domain mapping rule for the MAP-T domain.
Step 5	**basic-mapping-rule** **Example:** `Device(config-nat64-mapt)# basic-mapping-rule`	Configures the basic mapping rule (BMR) for the MAP-T domain and enters NAT64 MAP-T BMR configuration mode.
Step 6	**ipv6-prefix** *prefix/length* **Example:** `Device(config-nat64-mapt-bmr)# ipv6-prefix 2001:DA8:B001::/56`	Configures an IPv6 address and prefix for the MAP-T BMR.
Step 7	**ipv4-prefix** *prefix/length* **Example:** `Device(config-nat64-mapt-bmr)# ipv4-prefix 202.1.0.128/28`	Configures an IPv4 address and prefix for the MAP-T BMR.

	Command or Action	Purpose
Step 8	**port-parameters share-ratio** *ratio* [**start-port** *port-number*] **Example:** `Device(config-nat64-mapt-bmr)# port-parameters share-ratio 16 start-port 1024`	Configures port parameters for the MAP-T BMR.
Step 9	**end** **Example:** `Device(config-nat64-mapt-bmr)# end`	Exits NAT64 MAP-T BMR configuration mode and returns to privileged EXEC mode.
Step 10	**show nat64 map-t domain** *number* **Example:** `Device# show nat64 map-t domain 1`	Displays MAP-T domain information.

Example:

The following is sample output from the **show nat64 map-t domain** command:

```
Device# show nat64 map-t domain 1

MAP-T Domain 1
Mode MAP-T
Default-mapping-rule
Ip-v6-prefix 2001:DA8:B001:FFFF::/64
Basic-mapping-rule
Ip-v6-prefix 2001:DA8:B001::/56
Ip-v4-prefix 202.1.0.128/28
Port-parameters
Share-ratio 16 Contiguous-ports 64 Start-port 1024
Share-ratio-bits 4 Contiguous-ports-bits 6 Port-offset-bits 6
```

Configuration Examples for Mapping of Address and Port Using Translation

Example: Configuring Mapping of Address and Port Using Translation

```
Device# configure terminal
Device(config)# nat64 map-t domain 1
Device(config-nat64-mapt)# $ping-rule 2001:DA8:B001:FFFF::/64
Device(config-nat64-mapt)# basic-mapping-rule
Device(config-nat64-mapt-bmr)# ipv6-prefix 2001:DA8:B001::/56
Device(config-nat64-mapt-bmr)# ipv4-prefix 202.1.0.128/28
Device(config-nat64-mapt-bmr)# $ters share-ratio 16 start-port 1024
Device(config-nat64-mapt-bmr)# end
```

Example: MAP-T Deployment Scenario

The following illustration shows a mapping of address and port using translation (MAP-T) deployment scenario.

The following is the configuration for the MAP-T deployment scenario:

```
Device(config)# nat64 map-t
Device(config)# nat64 map-t domain 1
Device(config-nat64-mapt)# $ping-rule 2001:DA8:B001:FFFF::/64
Device(config-nat64-mapt)# basic-mapping-rule
Device(config-nat64-mapt-bmr)# ipv6-prefix 2001:DA8:B001::/56
Device(config-nat64-mapt-bmr)# ipv4-prefix 202.1.0.128/28
Device(config-nat64-mapt-bmr)# $ters share-ratio 16 start-port 1024
Device(config-nat64-mapt-bmr)# end
```

At the PC:

An IPv4 packet goes from 202.1.0.130 to 11.1.1.1. At the customer edge (CE) device the Mapping of address and port mapping using translation (MAP-T) function translates the packet to Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

At the border router the MAP-T border router translates the packet to

Packet goes from 192.168.1.2 ---> 74.1.1.1, source 4000, destination port : 5000

At the CPE the MAP-T CE function translates the

packet to Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

At the BR the MAP-T BR function translates the packet to

Src:203.38.102.130 Dst:74.1.1.1 SrcPort:4000 DstPort:5000

From End device:

Src:74.1.1.1 Dst:203.38.102.130 SrcPort:4000 DstPort:5000

At the BR the MAP-T BR function translates the packet to

Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

At the CE the MAP-T CE function translates the packet from

Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

To

Src:74.1.1.1 Dst:203.38.102.130 SrcPort:4000 Dstport:5000

Additional References for Mapping of Address and Port Using Translation

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title
MAP	Mapping of Address and Port (MAP)
MAP Translation	MAP Translation (MAP-T) - specification
RFC 6052	IPv6 Addressing of IPv4/IPv6 Translators
RFC 6144	Framework for IPv4/IPv6 Translation
RFC 6145	IP/ICMP Translation Algorithm

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Mapping of Address and Port Using Translation

Glossary

EA bits—Embedded address bits. The IPv4 EA bits in the IPv6 address identify an IPv4 prefix/address (or part thereof) or a shared IPv4 address (or part thereof) and a port-set identifier.

IP fragmentation—The process of breaking a datagram into a number of pieces that can be reassembled later. The IP source, destination, identification, total length, and fragment offset fields, along with the More fragments and Don't Fragment (DF) flags in the IP header, are used for IP fragmentation and reassembly. A DF bit is a bit within the IP header that determines whether a device is allowed to fragment a packet.

IPv4-translatable address—IPv6 addresses that are used to represent IPv4 hosts. These addresses have an explicit mapping relationship to IPv6 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. Both stateless and stateful translators use IPv4-translatable (also called IPv4-converted) IPv6 addresses to represent IPv4 hosts.

IPv6-translatable address—IPv6 addresses that are assigned to IPv6 hosts for stateless translation. These IPv6-translatable addresses (also called IPv6-converted addresses) have an explicit mapping relationship to IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. The stateless translator uses corresponding IPv4 addresses to represent IPv6 hosts. The stateful translator does not use IPv6-translatable addresses because IPv6 hosts are represented by the IPv4 address pool in the translator via dynamic states.

MAP rule—A set of parameters that define the mapping between an IPv4 prefix, an IPv4 address or a shared IPv4 address, and an IPv6 prefix or address. Each MAP domain uses a different mapping rule set.

MAP-T border router—A mapping of address and port using translation (MAP-T)-enabled router or translator at the edge of a MAP domain that provides connectivity to the MAP-T domain. A border relay router has at least one IPv6-enabled interface and one IPv4 interface connected to the native IPv4 network, and this router can serve multiple MAP-T domains.

MAP-T CE—A device that functions as a customer edge (CE) router in a MAP-T deployment. A typical MAP-T CE device that adopts MAP rules serves a residential site with one WAN-side interface and one or more LAN-side interfaces. A MAP-T CE device can also be referred to as a “CE” within the context of a MAP-T domain.

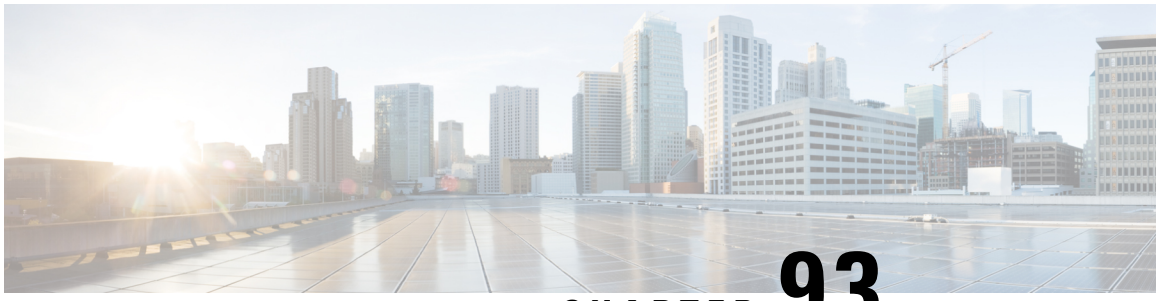
MAP-T domain—Mapping of address and port using translation (MAP-T) domain. One or more customer edge (CE) devices and a border router, all connected to the same IPv6 network. A service provider may deploy a single MAP-T domain or use multiple MAP domains.

MRT—MAP rule table. Address and port-aware data structure that supports the longest match lookups. The MRT is used by the MAP-T forwarding function.

path MTU—Path maximum transmission unit (MTU) discovery prevents fragmentation in the path between endpoints. Path MTU discovery is used to dynamically determine the lowest MTU along the path from a packet’s source to its destination. Path MTU discovery is supported only by TCP and UDP. Path MTU discovery is mandatory in IPv6, but it is optional in IPv4. IPv6 devices never fragment a packet—only the sender can fragment packets.

stateful translation—Creates a per-flow state when the first packet in a flow is received. A translation algorithm is said to be stateful if the transmission or reception of a packet creates or modifies a data structure in the relevant network element. Stateful translation allows the use of multiple translators interchangeably and also some level of scalability. Stateful translation enables IPv6 clients and peers without mapped IPv4 addresses to connect to IPv4-only servers and peers.

stateless translation—A translation algorithm that is not stateful. A stateless translation requires configuring a static translation table or may derive information algorithmically from the messages that it is translating. Stateless translation requires less computational overhead than stateful translation. It also requires less memory to maintain the state because the translation tables and the associated methods and processes exist in a stateful algorithm and do not exist in a stateless one. Stateless translation enables IPv4-only clients and peers to initiate connections to IPv6-only servers or peers that are equipped with IPv4-embedded IPv6 addresses. It also enables scalable coordination of IPv4-only stub networks or ISP IPv6-only networks. Because the source port in an IPv6-to-IPv4 translation may have to be changed to provide adequate flow identification, the source port in the IPv4-to-IPv6 direction need not be changed.



CHAPTER 93

Disabling Flow Cache Entries in NAT and NAT64

The Disabling Flow Cache Entries in NAT and NAT64 feature allows you to disable flow cache entries for dynamic and static Network Address Translation (NAT) translations. Disabling flow cache entries for dynamic and static translations saves memory usage and helps in the scaling of NAT translations.

Note Disabling flow cache entries results in lesser performance as this functionality does multiple database searches to find the most specific translation to use.

This module describes the feature and explains how to configure it.

Restrictions for Disabling Flow Cache Entries in NAT and NAT64

- You cannot disable flow cache entries in interface overload configuration because session entries are created even if flow entry creation is disabled.
- Flow cache entries are created for application layer gateway (ALG) traffic because flow-specific information needs to be stored in the session entry for ALG traffic.

Information About Disabling Flow Cache Entries in NAT and NAT64

Disabling of Flow Cache Entries Overview

By default, Network Address Translation (NAT) creates a session (which is a 5-tuple entry) for every translation. A session is also called a flow cache entry. Flow cache entries create a NAT translation for every Internet Control Message Protocol (ICMP), TCP, and UDP flow and, hence, consume a lot of system memory.

Port Address Translation (PAT) or interface overload configurations must have flow cache entries enabled. However, dynamic and static NAT configurations can disable flow cache entries. Instead of creating sessions, dynamic and static NAT translations can translate a packet off the binding (or bindings if both inside and outside bindings are available). A binding or a half entry is an association between a local IP address and a global IP address.

Note NAT, NAT64 (stateful and stateless), and carrier-grade NAT (CGN) translations support the disabling of flow cache entries.

When flow cache entry is enabled and a user has 100 sessions, 1 bind and 100 session are created. However, when flow cache entry is disabled, only one single bind is created for these sessions. Disabling flow cache entries for dynamic and static translations saves memory usage and provides more scalability for your dynamic or static translations.

Note Disabling flow cache entries will result in lesser performance as this functionality performs multiple database searches to find the most specific translation to use.

When a packet is received for translation, the following processing happens:

- If your NAT configuration is PAT, the configuration to disable flow cache entries is ignored and the packet is processed normally.
- If your configuration is not PAT, the following processing happens:
  - If the packet is an application layer gateway (ALG) packet, a session is created.
  - If the packet is a non-ALG packet, a temporary session is created and this session is sent for translation. The packet is sent to Layer 3 or Layer 4 if your configuration is NAT or to Layer 4 or Layer 7 if your configuration is NAT64 (stateful or stateless).

How to Disable Flow Cache Entries in NAT and NAT64

Disabling Flow Cache Entries in Dynamic NAT

Flow cache entries are enabled by default when Network Address Translation (NAT) is configured. To disable flow cache entries, use the **no ip nat create flow-entries** command. Perform this task to disable flow cache entries in the dynamic translation of inside source address.

Note Port Address Translation (PAT) or interface overload configuration, which is a type of dynamic NAT, requires flow cache entries. You cannot disable flow cache entries for PAT configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source source-wildcard*
5. **ip nat inside source list** *access-list-number* **pool** *name*
6. **no ip nat create flow-entries**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **ip nat inside**
10. **exit**
11. **interface** *type number*
12. **ip address** *ip-address mask*
13. **ip nat outside**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:**	Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
	`Device(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28`	
Step 4	**access-list** *access-list-number* **permit** *source source-wildcard* **Example:** `Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list that permits IP addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Establishes a dynamic source translation by specifying the pool and the access list specified in Steps 3 and 4, respectively.
Step 6	**no ip nat create flow-entries** **Example:** `Device(config)# no ip nat create flow-entries`	Disables the creation of flow cache entries.
Step 7	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 8	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 9	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Specifies an interface and enters interface configuration mode.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 13	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects an interface to the outside network.

	Command or Action	Purpose
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling Flow Cache Entries in Static NAT64

Flow cache entries are enabled by default in NAT. Perform the following task to disable flow entries in your stateful Network Address Translation 64 (NAT64) configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateful** *ipv6-prefix*/*length*
16. **nat64 v6v4 static** *ipv6-address ipv4-address*
17. **nat64 settings flow-entries disable**
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:**	Enables the forwarding of IPv6 unicast datagrams.

	Command or Action	Purpose
	`Device(config)# ipv6 unicast-routing`	
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 7	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Specifies an interface type and enters interface configuration mode.
Step 11	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 translation on an IPv4 interface.

	Command or Action	Purpose
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 15	**nat64 prefix stateful** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateful 2001:DB8:1::1/96`	Defines the stateful NAT64 prefix to be added to IPv4 hosts to translate the IPv4 address into an IPv6 address. • The stateful NAT64 prefix can be configured in global configuration mode or in interface mode.
Step 16	**nat64 v6v4 static** *ipv6-address ipv4-address* **Example:** `Device(config)# nat64 v6v4 static 2001:DB8:1::FFFE 209.165.201.1`	Enables NAT64 IPv6-to-IPv4 static address mapping.
Step 17	**nat64 settings flow-entries disable** **Example:** `Device(config)# nat64 settings flow-entries disable`	Disables flow cache entries in the NAT64 configuration.
Step 18	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Disabling Flow Cache Entries in Static CGN

Flow cache entries are enabled by default when Network Address Translation (NAT) is configured. Perform this task to disable flow cache entries in a static carrier-grade NAT (CGN) configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat settings mode cgn**
4. **ip nat inside source static** *local-ip global-ip*
5. **no ip nat create flow-entries**
6. **interface virtual-template** *number*
7. **ip nat inside**
8. **exit**
9. **interface** *type number*
10. **ip nat outside**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat settings mode cgn** **Example:** `Device(config)#  ip nat settings mode cgn`	Enables CGN operating mode.
Step 4	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 192.168.2.1 192.168.34.2`	Enables static CGN of the inside source address.
Step 5	**no ip nat create flow-entries** **Example:** `Device(config)# no ip nat create flow-entries`	Disables flow cache entries in static CGN mode.
Step 6	**interface virtual-template** *number* **Example:** `Device(config)# interface virtual-template 1`	Creates a virtual template interface that can be configured and applied dynamically when creating virtual access interfaces and enters interface configuration mode.
Step 7	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/1/1`	Specifies an interface and enters interface configuration mode.
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects an interface to the outside network.
Step 11	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

Configuration Examples for Disabling Flow Cache Entries in NAT and NAT64

Example: Disabling Flow Cache Entries in Dynamic NAT

```
Device# configure terminal
Device(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28
Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255
Device(config)# ip nat inside source list 1 pool net-208
Device(config)# no ip nat create flow-entries
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# ip address 10.114.11.39 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# ip address 172.16.232.182 255.255.255.240
Device(config-if)# ip nat outside
Device(config-if)# end
```

Example: Disabling Flow Cache Entries in Static NAT64

The following example shows a static stateful Network Address Translation 64 (NAT64):

```
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# description interface facing ipv6
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 address 2001:DB8:1::1/96
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# description interface facing ipv4
Device(config-if)# ip address 209.165.201.1 255.255.255.0
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# nat64 prefix stateful 2001:DB8:1::1/96
Device(config)# nat64 v6v4 static 2001:DB8:1::FFFE 209.165.201.1
Device(config)# nat64 settings flow-entries disable
Device(config)# end
```

Example: Disabling Flow Cache Entries in Static CGN

The following example shows a stateful carrier-grade NAT (CGN) configuration that disables the creation of flow cache entries:

```
Device# configure terminal
Device(config)# ip nat settings mode cgn
Device(config)# ip nat inside source static 192.168.2.1 192.168.34.2
Device(config)# no ip nat create flow-entries
Device(config)# interface virtual-template 1
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 2/1/1
Device(config-if)# ip nat outside
Device(config-if)# end
```

Additional References for Disabling Flow Cache Entries in NAT and NAT64

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference
Carrier-grade NAT	"Carrier-Grade Network Address Translation" module in *IP Addressing NAT Configuration Guide*
Stateful NAT64	"Stateful Network Address Translation 64" module in *IP Addressing NAT Configuration Guide*
Stateless NAT64	"Stateless Network Address Translation 64" module in *IP Addressing NAT Configuration Guide*

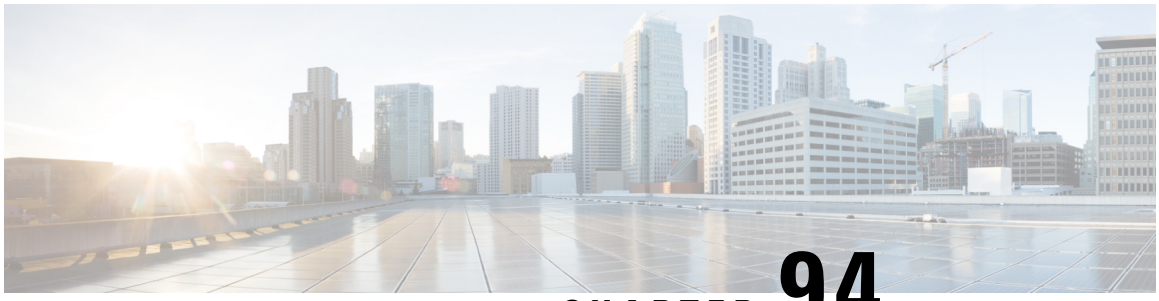
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Disabling Flow Cache Entries in NAT and NAT64

Table 135: Feature Information for Disabling Flow Cache Entries in NAT and NAT64

Feature Name	Releases	Feature Information
Disabling Flow Cache Entries in NAT and NAT64	Cisco IOS XE Release 3.10S	The Disabling of Flow Cache Entries in NAT and NAT64 feature allows you to disable flow entries for dynamic and static NAT translations. By default, flow entries are created for all Network Address Translation (NAT) translations. The following commands were introduced or modified: **ip nat create flow-entries**, **nat64 settings flow-entries disable**, and **show ip nat translations**.



CHAPTER 94

Paired-Address-Pooling Support in NAT

The ability of Network Address Translation (NAT) to consistently represent a local IP address as a single global IP address is termed paired address pooling. Paired address pooling is supported only on Port Address Translation (PAT).

Prior to the introduction of the Paired-Address-Pooling Support feature, if you have a PAT configuration, and you need a new global address or port, the next available address in the IP address pool is allocated. There was no mechanism to ensure that a local address is consistently mapped to a single global address. The Paired-Address-Pooling Support feature provides the ability to consistently map a local address to a global address.

Starting from IOS XE Polaris 16.8 release, you can specify an NAT pool for which PAP support is to be activated. This feature is helpful when you have to apply PAP support to a specific dynamic NAT traffic stream.

Restrictions for Paired-Address-Pooling Support in NAT

Paired address pooling uses more memory, and the scaling of translations is much lower than standard Network Address Translation (NAT) configuration due to the following reasons:

- Use of a new data structure that tracks each local address.
- Use of the paired-address-pooling limit. When the number of users on a global address reaches the configured limit, the next global address is used for paired address pooling. The paired-address-pooling limit uses more memory and requires more global addresses in the address pool than standard NAT.
- Two IP address pools with same IP addresses in two different mapping is not supported.

  The following example shows two non-VRF mappings. The addresses used in these two pools mappings should not overlap.

```
ip nat pool natpool1 83.0.0.56 83.0.0.56 prefix-length 24
```

```
ip nat pool natpool2 83.0.0.56 83.0.0.56 prefix-length 24
ip nat inside source list acl2 pool natpool2 overload
ip nat inside source list acl1 pool natpool1 overload
```

This following example is a combination of non-VRF and VRF-to-global mappings. In this example as well, sharing IP addresses in pools are not supported.

```
ip nat pool natpool1 82.0.0.15 82.0.0.15 prefix-length 24
ip nat pool natpool2 82.0.0.15 82.0.0.15 prefix-length 24
ip nat inside source list acl2 pool natpool2 overload             //non-vrf mapping//
ip nat inside source list acl1 pool natpool1 vrf vrf1 overload     //vrf mapping//
```

The only case where same pools can be used in two different mapping is for the **match-in-vrf** mappings.

Information About Paired-Address-Pooling Support in NAT

Paired-Address-Pooling Support Overview

An IP address pool is a group of IP addresses. You create an IP address pool by assigning a range of IP addresses and a name to it. You allocate or assign addresses in the pool to users.

The ability of Network Address Translation (NAT) to consistently represent a local IP address as a single global IP address is termed paired address pooling. A local address is any address that appears on the inside of a network, and a global address is any address that appears on the outside of the network. You can configure paired address pooling only for Port Address Translation (PAT) because dynamic and static NAT configurations are paired configurations by default. PAT, also called overloading, is a form of dynamic NAT that maps multiple, unregistered IP addresses to a single, registered IP address (many-to-one) by using different ports. Paired address pooling is supported in both classic (default) and carrier-grade NAT (CGN) mode.

In a paired-address-pooling configuration, a local address is consistently represented as a single global address. For example, if User A is paired with the global address G1, that pairing will last as long as there are active sessions for User A. If there are no active sessions, the pairing is removed. When User A has active sessions again, the user may be paired with a different global address.

If a local address initiates new sessions, and resources (ports) are insufficient for its global address, packets are dropped. When the number of users on a global address reaches the configured limit, the next global address is used for paired address pooling. When a user who is associated with a global address through paired address pooling is unable to get a port number, then the packet is dropped, the NAT drop code is incremented, and Internet Control Message Protocol (ICMP) messages are not sent.

Paired-address-pooling uses the fill-it-up method for address selection. The fill-it-up method fits (adds) the maximum possible users into a single global address before going to the next global address.

How to Configure Paired-Address-Pooling Support

Configuring Paired-Address-Pooling Support in NAT

Note If you change the Network Address Translation (NAT) configuration mode to paired-address-pooling configuration mode and vice versa, all existing NAT sessions are removed.

To configure NAT paired-address-pooling mode, use the **ip nat settings pap** command. To remove it, use the **no ip nat settings pap** command.

After you configure paired-address-pooling mode, all pool-overload mappings will act in the paired-address-pooling manner.

Based on your NAT configuration, you can use NAT static or dynamic rules.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat settings pap** [**limit** {**1000** | **120** | **250** | **30** | **500** | **60**}]
4. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
5. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
6. **ip nat inside source list** *access-list-number* **pool** *name* **overload**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **ip nat inside**
10. **exit**
11. **interface** *type number*
12. **ip address** *ip-address mask*
13. **ip nat outside**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat settings pap** [**limit** {**1000** | **120** | **250** | **30** | **500** | **60**}] **Example:** `Device(config)# ip nat settings pap`	Configures NAT paired address pooling configuration mode. • Use the **limit** keyword to limit of the number of local addresses you can use per global address. The default is 120.
Step 4	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.240`	Defines a pool of global addresses to be allocated as needed.
Step 5	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting addresses that are to be translated.
Step 6	**ip nat inside source list** *access-list-number* **pool** *name* **overload** **Example:** `Device(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic Port Address Translation (PAT) or NAT overload and specifies the access list and the IP address pool defined in Step 4 and Step 5.
Step 7	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 8	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 9	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/2`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 13	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

How to Configure Paired-Address-Pooling Support For a NAT Pool

Configuring Paired-Address-Pooling Support For a NAT Pool

Note If you change the Network Address Translation (NAT) configuration mode to paired-address-pooling configuration mode and vice versa, all existing NAT sessions are removed.

To configure NAT paired-address-pooling mode, use the **ip nat settings pap** command. To remove it, use the **no ip nat settings pap** command.

After you configure paired-address-pooling mode, all pool-overload mappings will act in the paired-address-pooling manner.

Based on your NAT configuration, you can use NAT static or dynamic rules.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat settings pap** [**limit** {**1000** | **120** | **250** | **30** | **500** | **60**}]
4. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
5. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
6. **ip nat inside source list** *access-list-number* **pool** *name* **overload**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **ip nat inside**
10. **exit**

11. **interface** *type number*
12. **ip address** *ip-address mask*
13. **ip nat outside**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat settings pap** [**limit** {**1000** | **120** | **250** | **30** | **500** | **60**}] **Example:** `Device(config)# ip nat settings pap`	Configures NAT paired address pooling configuration mode. • Use the **limit** keyword to limit of the number of local addresses you can use per global address. The default is 120.
Step 4	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.240`	Defines a pool of global addresses to be allocated as needed.
Step 5	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting addresses that are to be translated.
Step 6	**ip nat inside source list** *access-list-number* **pool** *name* **overload** **Example:** `Device(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic Port Address Translation (PAT) or NAT overload and specifies the access list and the IP address pool defined in Step 4 and Step 5.
Step 7	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 8	**ip address** *ip-address mask* **Example:**	Sets a primary IP address for the interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.114.11.39 255.255.255.0`	
Step 9	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/2`	Specifies an interface and enters interface configuration mode.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 13	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Paired-Address-Pooling Support in NAT

Example: Configuring Paired Address Pooling Support in NAT

The following example shows how to configure paired address pooling along with Network Address Translation (NAT) rules. This example shows a dynamic NAT configuration with access lists and address pools. Based on your NAT configuration, you can configure static or dynamic NAT rules.

```
Device# configure terminal
Device(config)# ip nat settings pap
Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.240
Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255
Device(config)# ip nat inside source list 1 pool net-208 overload
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# ip address 10.114.11.39 255.255.255.0
Device(config-if)# ip nat inside
```

```
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/2
Device(config-if)# ip address 172.16.232.182 255.255.255.240
Device(config-if)# ip nat outside
Device(config-if)# end
```

Additional References for Paired-Address-Pooling Support in NAT

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference

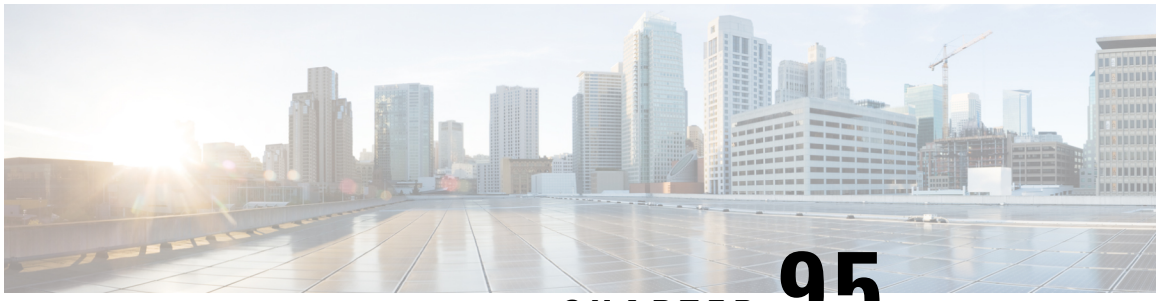
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Paired-Address-Pooling Support in NAT

Table 136: Feature Information for Paired-Address-Pooling Support in NAT

Feature Name	Releases	Feature Information
Paired-Address-Pooling Support in NAT	Cisco IOS XE Release 3.9S	The ability of Network Address Translation (NAT) to consistently represent a local IP address as a single global IP address is termed paired address pooling. Paired address pooling is supported only on Port Address Translation (PAT). The following command was introduced or modified: **ip nat settings pap**.



CHAPTER 95

Bulk Logging and Port Block Allocation

The Bulk Logging and Port Block Allocation feature allocates a block of ports for translation instead of allocating individual ports. This feature is supported only in carrier-grade Network Address Translation (CGN) mode.

This module provides information about the feature and how to configure it.

Prerequisites for Bulk Logging and Port Block Allocation

- Enable the carrier-grade Network Address Translation (CGN) mode before enabling the Bulk Logging and Port Block Allocation feature.
- Enable paired-address pooling for this feature to work.

Restrictions for Bulk Logging and Port Block Allocation

- The Bulk Logging and Port Block Allocation feature is not supported on interface overload configurations because Network Address Translation (NAT) does not own the port space, the device owns it. You can configure an interface-overload mapping with this feature; however, no messages will be logged for the configuration.
- Destination information is not logged.
- Application layer gateways (ALGs) that require consecutive port pairings only work when bulk-port allocation is configured with a step size of one. For more information on step size, see "Bulk Logging and Port Block Allocation Overview, on page 1284."
- Only bulk logging of messages is performed when this feature is enabled.

- ALG ports can be used for bulk-port allocation; however, this can cause degraded performance in sessions associated with these ports. If your configuration does not need ALGs, we recommend that you disable ALGs using the CLI.
- Syslog is not supported.
- Low ports, ports below 1024, are not supported; any application that requires a low port does not work with this feature.
- Bulk-port allocation pools must not overlap with static NAT mappings (particularly static mappings with ports) for this feature to work.
- The **ip nat service full-range** command is not supported.

Information About Bulk Logging and Port Block Allocation

Bulk Logging and Port Block Allocation Overview

The Bulk Logging and Port Block Allocation feature allocates ports to users in blocks, instead of allocating individual ports. When a session is started from inside the network, instead of allocating a single global IP address and a global port, multiple global ports of a single global IP address are allocated for Network Address Translation (NAT) of traffic. Based on the volume of translations, additional blocks of ports can be allocated.

To allocate port sets, you can use either the consecutive port-set method or the scattered port-set method. In the consecutive port-set method, a user is allocated a set of ports with consecutive port numbers. It is easy to determine the port numbers in the consecutive method and this as a result, can be a security threat.

The Bulk Logging and Port Block Allocation feature uses the scattered port-set method, which allows you to define a start port number, a step value, and the number of ports to allocate. For example, if the starting port number is 4000, the step value is four, and the number of ports is 512, then the step value of four is added to 4000 to get the second port number. Four is added again to 4004 to get the third port number and this process repeats until you have 512 ports in the port set. This method of port-set allocation provides better security.

Some application layer gateways (ALGs) require two consecutive global ports to operate correctly. These ALGs are supported with this feature only when a step value of one is configured, which allocates a consecutive port set.

You must enable NAT paired-address pooling support for this feature to work. This feature also supports Point-to-Point Tunneling Protocol (PPTP).

Note This feature is supported only in carrier-grade NAT (CGN) mode; therefore only source information is logged when this feature is configured. Destination information is not logged. For more information about CGN, see the "Carrier-Grade Network Address Translation" module in *IP Addressing: NAT Configuration Guide*.

Port Size in Bulk Logging and Port Block Allocation

Port size is configurable and determines the number of ports allocated in each port set. However, ports below 1024, also known as low ports, will not work when bulk logging and port-block allocation is configured.

The first port that is allocated is always the first port in the set. Initially, ports are likely to be allocated in a linear method; however, as sessions are released and ports are freed, the allocation is semi-random. A port set is freed when the last session referencing it is freed.

A few port sets are reserved for users using a specific global IP address. Therefore, when allocated ports are used up, a session can use a reserved port set. If all reserved port sets are used, the session is dropped.

The default port size is 512 ports, but it can differ based on the configured paired-address pooling limit. The following table provides information of the default port size when various paired-address pooling limits are configured:

Table 137: Default Port Size Based on Paired-Address Pooling Support

Paired-Address Pooling Limit	Default Bulk-Port Allocation Port Size	Maximum Port Step Size
120	512 ports	8
30	2048 ports	2
60	1024 ports	4
250	256 ports	4
500	128 ports	8
1000	64 ports	16

High-Speed Logging in Bulk Logging and Port Block Allocation

The Bulk Logging and Port Block Allocation feature reduces the volume of Network Address Translation (NAT) high-speed logging (HSL). The reduction is accomplished by dynamically allocating a block of global ports instead of a single global port.

Messages are usually logged when a session is created and destroyed. In bulk port allocation, messages are logged when a port set is allocated or freed.

The following table provides information about HSL fields, their format and value:

Table 138: HSL Field Description

Field	Format	ID	Value
Source IP address	IPv4 address	8	Varies
Translated source address	IPv4 address	225	Varies
VRF[12] ID	32-bit ID	234	Varies
Protocol	8-bit value	4	Varies
Event	8-bit value	230	• 0—Invalid • 1—Add event • 2—Delete event

Field	Format	ID	Value
UNIX timestamp in milliseconds	64-bit value	323	Varies
Port block start	16-bit port	361	Varies
Port block step size	16-bit step size	363	Varies
Number of ports in the block	16-bit number	364	Varies

[12] virtual routing and forwarding

How to Configure Bulk Logging and Port Block Allocation

Configuring Bulk Logging and Port-Block Allocation

Before you configure bulk logging and port-block allocation, you must:

- Enable carrier-grade Network Address Translation (CGN) mode.
- Enable NAT paired-address pooling.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **interface** *type number*
7. **ip nat outside**
8. **exit**
9. **ip nat settings mode cgn**
10. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
11. **access-list** *access-list-number* **permit source** [*source-wildcard*]
12. **ip nat inside source list** *access-list-number* **pool** *name*
13. **ip nat settings pap bpa set-size 512 step-size 8**
14. **ip nat log translations flow-export v9 udp destination** *addr port*
15. **end**
16. **show ip nat translations**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to Network Address Translation (NAT).
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**ip nat settings mode cgn** **Example:** `Device(config)# ip nat settings mode cgn`	Enables CGN mode.
Step 10	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.132 prefix-length 24`	Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
Step 11	**access-list** *access-list-number* **permit source** [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit source 192.168.34.0 0.0.0.255`	Defines a standard access list that permits addresses that are to be translated.
Step 12	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Establishes dynamic NAT by specifying the access list and the IP address pool defined in Step 10 and Step 11.
Step 13	**ip nat settings pap bpa set-size 512 step-size 8** **Example:** `Device(config)# ip nat settings pap bpa set-size 512 step-size 8`	Configures bulk-port allocation.
Step 14	**ip nat log translations flow-export v9 udp destination** *addr port* **Example:** `Device(config)# ip nat log translations flow-export v9 udp destination 10.1.1.1 2055`	Enables the high-speed logging (HSL) of all NAT translations.
Step 15	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 16	**show ip nat translations** **Example:** `Device# show ip nat translations`	Displays active NAT translations.

Configuration Examples for Bulk Logging and Port Block Allocation

Example: Configuring Bulk Logging and Port Block Allocation

In the following example, dynamic carrier-grade NAT (CGN) and paired-address pooling is configured for bulk-port allocation.

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip nat outside
```

```
Device(config-if)# exit
Device(config)# ip nat settings mode cgn
Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.132 prefix-length 24
Device(config)# access-list 1 permit source 192.168.34.0 0.0.0.255
Device(config)# ip nat inside source list 1 pool net-208
Device(config)# ip nat settings pap bpa set-size 512 step-size 8
Device(config)# ip nat log translations flow-export v9 udp destination 10.1.1.1 2055
Device(config)# end
```

Verifying Bulk Logging and Port Block Allocation

SUMMARY STEPS

1. **show ip nat bpa**
2. **show ip nat pool name***pool-name*

DETAILED STEPS

Step 1 show ip nat bpa

Example:

```
Device# show ip nat bpa
```

Displays Network Address Translation (NAT) bulk logging and port-block allocation settings.

The following is sample output from the **show ip nat bpa** command:

```
Device# show ip nat bpa

Paired Address Pooling (PAP)
Limit: 120 local addresses per global address
Bulk Port Allocation (BPA)
Port set size: 1024 ports in each port set allocation
Port step size: 1
Single set: True
```

Step 2 show ip nat pool name*pool-name*

Example:

```
Device# show ip nat pool name pool1
```

Displays NAT pool and port statistics.

The following is sample output from the **show ip nat pool name pool1** command:

```
Device# show ip nat pool name pool1

NAT Pool Statistics
Pool name pool1, id 1
Assigned Available
Addresses 0 5
UDP Low Ports 0 0
TCP Low Ports 0 0
UDP High Ports 0 150
TCP High Ports 0 150
(Low ports are less than 1024. High ports are greater than or equal to 1024.)
```

The following is sample output from the **show ip nat pool name pool3** command:

```
Device# show ip nat pool name pool3

NAT Pool Statistics
Pool name pool3, id 4
Assigned Available
Addresses 0 9
UDP Low Ports 0 0
TCP Low Ports 0 0
UDP High Ports 0 1080
TCP High Ports 0 1080
(Low ports are less than 1024. High ports are greater than or equal to 1024.)
```

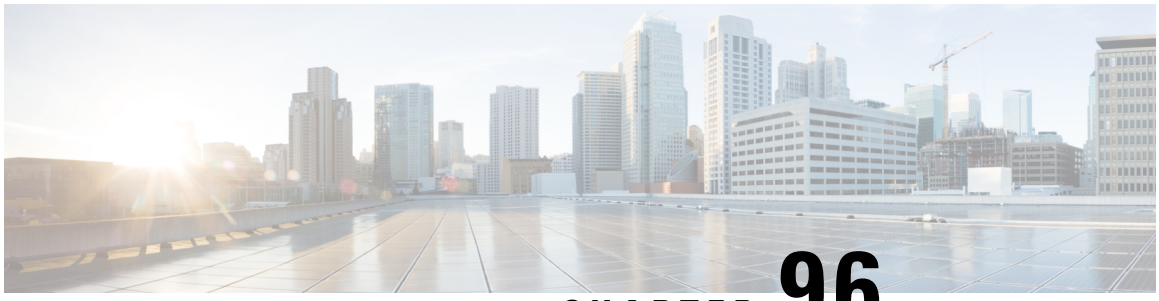
Additional References for Bulk Logging and Port Block Allocation

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Master Command List
NAT commands	Cisco IOS IP Addressing Services Command Reference
Carrier-grade NAT	"Carrier-Grade Network Address Translation" module in the *IP Addressing NAT Configuration Guide*
Paired-address pooling support	"Paired-Address Pooling Support in NAT" module in the *IP Addressing NAT Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support



CHAPTER 96

MSRPC ALG Support for Firewall and NAT

The MSRPC ALG Support for Firewall and NAT feature provides support for the Microsoft (MS) Remote Procedure Call (RPC) application-level gateway (ALG) on the firewall and Network Address Translation (NAT). The MSRPC ALG provides deep packet inspection (DPI) of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters to define match criteria that can be searched in an MSRPC packet.

The MSRPC ALG additionally supports the Virtual Transport Control Protocol (vTCP) functionality which provides a framework for various ALG protocols to appropriately handle the TCP segmentation and parse the segments in the Cisco IOS zone-based firewall, Network Address Translation (NAT) and other applications.

Prerequisites for MSRPC ALG Support for Firewall and NAT

- You must enable the Cisco IOS XE firewall and Network Address Translation (NAT) before applying the Microsoft (MS) Remote Procedure Call (RPC) application-level gateway (ALG) on packets.

Note MSRPC ALG is automatically enabled if traffic is sent to TCP port 135 by either Cisco IOS XE firewall or NAT, or both.

Restrictions for MSRPC ALG Support for Firewall and NAT

- Only TCP-based MSRPC is supported.
- You cannot configure the **allow** and **reset** commands together.
- You must configure the **match protocol msrpc** command for DPI.

- Only traffic that reaches destination port 135 is supported. This setting can be changed by configuration.

Information About MSRPC ALG Support for Firewall and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

MSRPC

MSRPC is a framework that developers use to publish a set of applications and services for servers and enterprises. RPC is an interprocess communication technique that allows the client and server software to communicate over the network. MSRPC is an application-layer protocol that is used by a wide array of Microsoft applications. MSRPC supports both connection-oriented (CO) and connectionless (CL) Distributed Computing Environment (DCE) RPC modes over a wide variety of transport protocols. All services of MSRPC establish an initial session that is referred to as the primary connection. A secondary session over a port range between 1024 to 65535 as the destination port is established by some services of MSRPC.

For MSRPC to work when firewall and NAT are enabled, in addition to inspecting MSRPC packets, the ALG is required to handle MSRPC specific issues like establishing dynamic firewall sessions and fixing the packet content after the NAT.

By applying MSRPC protocol inspection, most MSRPC services are supported, eliminating the need for Layer 7 policy filters.

MSRPC ALG on Firewall

After you configure the firewall to inspect the MSRPC protocol, the MSRPC ALG starts parsing MSRPC messages. The following table describes the types of Protocol Data Units (PDU) supported by the MSRPC ALG Support on Firewall and NAT feature:

Table 139: Supported PDU Types

PDU	Number	Type	Description
REQUEST	0	call	Initiates a call request.
RESPONSE	2	call	Responds to a call request.
FAULT	3	call	Indicates an RPC runtime, RPC stub, or RPC-specific exception.
BIND	11	association	Initiates the presentation negotiation for the body data.
BIND_ACK	12	association	Accepts a bind request.
BIND_NAK	13	association	Rejects an association request.
ALTER_CONTEXT	14	association	Requests additional presentation negotiation for another interface and/or version, or to negotiate a new security context, or both.
ALTER_CONTEXT_RESP	15	association	Responds to the ALTER_CONTEXT PDU. Valid values are accept or deny.
SHUTDOWN	17	call	Requests a client to terminate the connection and free the related resources.
CO_CANCEL	18	call	Cancels or orphans a connection. This message is sent when a client encounters a cancel fault.
ORPHANED	19	call	Terminates a request that in progress and that has not been entirely transmitted yet, or aborts a (possibly lengthy) response that is in progress.

MSRPC ALG on NAT

When NAT receives an MSRPC packet, it invokes the MSRPC ALG that parses the packet payload and forms a token to translate any embedded IP addresses. This token is passed to NAT, which translates addresses or ports as per your NAT configuration. The translated addresses are then written back into the packet payload by the MSRPC ALG.

If you have configured both the firewall and NAT, NAT calls the ALG first.

MSRPC Stateful Parser

The MSRPC state machine or the parser is the brain of the MSRPC ALG. The MSRPC stateful parser keeps all stateful information within the firewall or NAT depending on which feature invokes the parser first. The parser provides DPI of MSRPC protocol packets. It checks for protocol conformance and detects

out-of-sequence commands and malformed packets. As the packet is parsed, the state machine records various data and fills in the correct token information for NAT and firewall inspection.

How to Configure MSRPC ALG Support for Firewall and NAT

Note By default, MSRPC ALG is automatically enabled when NAT is enabled. There is no need to explicitly enable MSRPC ALG in the NAT-only configuration. You can use the **no ip nat service msrpc** command to disable MSRPC ALG on NAT.

Configuring a Layer 4 MSRPC Class Map and Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any msrpc-cmap`	Creates an inspect type class map for the traffic class and enters QoS class-map configuration mode.

	Command or Action	Purpose
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol msrpc`	Configures the match criteria for a class map on the basis of a specified protocol. • Only Cisco IOS XE stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect msrpc-pmap`	Creates a Layer 3 or Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect msrpc-class-map`	Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 8	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS XE stateful packet inspection.
Step 9	**end** **Example:** `Router(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring a Zone Pair and Attaching an MSRPC Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone-name*
4. **exit**
5. **zone security** *security-zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]]
8. **service-policy type inspect** *policy-map-name*

9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Rotuer# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *security-zone-name* **Example:** `Router(config)# zone security in-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** *security-zone-name* **Example:** `Router(config)# zone security out-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]] **Example:** `Router(config)# zone-pair security in-out source in-zone destination out-zone`	Creates a zone pair and enters security zone pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

	Command or Action	Purpose
Step 9	**end** **Example:** `Router(config-sec-zone-pair)# end`	Exits security zone pair configuration mode and enters privileged EXEC mode.

Enabling vTCP Support for MSRPC ALG

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **alg vtcp service msrpc**
4. **exit**
5. **set platform hardware qfp active feature alg msrpc tolerance on**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**alg vtcp service msrpc** **Example:** `Rotuer(config)# alg vtcp service msrpc`	Enables vTCP functionality for MSRPC ALG. **Note** By default, MSRPC ALG supports vTCP.
Step 4	**exit** **Example:** `Rotuer(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**set platform hardware qfp active feature alg msrpc tolerance on** **Example:** `Rotuer# set platform hardware qfp active feature alg msrpc tolerance on`	Enables MSRPC unknown message tolerance. **Note** By default, the tolerance is switched off.

Disabling vTCP Support for MSRPC ALG

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no alg vtcp service msrpc**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no alg vtcp service msrpc** **Example:** `Rotuer(config)# no alg vtcp service msrpc`	Disables vTCP functionality for MSRPC ALG.
Step 4	**end** **Example:** `Rotuer(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for MSRPC ALG Support for Firewall and NAT

Example: Configuring a Layer 4 MSRPC Class Map and Policy Map

```
Router# configure terminal
Router(config)# class-map type inspect match-any msrpc-cmap
Router(config-cmap)# match protocol msrpc
Router(config-cmap)# exit
Router(config)# policy-map type inspect msrpc-pmap
Router(config-pmap)# class type inspect msrpc-cmap
Router(config-pmap-c)# inspect
```

```
Router(config-pmap-c)# end
```

Example: Configuring a Zone Pair and Attaching an MSRPC Policy Map

```
Router# configure terminal
Router(config)# zone security in-zone
Router(config-sec-zone)# exit
Router(config)# zone security out-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security in-out source in-zone destination out-zone
Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap
Router(config-sec-zone-pair)# end
```

Example: Enabling vTCP Support for MSRPC ALG

```
Router# configure terminal
Router(config)# alg vtcp service msrpc
Router(config)# end
```

Example: Disabling vTCP Support for MSRPC ALG

```
Router# configure terminal
Router(config)# no alg vtcp service msrpc
Router(config)# end
```

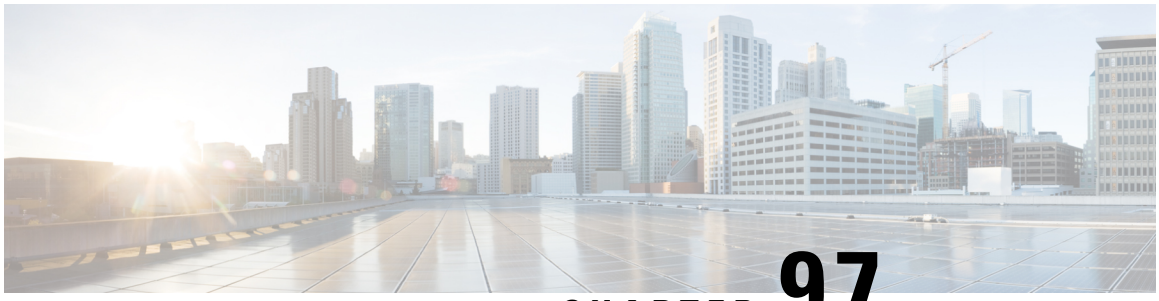
Feature Information for MSRPC ALG Support for Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 140: Feature Information for MSRPC ALG Support for Firewall and NAT

Feature Name	Releases	Feature Information
MSRPC ALG Support for Firewall and NAT	Cisco IOS XE Release 3.5S	The MSRPC ALG Support for Firewall and NAT feature provides support for the MSRPC ALG on the firewall and NAT. The MSRPC ALG provides deep packet inspection of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters that define match criteria that can be searched in an MSRPC packet. The following commands were introduced or modified: **ip nat service msrpc**, **match protocol msrpc**.
MSRPC ALG Inspection Improvements for Zone-based Firewall and NAT	Cisco IOS XE Release 3.14S	The MSRPC ALG Inspection Improvements for Zone-based Firewall and NAT feature supports Virtual Transport Control Protocol (vTCP) functionality which provides a framework for various ALG protocols to appropriately handle the TCP segmentation and parse the segments in the Cisco firewall, Network Address Translation (NAT) and other applications. The following command was introduced: **alg vtcp service msrpc**.



CHAPTER 97

Sun RPC ALG Support for Firewalls and NAT

The Sun RPC ALG Support for Firewalls and NAT feature adds support for the Sun Microsystems remote-procedure call (RPC) application-level gateway (ALG) on the firewall and Network Address Translation (NAT). Sun RPC is an application layer protocol that enables client programs to call functions in a remote server program. This module describes how to configure the Sun RPC ALG.

Restrictions for Sun RPC ALG Support for Firewalls and NAT

- If you configure the inspect action for Layer 4 or Layer 7 class maps, packets that match the Port Mapper Protocol well-known port (111) pass through the firewall without the Layer 7 inspection. Without the Layer 7 inspection, firewall pinholes are not open for traffic flow, and the Sun remote-procedure call (RPC) is blocked by the firewall. As a workaround, configure the **match program-number** command for Sun RPC program numbers.
- Only Port Mapper Protocol Version 2 is supported; none of the other versions are supported.
- Only RPC Version 2 is supported.

Information About Sun RPC ALG Support for Firewalls and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.

- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Sun RPC

The Sun remote-procedure call (RPC) application-level gateway (ALG) performs a deep packet inspection of the Sun RPC protocol. The Sun RPC ALG works with a provisioning system that allows network administrators to configure match filters. Each match filter define a match criterion that is searched in a Sun RPC packet, thereby permitting only packets that match the criterion.

In an RPC, a client program calls procedures in a server program. The RPC library packages the procedure arguments into a network message and sends the message to the server. The server, in turn, uses the RPC library and takes the procedure arguments from the network message and calls the specified server procedure. When the server procedure returns to the RPC, return values are packaged into a network message and sent back to the client.

For a detailed description of the Sun RPC protocol, see RFC 1057, *RPC: Remote Procedure Call Protocol Specification Version 2*.

Sun RPC ALG Support for Firewalls

You can configure the Sun RPC ALG by using the zone-based firewall that is created by using policies and class maps. A Layer 7 class map allows network administrators to configure match filters. The filters specify the program numbers to be searched for in Sun RPC packets. The Sun RPC Layer 7 policy map is configured as a child policy of the Layer 4 policy map with the **service-policy** command.

When you configure a Sun RPC Layer 4 class map without configuring a Layer 7 firewall policy, the traffic returned by the Sun RPC passes through the firewall, but sessions are not inspected at Layer 7. Because sessions are not inspected, the subsequent RPC call is blocked by the firewall. Configuring a Sun RPC Layer 4 class map and a Layer 7 policy allows Layer 7 inspection. You can configure an empty Layer 7 firewall policy, that is, a policy without any match filters.

Sun RPC ALG Support for NAT

By default, the Sun RPC ALG is automatically enabled when Network Address Translation (NAT) is enabled. You can use the **no ip nat service alg** command to disable the Sun RPC ALG on NAT.

How to Configure Sun RPC ALG Support for Firewalls and NAT

For Sun RPC to work when the firewall and NAT are enabled, the ALG must inspect Sun RPC packets. The ALG also handles Sun RPC-specific issues such as establishing dynamic firewall sessions and fixing the packet content after NAT translation.

Configuring the Firewall for the Sun RPC ALG

You must configure a Layer 7 Sun remote-procedure call (RPC) policy map if you have configured the inspect action for the Sun RPC protocol (that is, if you have specified the **match protocol sunrpc** command in a Layer 4 class map).

We recommend that you do not configure both security zones and inspect rules on the same interface because this configuration may not work.

Perform the following tasks to configure a firewall for the Sun RPC ALG:

Configuring a Layer 4 Class Map for a Firewall Policy

Perform this task to configure a Layer 4 class map for classifying network traffic. When you specify the **match-all** keyword with the **class-map type inspect** command, the Sun RPC traffic matches all Sun remote-procedure call (RPC) Layer 7 filters (specified as program numbers) in the class map. When you specify the **match-any** keyword with the **class-map type inspect** , the Sun RPC traffic must match at least one of the Sun RPC Layer 7 filters (specified as program numbers) in the class map.

To configure a Layer 4 class map, use the **class-map type inspect {match-any | match-all}** *classm-map-name* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** {**match-any** | **match-all**} *class-map-name*
4. **match protocol** *protocol-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect** {**match-any**	**match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect match-any sunrpc-l4-cmap`	Creates a Layer 4 inspect type class map and enters QoS class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sunrpc`	Configures a match criterion for a class map on the basis of the specified protocol.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits QoS class-map configuration mode and enters privileged EXEC mode.

Configuring a Layer 7 Class Map for a Firewall Policy

Perform this task to configure a Layer 7 class map for classifying network traffic. This configuration enables programs such as mount (100005) and Network File System (NFS) (100003) that use Sun RPC. 100005 and 100003 are Sun RPC program numbers. By default, the Sun RPC ALG blocks all programs.

For more information about Sun RPC programs and program numbers, see RFC 1057, *RPC: Remote Procedure Call Protocol Specification Version 2*.

Use the **class-map type inspect** *protocol-name* command to configure a Layer 7 class map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *protocol-name* {**match-any** | **match-all**} *class-map-name*
4. **match program-number** *program-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect** *protocol-name* {**match-any** | **match-all**} *class-map-name* **Example:** `Device(config)# class-map type inspect sunrpc match-any sunrpc-l7-cmap`	Creates a Layer 7 (application-specific) inspect type class map and enters QoS class-map configuration mode.
Step 4	**match program-number** *program-number* **Example:** `Device(config-cmap)# match program-number 100005`	Specifies the allowed RPC protocol program number as a match criterion.
Step 5	**end** **Example:**	Exits QoS class-map configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-cmap)# end`	

Configuring a Sun RPC Firewall Policy Map

Perform this task to configure a Sun remote-procedure call (RPC) firewall policy map. Use a policy map to allow packet transfer for each Sun RPC Layer 7 class that is defined in a class map for a Layer 7 firewall policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *protocol-name policy-map-name*
4. **class type inspect** *protocol-name class-map-name*
5. **allow**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *protocol-name policy-map-name* **Example:** `Device(config)# policy-map type inspect sunrpc sunrpc-l7-pmap`	Creates a Layer 7 (protocol-specific) inspect type policy map and enters QoS policy-map configuration mode.
Step 4	**class type inspect** *protocol-name class-map-name* **Example:** `Device(config-pmap)# class type inspect sunrpc sunrpc-l7-cmap`	Specifies the traffic class on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 5	**allow** **Example:** `Device(config-pmap-c)# allow`	Allows packet transfer.
Step 6	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and returns to privileged EXEC mode.

Attaching a Layer 7 Policy Map to a Layer 4 Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class** {*class-map-name* | **class-default**}
5. **inspect** [*parameter-map-name*]
6. **service-policy** *protocol-name policy-map-name*
7. **exit**
8. **class class-default**
9. **drop**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect sunrpc-l4-pmap`	Creates a Layer 4 inspect type policy map and enters QoS policy-map configuration mode.
Step 4	**class** {*class-map-name*	**class-default**} **Example:** `Device(config-pmap)# class sunrpc-l4-cmap`	Associates (class) on which an action is to be performed and enters QoS policy-map class configuration mode.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 6	**service-policy** *protocol-name policy-map-name* **Example:** `Device(config-pmap-c)# service-policy sunrpc sunrpc-l7-pmap`	Attaches the Layer 7 policy map to a top-level Layer 4 policy map.
Step 7	**exit** **Example:**	Exits QoS policy-map class configuration mode and returns to QoS policy-map configuration mode.

	Command or Action	Purpose
	`Device(config-pmap-c)# exit`	
Step 8	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies the default class (commonly known as the class-default class) before you configure its policy and enters QoS policy-map class configuration mode.
Step 9	**drop** **Example:** `Device(config-pmap-c)# drop`	Configures a traffic class to discard packets belonging to a specific class.
Step 10	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and returns to privileged EXEC mode.

Creating Security Zones and Zone Pairs and Attaching a Policy Map to a Zone Pair

You need two security zones to create a zone pair. However, you can create only one security zone and the second one can be the system-defined security zone. To create the system-defined security zone or self zone, configure the **zone-pair security** command with the **self** keyword.

Note If you select a self zone, you cannot configure the inspect action.

In this task, you will do the following:

- Create security zones.
- Define zone pairs.
- Assign interfaces to security zones.
- Attach a policy map to a zone pair.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
12. **zone-member security** *zone-name*

13. **exit**
14. **interface** *type number*
15. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
16. **zone-member security** *zone-name*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security z-client`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone or self zone as either the source or destination zone.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security z-server`	Creates a security zone and enters security zone configuration mode. • Your configuration must have two security zones to create a zone pair: a source zone and a destination zone. • In a zone pair, you can use the default zone as either the source or destination zone.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name* **Example:**	Creates a zone pair and enters security zone-pair configuration mode.

	Command or Action	Purpose
	`Device(config)# zone-pair security clt2srv source z-client destination z-server`	
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sunrpc-l4-pmap`	Attaches a firewall policy map to a zone pair.
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/0/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 192.168.6.5 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 12	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z-client`	Attaches an interface to a security zone.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 14	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 2/1/1`	Configures an interface type and enters interface configuration mode.
Step 15	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 192.168.6.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 16	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security z-server`	Attaches an interface to a security zone.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Sun RPC ALG Support for Firewall and NAT

Example: Configuring a Layer 4 Class Map for a Firewall Policy

```
Device# configure terminal
Device(config)# class-map type inspect match-any sunrpc-l4-cmap
Device(config-cmap)# match protocol sunrpc
Device(config-cmap)# end
```

Example: Configuring a Layer 7 Class Map for a Firewall Policy

```
Device# configure terminal
Device(config)# class-map type inspect sunrpc match-any sunrpc-l7-cmap
Device(config-cmap)# match program-number 100005
Device(config-cmap)# end
```

Example: Configuring a Sun RPC Firewall Policy Map

```
Device# configure terminal
Device(config)# policy-map type inspect sunrpc sunrpc-l7-pmap
Device(config-pmap)# class type inspect sunrpc sunrpc-l7-cmap
Device(config-pmap-c)# allow
Device(config-pmap-c)# end
```

Example: Attaching a Layer 7 Policy Map to a Layer 4 Policy Map

```
Device# configure terminal
Device(config)# policy-map type inspect sunrpcl4-pmap
Device(config-pmap)# class sunrpcl4-cmap
Device(config-pmap-c)# inspect
Device(config-pmap-c)# service-policy sunrpc sunrpc-l7-pmap
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# drop
Device(config-pmap-c)# end
```

Example: Creating Security Zones and Zone Pairs and Attaching a Policy Map to a Zone Pair

```
Device# configure terminal
Device(config)# zone security z-client
Device(config-sec-zone)# exit
```

```
Device(config)# zone security z-server
Device(config-sec-zone)# exit
Device(config)# zone-pair security clt2srv source z-client destination z-server
Device(config-sec-zone-pair)# service-policy type inspect sunrpc-l4-pmap
Device(config-sec-zone-pair)# exit
Device(config)# interface gigabitethernet 2/0/0
Device(config-if)# ip address 192.168.6.5 255.255.255.0
Device(config-if)# zone-member security z-client
Device(config-if)# exit
Device(config)# interface gigabitethernet 2/1/1
Device(config-if)# ip address 192.168.6.1 255.255.255.0
Device(config-if)# zone-member security z-server
Device(config-if)# end
```

Example: Configuring the Firewall for the Sun RPC ALG

The following is a sample firewall configuration for the Sun remote-procedure call (RPC) application-level gateway (ALG) support:

```
class-map type inspect sunrpc match-any sunrpc-l7-cmap
 match program-number 100005
!
class-map type inspect match-any sunrpc-l4-cmap
 match protocol sunrpc
!
!
policy-map type inspect sunrpc sunrpc-l7-pmap
 class type inspect sunrpc sunrpc-l7-cmap
  allow
!
!
policy-map type inspect sunrpc-l4-pmap
 class type inspect sunrpc-l4-cmap
  inspect
  service-policy sunrpc sunrpc-l7-pmap
!
 class class-default
  drop
!
!
zone security z-client
!
zone security z-server
!
zone-pair security clt2srv source z-client destination z-server
 service-policy type inspect sunrpc-l4-pmap
!
interface GigabitEthernet 2/0/0
 ip address 192.168.10.1 255.255.255.0
 zone-member security z-client
!
interface GigabitEthernet 2/1/1
 ip address 192.168.23.1 255.255.255.0
 zone-member security z-server
!
```

Additional References for Sun RPC ALG Support for Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
IP Addressing commands	IP Addressing Services Command Reference
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
RFC 1057	*RPC: Remote Procedure Call Protocol Specification Version 2*

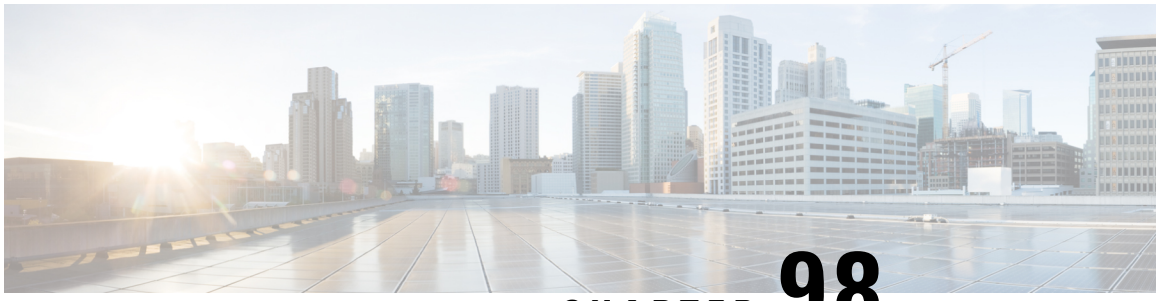
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Sun RPC ALG Support for Firewalls and NAT

Table 141: Feature Information for Sun RPC ALG Support for Firewalls and NAT

Feature Name	Releases	Feature Information
Sun RPC ALG Support for Firewalls and NAT	Cisco IOS XE Release 3.2S	The Sun RPC ALG Support for Firewalls and NAT feature adds support for the Sun RPC ALG on the firewall and NAT. The following command was introduced or modified: **match protocol**.



CHAPTER 98

vTCP for ALG Support

Virtual Transport Control Protocol (vTCP) functionality provides a framework for various Application Layer Gateway (ALG) protocols to appropriately handle the Transport Control Protocol (TCP) segmentation and parse the segments in the Cisco firewall, Network Address Translation (NAT) and other applications.

Prerequisites for vTCP for ALG Support

Your system must be running Cisco IOS XE Release 3.1 or a later Cisco IOS XE software release. The latest version of NAT or firewall ALG should be configured.

Restrictions for vTCP for ALG Support

- To aid ALG payload parsing, vTCP supports reassembly of TCP segments. In order to protect system resources, the amount of memory that vTCP can consume for reassembly is restricted to 8K for FTP, H323, LDAP, NETBIOS, PPTP, SCCP, SUNRPC, and TFTP. Connections will be reset once the limits are reached.
- vTCP does not support the high availability functionality. High availability mainly relies on the firewall or Network Address Translation (NAT) to synchronize the session information to the standby forwarding engine.
- vTCP does not support asymmetric routing. vTCP validates and assembles packet segments based on their sequence number. If packet segments that belong to the same Layer 7 message go through different devices, vTCP will not record the proper state or do an assembly of these segments.

Information About vTCP for ALG Support

Overview of vTCP for ALG Support

When a Layer 7 protocol uses TCP for transportation, the TCP payload can be segmented due to various reasons, such as application design, maximum segment size (MSS), TCP window size, and so on. The application-level gateways (ALGs) that the firewall and NAT support do not have the capability to recognize TCP fragments for packet inspection. vTCP is a general framework that ALGs use to understand TCP segments and to parse the TCP payload.

vTCP helps applications like NAT and Session Initiation Protocol (SIP) that require the entire TCP payload to rewrite the embedded data. The firewall uses vTCP to help ALGs support data splitting between packets.

When you configure firewall or NAT ALGs, the vTCP functionality is activated.

vTCP currently supports Real Time Streaming Protocol (RTSP) and DNS ALGs.

TCP Acknowledgment and Reliable Transmission

Because vTCP resides between two TCP hosts, a buffer space is required to store TCP segments temporarily, before they are sent to other hosts. vTCP ensures that data transmission occurs properly between hosts. vTCP sends a TCP acknowledgment (ACK) to the sending host if vTCP requires more data for data transmission. vTCP also keeps track of the ACKs sent by the receiving host from the beginning of the TCP flow to closely monitor the acknowledged data.

vTCP reassembles TCP segments. The IP header and the TCP header information of the incoming segments are saved in the vTCP buffer for reliable transmission.

vTCP can make minor changes in the length of outgoing segments for NAT-enabled applications. vTCP can either squeeze the additional length of data to the last segment or create a new segment to carry the extra data. The IP header or the TCP header content of the newly created segment is derived from the original incoming segment. The total length of the IP header and the TCP header sequence numbers are adjusted accordingly.

vTCP with NAT and Firewall ALGs

ALG is a subcomponent of NAT and the firewall. Both NAT and the firewall have a framework to dynamically couple their ALGs. When the firewall performs a Layer 7 inspection or NAT performs a Layer 7 fix-up, the parser function registered by the ALGs is called and ALGs take over the packet inspection. vTCP mediates between NAT and the firewall and the ALGs that use these applications. In other words, packets are first processed by vTCP and then passed on to ALGs. vTCP reassembles the TCP segments in both directions within a TCP connection.

How to Configure vTCP for ALG Support

The RTSP, DNS, NAT, and the firewall configurations enable vTCP functionality by default. Therefore no new configuration is required to enable vTCP functionality.

Enabling RTSP to Activate vTCP

Perform this task to enable RTSP packet inspection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **class class-default**
10. **exit**
11. **exit**
12. **zone security** *zone-name1*
13. **exit**
14. **zone security** *zone-name2*
15. **exit**
16. **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*
17. **service-policy type inspect** *policy-map-name*
18. **exit**
19. **interface** *type number*
20. **zone-member security** *zone-name1*
21. **exit**
22. **interface** *type number*
23. **zone-member security** *zone-name*
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:**	Creates an inspect type class map and enters class-map configuration mode.

	Command or Action	Purpose
	`Router(config)# class-map type inspect match-any rtsp_class1`	
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol rtsp`	Configures the match criteria for a class map on the basis of the named protocol. • Use DNS in place of RTSP to configure DNS as the match protocol.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect rtsp_policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect rtsp_class1`	Specifies the class on which the action is performed and enters policy-map-class configuration mode.
Step 8	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**class class-default** **Example:** `Router(config-pmap-c)# class class-default`	Specifies that these policy map settings apply to the predefined default class. If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 10	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy-map configuration mode.
Step 11	**exit** **Example:** `Router(config-pmap)# exit`	Returns to global configuration mode.
Step 12	**zone security** *zone-name1* **Example:**	Creates a security zone to which interfaces can be assigned and enters security-zone configuration mode.

	Command or Action	Purpose
	`Router(config)# zone security private`	
Step 13	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 14	**zone security** *zone-name2* **Example:** `Router(config)# zone security public`	Creates a security zone to which interfaces can be assigned and enters security-zone configuration mode.
Step 15	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 16	**zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name* **Example:** `Router(config)# zone-pair security pair-two source private destination public`	Creates a pair of security zones and enters security-zone-pair configuration mode. • To apply a policy, you must configure a zone pair.
Step 17	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy rtsp_policy`	Attaches a firewall policy map to the destination zone pair. • If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 18	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Returns to global configuration mode.
Step 19	**interface** *type number* **Example:** `Router(config)# GigabitEthernet0/1/0`	Specifies an interface for configuration. • Enters interface configuration mode.
Step 20	**zone-member security** *zone-name1* **Example:** `Router(config-if)# zone-member security private`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.

	Command or Action	Purpose
Step 21	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 22	**interface** *type number* **Example:** `Router(config)# GigabitEthernet0/1/0`	Specifies an interface for configuration. • Enters interface configuration mode.
Step 23	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security public`	Assigns an interface to a specified security zone. • When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 24	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

The following commands can be used to troubleshoot your RTSP-enabled configuration:

- **clear zone-pair**
- **show policy-map type inspect zone-pair**
- **show zone-pair security**

Configuration Examples for vTCP for ALG Support

Example RTSP Configuration

The following example shows how to configure the RTSP inspection:

```
class-map type inspect match-any rtsp_class1
match protocol rtsp
policy-map type inspect rtsp_policy
class type inspect rtsp_class1
inspect
class class-default
zone security private
```

```
zone security public
zone-pair security pair-two source private destination public
service-policy type inspect rtsp_policy
interface GigabitEthernet0/1/0
 ip address 10.0.0.1 255.0.0.0
zone-member security private
!
interface GigabitEthernet0/1/1
 ip address 10.0.1.1 255.0.0.0
  zone-member security public
```

Additional References for vTCP for ALG Support

Related Documents

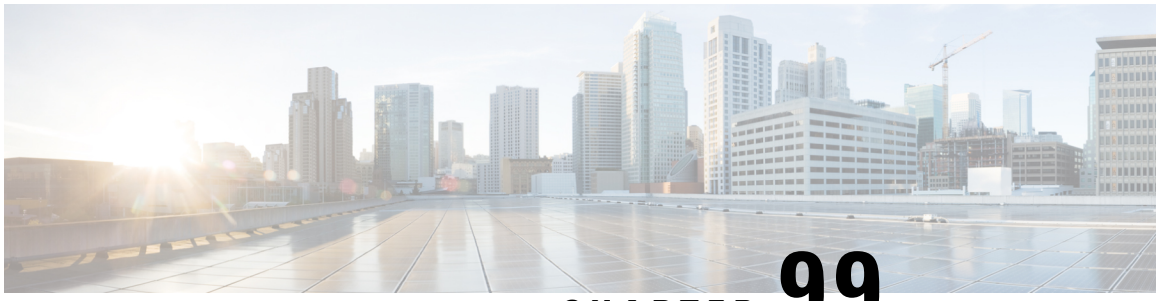
Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS firewall commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
Cisco Firewall--SIP Enhancements: ALG	*Security Configuration Guide: Securing the Data Plane*
Network Address Translation	*IP Addressing Services Configuration*

Standards and RFCs

Standard/RFC	Title
RFC 793	*Transport Control Protocol*
RFC 813	*Window and Acknowledge Strategy in TCP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 99

ALG—H.323 vTCP with High Availability Support for Firewall and NAT

The ALG—H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 application-level gateway (ALG) to support a TCP segment that is not a single H.323 message. Virtual TCP (vTCP) supports TCP segment reassembly. Prior to this introduction of the feature, the H.323 ALG processed a TCP segment only if it was a complete H.323 message. If the TCP segment was more than one message, the H.323 ALG ignored the TCP segment and the packet was passed without processing.

This module describes how to configure the ALG—H.323 vTCP with high availability (HA) support for firewalls.

Restrictions for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

- When an incoming TCP segment is not a complete H.323 message, the H.323 ALG buffers the TCP segment while waiting for the rest of the message. The buffered data is not synchronized to the standby device for high availability (HA).
- The performance of the H.323 ALG may get impacted when vTCP starts to buffer data.

Information About ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

Basic H.323 ALG Support

H.323 is a recommendation published by the ITU-T defining a series of network elements and protocols for multimedia transmission through packet-based networks. H.323 defines a number of network elements used in multimedia transmission.

Although most H.323 implementations today utilize TCP as the transport mechanism for signaling, H.323 Version 2 enables basic UDP transport.

- H.323 Terminal—This element is an endpoint in the network, providing two-way communication with another H.323 terminal or gateway.
- H.323 Gateway—This element provides protocol conversion between H.323 terminals and other terminals that do not support H.323.
- H.323 Gatekeeper—This element provides services like address translation, network access control, and bandwidth management and account for H.323 terminals and gateways.

The following core protocols are described by the H.323 specification:

- H.225—This protocol describes call signaling methods used between any two H.323 entities to establish communication.
- H.225 Registration, Admission, and Status (RAS)—This protocol is used by the H.323 endpoint and gateway for address resolution and admission control services.
- H.245—This protocol is used for exchanging the capabilities of multimedia communication and for the opening and closing of logical channels for audio, video, and data.

In addition to the protocols listed, the H.323 specification describes the use of various IETF protocols like the Real Time Transport (RTP) protocol and audio (G.711, G.729, and so on) and video (H.261, H.263, and H.264) codecs.

NAT requires a variety of ALGs to handle Layer 7 protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels. The H.323 ALG performs these specific services for H.323 messages.

Overview of vTCP for ALG Support

When a Layer 7 protocol uses TCP for transportation, the TCP payload can be segmented due to various reasons, such as application design, maximum segment size (MSS), TCP window size, and so on. The application-level gateways (ALGs) that the firewall and NAT support do not have the capability to recognize TCP fragments for packet inspection. vTCP is a general framework that ALGs use to understand TCP segments and to parse the TCP payload.

vTCP helps applications like NAT and Session Initiation Protocol (SIP) that require the entire TCP payload to rewrite the embedded data. The firewall uses vTCP to help ALGs support data splitting between packets.

When you configure firewall or NAT ALGs, the vTCP functionality is activated.

vTCP currently supports Real Time Streaming Protocol (RTSP) and DNS ALGs.

TCP Acknowledgment and Reliable Transmission

Because vTCP resides between two TCP hosts, a buffer space is required to store TCP segments temporarily, before they are sent to other hosts. vTCP ensures that data transmission occurs properly between hosts. vTCP sends a TCP acknowledgment (ACK) to the sending host if vTCP requires more data for data transmission. vTCP also keeps track of the ACKs sent by the receiving host from the beginning of the TCP flow to closely monitor the acknowledged data.

vTCP reassembles TCP segments. The IP header and the TCP header information of the incoming segments are saved in the vTCP buffer for reliable transmission.

vTCP can make minor changes in the length of outgoing segments for NAT-enabled applications. vTCP can either squeeze the additional length of data to the last segment or create a new segment to carry the extra data. The IP header or the TCP header content of the newly created segment is derived from the original incoming segment. The total length of the IP header and the TCP header sequence numbers are adjusted accordingly.

vTCP with NAT and Firewall ALGs

ALG is a subcomponent of NAT and the firewall. Both NAT and the firewall have a framework to dynamically couple their ALGs. When the firewall performs a Layer 7 inspection or NAT performs a Layer 7 fix-up, the parser function registered by the ALGs is called and ALGs take over the packet inspection. vTCP mediates between NAT and the firewall and the ALGs that use these applications. In other words, packets are first processed by vTCP and then passed on to ALGs. vTCP reassembles the TCP segments in both directions within a TCP connection.

Overview of ALG—H.323 vTCP with High Availability Support

The ALG-H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 application-level gateway (ALG) to support a TCP segment that is not a single H.323 message. After the H.323 ALG is coupled with vTCP, the firewall and NAT interact with the H.323 ALG through vTCP. When

vTCP starts to buffer data, the high availability (HA) function is impacted, because vTCP cannot synchronize the buffered data to a standby device. If the switchover to the standby device happens when vTCP is buffering data, the connection may be reset if the buffered data is not synchronized to the standby device. After the buffered data is acknowledged by vTCP, the data is lost and the connection is reset. The firewall and NAT synchronize the data for HA. vTCP only synchronizes the status of the current connection to the standby device, and in case of errors, the connection is reset.

How to Configure ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Configuring ALG-H.323 vTCP with High Availability Support for NAT

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **interface** *type number*
7. **ip nat outside**
8. **exit**
9. **ip nat pool** *pool-name start-ip end-ip* **prefix-length** *prefix-length*
10. **ip nat inside source list pool** *pool-name*
11. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Indicates that the interface is connected to the inside network (the network that is subject to NAT translation).
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 7	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**ip nat pool** *pool-name start-ip end-ip* **prefix-length** *prefix-length* **Example:** `Device(config)# ip nat pool pool1 10.1.1.10 10.1.1.100 prefix-length 24`	Defines a pool of IP addresses for NAT.
Step 10	**ip nat inside source list pool** *pool-name* **Example:** `Device(config)# ip nat inside source list pool pool1`	Enables NAT of the inside source address.
Step 11	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 10.0.0.0 255.255.255.0`	Defines a standard IP access list and permits access to packets if conditions are matched.
Step 12	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Example

The following is sample output from the **show ip nat statistics** command:

```
Device# show ip nat statistics

Total active translations: 2 (0 static, 2 dynamic; 1 extended)
Outside interfaces:
  GigabitEthernet0/0/1
Inside interfaces:
  GigabitEthernet0/1/1
Hits: 0  Misses: 25
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool pool1 refcount 2
 pool pool1: netmask 255.255.255.0
        start 10.1.1.10 end 10.1.1.100
        type generic, total addresses 91, allocated 1 (1%), misses 0
nat-limit statistics:
 max entry: max allowed 0, used 0, missed 0
Pool stats drop: 0  Mapping stats drop: 0
Port block alloc fail: 0
IP alias add fail: 0
Limit entry add fail: 0
```

The following is sample output from the **show ip nat translations** command:

```
Device# show ip nat translations

Pro  Inside global         Inside local          Outside local         Outside global
---  10.1.1.10             10.2.1.2              ---                   ---
udp  10.1.1.10:75          10.2.1.2:75           10.1.1.1:69           10.1.1.1:69
Total number of translations: 2
```

Configuration Examples for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Example: Configuring ALG-H.323 vTCP with High Availability Support for NAT

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# ip nat pool pool1 10.1.1.10 10.1.1.100 prefix-length 24
Device(config)# ip nat inside source list pool pool1
Device(config)# access-list 1 permit 10.0.0.0 255.255.255.0
Device(config)# end
```

Additional References for ALG-H.323 vTCP with High Availability Support for Firewall and NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
Firewall commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z
NAT commands	IP Addressing Services Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 142: Feature Information for ALG—H.323 vTCP with High Availability Support for Firewall and NAT

Feature Name	Releases	Feature Information
ALG—H.323 vTCP with High Availability Support for Firewall and NAT	Cisco IOS XE Release 3.7S	The ALG—H.323 vTCP with High Availability Support for Firewall and NAT feature enhances the H.323 ALG to support a TCP segment that is not a single H.323 message. vTCP supports segment reassembly. Prior to the introduction of this feature, the H.323 ALG processed a TCP segment only if it was a complete H.323 message. If the TCP segment was more than one message, the H.323 ALG ignored the TCP segment and the packet was passed without processing.

CHAPTER 100

SIP ALG Hardening for NAT and Firewall

The SIP ALG Hardening for NAT and Firewall feature provides better memory management and RFC compliance over the existing Session Initiation Protocol (SIP) application-level gateway (ALG) support for Network Address Translation (NAT) and firewall. This feature provides the following enhancements:

- Management of the local database for all SIP Layer 7 data
- Processing of the Via header
- Support for logging additional SIP methods
- Support for Provisional Response Acknowledgment (PRACK) call flow
- Support for the Record-Route header

The above enhancements are available by default; no additional configuration is required on NAT or firewall.

This module explains the SIP ALG enhancements and describes how to enable NAT and firewall support for SIP.

Restrictions for SIP ALG Hardening for NAT and Firewall

- Session Initiation Protocol (SIP) application-level gateway (ALG) does not provide any security features.
- SIP ALG manages the local database based on call IDs. There might be a corner case involving two calls coming from two different clients with the same call ID, resulting in call ID duplication.

Information About SIP ALG Hardening for NAT and Firewall

SIP Overview

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method or function on the server and at least one response.

SIP invitations that are used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to users' current locations, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols.

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

SIP ALG Local Database Management

A Session Initiation Protocol (SIP) trunk is a direct connection of an IP PBX to a service provider over an IP network using SIP. There can be numerous concurrent calls in a SIP trunk. During the call setup process, all calls use the same control channel for call establishment. More than one call uses the same control channel for call setup. When the same control channel is used by more than one call, the stateful information stored in the control-channel sessions becomes unreliable. SIP stateful information consists of media channel information such as the IP address and port number used by client and server endpoints to send media data. The media channel information is used to create a firewall pinhole and a Network Address Translation (NAT) door for the data channel in firewall and NAT, respectively. Because multiple calls use the same control channel for call setup, there will be multiple sets of media data.

In a SIP trunk, more than one call shares the same firewall and NAT session. NAT and firewall identify and manage a SIP session by using the 5 tuple in a SIP packet—source address, destination address, source port,

destination port, and protocol. The conventional method of using the 5 tuple to identify and match calls does not completely support SIP trunking and often leads to Layer 7 data memory leaks and call matching issues.

In contrast to other application-level gateways (ALGs), SIP ALG manages the SIP Layer 7 data by using a local database to store all media-related information contained in normal SIP calls and in SIP calls embedded in a SIP trunk. SIP ALG uses the Call-ID header field contained in a SIP message to search the local database for call matching and to manage and terminate calls. The Call-ID header field is a dialog identifier that identifies messages belonging to the same SIP dialog.

SIP ALG uses the call ID to perform search in the local database and to manage memory resources. In certain scenarios where SIP ALG is unable to free up a Layer 7 data record from the database, a session timer is used to manage and free resources to ensure that there are no stalled call records in the database.

Note Because all Layer 7 data is managed by SIP ALG by using a local database, SIP ALG never replies on firewall and NAT to free SIP Layer 7 data; SIP ALG frees the data by itself. If you use the **clear** command to clear all NAT translations and firewall sessions, the SIP Layer 7 data in the local database is not freed.

SIP ALG Via Header Support

A Session Initiation Protocol (SIP) INVITE request contains a Via header field. The Via header field indicates the transport paths taken by a SIP request. The Via header also contains information about the return path for subsequent SIP responses, which includes the IP address and the port to which the response message is to be sent.

SIP ALG creates a firewall pinhole or a Network Address Translation (NAT) door based on the first value in the Via header field for each SIP request received, except the acknowledge (ACK) message. If the port number information is missing from the first Via header, the port number is assumed to be 5060.

SIP ALG Method Logging Support

The SIP ALG Hardening for NAT and Firewall feature provides support for detailed logging of the following methods in Session Initiation Protocol (SIP) application-level gateway (ALG) statistics:

- PUBLISH
- OPTIONS
- 1XX (excluding 100,180,183)
- 2XX (excluding 200)

The existing SIP methods that are logged in SIP ALG statistics include ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, REFER, REGISTER, SUBSCRIBE, and 1XX-6XX.

SIP ALG PRACK Call-Flow Support

Session Initiation Protocol (SIP) defines two types of responses: final and provisional. Final responses convey the result of processing a request and are sent reliably. Provisional responses, on the other hand, provide information about the progress of processing a request but are not sent reliably.

Provisional Response Acknowledgement (PRACK) is a SIP method that provides an acknowledgment (ACK) system for provisional responses. PRACK allows reliable exchanges of SIP provisional responses between SIP endpoints. SIP reliable provisional responses ensure that media information is exchanged and resource reservation can occur before connecting the call.

SIP uses the connection, media, and attribute fields of the Session Description Protocol (SDP) during connection negotiation. SIP application-level gateway (ALG) supports SDP information within a PRACK message. If media information exists in a PRACK message, SIP ALG retrieves and processes the media information. SIP ALG also handles the creation of media channels for subsequent media streams. SIP ALG creates a firewall pinhole and a NAT door based on the SDP information in PRACK messages.

SIP ALG Record-Route Header Support

The Record-Route header field is added by a Session Initiation Protocol (SIP) proxy to a SIP request to force future requests in a SIP dialog to be routed through the proxy. Messages sent within a dialog then traverse all SIP proxies, which add a Record-Route header field to the SIP request. The Record-Route header field contains a globally reachable Uniform Resource Identifier (URI) that identifies the proxy.

SIP application-level gateway (ALG) parses the Contact header and uses the IP address and the port value in the Contact header to create a firewall pinhole and a Network Address Translation (NAT) door. In addition, SIP ALG supports the parsing of the Record-Route header to create a firewall pinhole and a NAT door for future messages that are routed through proxies.

How to Configure SIP ALG Hardening for NAT and Firewall

Enabling NAT for SIP Support

NAT support for SIP is enabled by default on port 5060. If this feature has been disabled, perform this task to re-enable NAT support for SIP. To disable the NAT support for SIP, use the **no ip nat service sip** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service sip** {**tcp** | **udp**} **port** *port-number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat service sip** {**tcp** | **udp**} **port** *port-number* **Example:** `Device(config)# ip nat service sip tcp port 5060`	Enables NAT support for SIP.
Step 4	**end** **Example:** `Device(config)# end`	Exist global configuration mode and returns to privileged EXEC mode.

Enabling SIP Inspection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **exit**
10. **class class-default**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Device(config)# class-map type inspect match-any sip-class1`	Creates an inspect type class map and enters class-map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Device(config-cmap)# match protocol sip`	Configures the match criterion for a class map based on the named protocol.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits class-map configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Device(config)# policy-map type inspect sip-policy`	Creates an inspect type policy map and enters policy-map configuration mode.
Step 7	**class type inspect** *class-map-name* **Example:** `Device(config-pmap)# class type inspect sip-class1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 8	**inspect** **Example:** `Device(config-pmap-c)# inspect`	Enables stateful packet inspection.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode and returns to policy-map configuration mode.
Step 10	**class class-default** **Example:** `Device(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 11	**end** **Example:** `Device(config-pmap)# end`	Exits policy-map configuration mode and returns to privileged EXEC mode.

Configuring a Zone Pair and Attaching a SIP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*

9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Device(config)# zone security zone2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Device(config-sec-zone)# exit`	Exits security zone configuration mode and returns to global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:** `Device(config)# zone-pair security in-out source zone1 destination zone2`	Creates a zone pair and returns to security zone-pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Device(config-sec-zone-pair)# service-policy type inspect sip-policy`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound for the device or initiated by the device) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Device(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for SIP ALG Hardening for NAT and Firewall

Example: Enabling NAT for SIP Support

```
Device> enable
Device# configure terminal
```

```
Device(config)# ip nat service sip tcp port 5060
Device(config)# end
```

Example: Enabling SIP Inspection

```
class-map type inspect match-any sip-class1
 match protocol sip
!
policy-map type inspect sip-policy
 class type inspect sip-class1
  inspect
!
class class-default
```

Example: Configuring a Zone Pair and Attaching a SIP Policy Map

```
zone security zone1
!
zone security zone2
!
zone-pair security in-out source zone1 destination zone2
 service-policy type inspect sip-policy
!
interface gigabitethernet 0/0/0
 zone security zone1
!
interface gigabitethernet 0/1/1
 zone security zone2
```

Additional References for SIP ALG Hardening for NAT and Firewall

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT configuration	*IP Addressing: NAT Configuration Guide*
Firewall configuration	*Security Configuration Guide: Zone-Based Policy Firewall*
NAT commands	Cisco IOS IP Addressing Services Command Reference
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Standards and RFCs

Standard/RFC	Title
RFC 3261	*SIP: Session Initiation Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SIP ALG Hardening for NAT and Firewall

Table 143: Feature Information for SIP ALG Hardening for NAT and Firewall

Feature Name	Releases	Feature Information
SIP ALG Hardening for NAT and Firewall	Cisco IOS XE Release 3.8S	The SIP ALG Hardening for NAT and Firewall feature provides better memory management and RFC compliance over the existing SIP ALG support for NAT and firewall.

CHAPTER 101

SIP ALG Resilience to DoS Attacks

The SIP ALG Resilience to DoS Attacks feature provides protection against Session Initiation Protocol (SIP) application layer gateway (ALG) denial of service (DoS) attacks. This feature supports a configurable lock limit, a dynamic blacklist, and configurable timers to prevent DoS attacks.

This module explains the feature and how to configure DoS prevention for the SIP application layer gateway (ALG). Network Address Translation and zone-based policy firewalls support this feature.

Information About SIP ALG Resilience to DoS Attacks

SIP ALG Resilience to DoS Attacks Overview

The SIP ALG Resilience to DoS Attacks feature provides protection against denial of service (DoS) attacks to the Session Initiation Protocol (SIP) application layer gateway (ALG). This feature supports a configurable lock limit, a dynamic blacklist, and configurable timers to prevent DoS attacks. This feature is supported by Network Address Translation (NAT) and zone-based policy firewalls.

SIP is an application-level signaling protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP DoS attacks are a major threat to networks.

The following are types of SIP DoS attacks:

- SIP register flooding: A registration flood occurs when many VoIP devices try to simultaneously register to a network. If the volume of registration messages exceeds the device capability, some messages are lost. These devices then attempt to register again, adding more congestion. Because of the network congestion, users may be unable to access the network for some time.
- SIP INVITE flooding: An INVITE flood occurs when many INVITE messages are sent to servers that cannot support all these messages. If the attack rate is very high, the memory of the server is exhausted.
- SIP broken authentication and session attack: This attack occurs when an attacker presumes the identity of a valid user, using digest authentication. When the authentication server tries to verify the identity of the attacker, the verification is ignored and the attacker starts a new request with another session identity. These attacks consume the memory of the server.

SIP ALG Dynamic Blacklist

One of the common methods of denial of service (DoS) attacks involves saturating the target network with external communication requests making the network unable to respond to legitimate traffic. To solve this issue, the SIP ALG Resilience to DoS Attacks feature uses configurable blocked lists. A blocked list is a list of entities that are denied a particular privilege, service, or access. Dynamic blacklists are disabled by default. When requests to a destination address exceed a predefined trigger criteria in the configured blocked list, the Session Initiation Protocol (SIP) application layer gateway (ALG) will drop these packets.

The following abnormal SIP session patterns are monitored by dynamic blocked lists:

- In the configured period of time if a source sends multiple requests to a destination and receives non-2xx (as per RFC 3261, any response with a status code between 200 and 299 is a "2xx response") final responses from the destination.
- In the configured period of time if a source sends multiple requests to a destination and does not receive any response from the destination.

SIP ALG Lock Limit

Both Network Address Translation (NAT) and the firewall use the Session Initiation Protocol (SIP) application layer gateway (ALG) to parse SIP messages and create sessions through tokens. To maintain session states, the SIP ALG uses a per call data structure and Layer 7 data to store call-related information that is allocated when a session is initiated and freed when a session is released. If the SIP ALG does not receive a message that indicates that the call has ended, network resources are held for the call.

Because Layer 7 data is shared between threads, a lock is required to access the data. During denial of service (DoS) and distributed DoS attacks, many threads wait to get the same lock, resulting in heavy CPU usage, which makes the system unstable. To prevent the system from becoming unstable, a limit is added to restrict the number of threads that can wait for a lock. SIP sessions are established by request/response mode. When there are too many concurrent SIP messages for one SIP call, packets that exceed the lock limit are dropped.

SIP ALG Timers

To exhaust resources on Session Initiation Protocol (SIP) servers, some denial of service (DoS) attacks do not indicate the end of SIP calls. To prevent these types of DoS attacks, a protection timer is added.

The SIP ALG Resilience to DoS Attacks feature uses the following timers:

- Call-duration timer that controls the maximum length of an answered SIP call.
- Call-proceeding timer that controls the maximum length of an unanswered SIP call.

When the configured maximum time is reached, the SIP application layer gateway (ALG) releases resources for this call, and future messages related to this call may not be properly parsed by the SIP ALG.

How to Configure SIP ALG Resilience to DoS Attacks

Configuring SIP ALG Resilience to DoS Attacks

You can configure the prevention of denial of service (DoS) parameters for the Session Initiation Protocol (SIP) application layer gateway (ALG) that is used by Network Address Translation (NAT) and the zone-based policy firewall.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **alg sip processor session max-backlog** *concurrent-processor-usage*
4. **alg sip processor global max-backlog** *concurrent-processor-usage*
5. **alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **destination** *ip-address*
6. **alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **block-time** *block-time* [**destination** *ip-address*]
7. **alg sip timer call-proceeding-timeout** *time*
8. **alg sip timer max-call-duration** *seconds*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**alg sip processor session max-backlog** *concurrent-processor-usage* **Example:** `Device(config)# alg sip processor session max-backlog 5`	Sets a per session limit for the number of backlog messages waiting for shared resources.
Step 4	**alg sip processor global max-backlog** *concurrent-processor-usage* **Example:** `Device(config)# alg sip processor global max-backlog 5`	Sets the maximum number of backlog messages waiting for shared resources for all SIP sessions.

	Command or Action	Purpose
Step 5	**alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **destination** *ip-address* **Example:** `Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 destination 10.1.1.1`	Configures dynamic SIP ALG blacklist criteria for the specified destination IP address.
Step 6	**alg sip blacklist trigger-period** *trigger-period* **trigger-size** *minimum-events* **block-time** *block-time* [**destination** *ip-address*] **Example:** `Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 block-time 30`	Configures the time period, in seconds, when packets from a source are blocked if the configured limit is exceeded.
Step 7	**alg sip timer call-proceeding-timeout** *time* **Example:** `Device(config)# alg sip timer call-proceeding-timeout 35`	Sets the maximum time interval, in seconds, to end SIP calls that do not receive a response.
Step 8	**alg sip timer max-call-duration** *seconds* **Example:** `Device(config)# alg sip timer max-call-duration 90`	Sets the maximum call duration, in seconds, for a successful SIP call.
Step 9	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying SIP ALG Resilience to DoS Attacks

Use the following commands to troubleshoot the feature.

SUMMARY STEPS

1. **enable**
2. **show alg sip**
3. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip**
4. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dbl**
5. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dblcfg**
6. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip processor**
7. **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip timer**
8. **debug alg** {**all** | **info** | **trace** | **warn**}

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show alg sip**

Displays all Session Initiation Protocol (SIP) application layer gateway (ALG) information.

Example:

```
Device# show alg sip

sip timer configuration
   Type                     Seconds
   max-call-duration        380
   call-proceeding-timeout  620

sip processor configuration
   Type           Backlog number
   session         14
   global          189

sip blacklist configuration
   dst-addr         trig-period(ms)    trig-size     block-time(sec)
   10.0.0.0             60                 30              2000
   10.1.1.1             20                 30              30
   192.0.2.115          1000               5               30
   198.51.100.34        20                 30              388
```

Step 3 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip**

Displays SIP ALG-specific statistics information in the Cisco Quantum Flow Processor (QFP).

Example:

```
Device# show platform hardware qfp active feature alg statistics sip

Events
...
Cr dbl entry:                10   Del dbl entry:                10
Cr dbl cfg entry:             8   Del dbl cfg entry:             4
start dbl trig tmr:          10   restart dbl trig tmr:       1014
stop dbl trig tmr:           10   dbl trig timeout:           1014
start dbl blk tmr:            0   restart dbl blk tmr:           0
stop dbl blk tmr:             0   dbl blk tmr timeout:           0
start dbl idle tmr:          10   restart dbl idle tmr:        361
stop dbl idle tmr:            1   dbl idle tmr timeout:          9

DoS Errors
Dbl Retmem Failed:            0    Dbl Malloc Failed:            0
DblCfg Retm Failed:           0    DblCfg Malloc Failed:         0
Session wlock ovflw:          0    Global wlock ovflw:           0
Blacklisted:                561
```

Step 4 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dbl**

Displays brief information about all SIP blocked list data.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip dbl
```

```
SIP dbl pool used chunk entries number: 1

 entry_id          src_addr         dst_addr         remaining_time(sec)
 a4a051e0a4a1ebd   10.74.30.189     10.74.5.30               25
```

Step 5 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip dblcfg**

Displays all SIP blocked list settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip dblcfg

SIP dbl cfg pool used chunk entries number: 4
 dst_addr        trig_period(ms)    trig_size    block_time(sec)
 10.1.1.1              20              30             30
 10.74.5.30            1000             5             30
 192.0.2.2             60              30           2000
 198.51.100.115        20              30            388
```

Step 6 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip processor**

Displays SIP processor settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip processor

Session:     14        Global:    189

Current global wlock count:      0
```

Step 7 **show platform hardware qfp** {**active** | **standby**} **feature alg statistics sip timer**

Displays SIP timer settings.

Example:

```
Device# show platform hardware qfp active feature alg statistics sip timer

call-proceeding:    620      call-duration:    380
```

Step 8 **debug alg** {**all** | **info** | **trace** | **warn**}

Example:

```
Device# debug alg warn
```

Enables the logging of ALG warning messages.

Configuration Examples for SIP ALG Resilience to DoS Attacks

Example: Configuring SIP ALG Resilience to DoS Attacks

```
Device# configure terminal
Device(config)# alg sip processor session max-backlog 5
Device(config)# alg sip processor global max-backlog 5
Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 destination 10.1.1.1
Device(config)# alg sip blacklist trigger-period 90 trigger-size 30 block-time 30
Device(config)# alg sip timer call-proceeding-timeout 35
Device(config)# alg sip timer max-call-duration 90
Device(config)# end
```

Additional References for SIP ALG Resilience to DoS Attacks

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Firewall commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
NAT commands	IP Addressing Services Command References

Standards and RFCs

Standard/RFC	Title
RFC 4028	*Session Timers in the Session Initiation Protocol (SIP)*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

CHAPTER 102

Match-in-VRF Support for NAT

The Match-in-VRF Support for NAT feature supports Network Address Translation (NAT) of packets that communicate between two hosts within the same VPN routing and forwarding (VRF) instance. In intra-VPN NAT, both the local and global address spaces for end hosts are isolated to their respective VPNs, and as a result, the translated addresses for the hosts overlap each other. The Match-in-VRF Support for NAT feature helps separate the address space for translated addresses among VPNs.

Restrictions for Match-in-VRF Support for NAT

- The Match-in-VRF Support for NAT feature is not supported on interface overload configuration.
- The **match-in-vrf** keyword for intra-VPN NAT is not supported with CGN.

Information About Match-in-VRF Support for NAT

Match-in-VRF Support for NAT

In Cisco IOS XE Release 3.5S and later releases, the Match-in-VRF Support for NAT feature supports NAT of packets that communicate between two hosts within the same VPN.

The VRF-aware NAT enables communication between hosts in the private address space in different VPN routing and forwarding (VRF) instances and common servers in the Internet or the global domain. Because IP addresses of the inside hosts overlap with each other, the VRF-aware NAT facilitates communication between these hosts by converting overlapped inside IP addresses into globally unique addresses. The Match-in-VRF Support for NAT feature extends VRF-aware NAT by supporting intra-VPN NAT capability. In the intra-VPN NAT, both the local and global address spaces for end hosts are isolated to their respective VPNs, and as a result translated addresses for hosts overlap each other. To separate the address space for translated addresses among VPNs, configure the **match-in-vrf** keyword in the NAT mapping (**ip nat inside**

source command) configuration. Both static and dynamic NAT configurations support the **match-in-vrf** keyword.

Note All NAT commands that support VRF support the **match-in-vrf** keyword. Because NAT outside rules (**ip nat outside source** command) support the match-in-VRF functionality by default, the **match-in-vrf** keyword is not supported by NAT outside rules.

In VRF-aware NAT, the IP alias and Address Resolution Protocol (ARP) entries for inside global addresses are configured in the global domain. For intra-VPN NAT, the IP alias and ARP entries for inside global addresses are configured in the VRF through which the translation happens. In intra-VPN NAT, configuration of the **match-in-vrf** keyword implies that at least one NAT outside interface is configured in the same VRF. The ARP entry in that VRF replies to the ARP request from the outside host.

If inside addresses are configured, the match-in-VRF is determined through inside mappings during the address translation of VRF traffic. If you have configured only outside mapping of IP addresses for address translations, the match-in-VRF will work. When a translation entry is created with both inside and outside mappings, the **match-in-vrf** keyword is determined by the inside mapping.

The Match-in-VRF Support for NAT feature supports the configuration of multiple dynamic mappings with the same IP address pool.

The following table provides you information about VRF support for NAT:

NAT Inside Interface	NAT Outside Interface
Global	Global IPv4 (non-MPLS)
MPLS IP	VRF **Note** You must use the **match-in-vrf** keyword in the configuration to indicate that communication is occurring within the VRF.
VRF	VRF **Note** Both VRFs must be in the same inside interface for this configuration to work.
VRF	MPLS **Note** You must use the **match-in-vrf** keyword in the configuration to indicate that communication is occurring within the VRF.
VRF	Global IPv4 (non-MPLS)

How to Configure Match-in-VRF Support for NAT

Configuring Static NAT with Match-in-VRF

Perform the following task to configure a static NAT translation and to enable NAT inside and outside traffic in the same VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**match-in-vrf**]]
4. **interface** *type number*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip nat inside**
7. **ip vrf forwarding** *vrf-name*
8. **exit**
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **ip nat outside**
12. **ip vrf forwarding** *vrf-name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**match-in-vrf**]] **Example:** `Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1 vrf vrf1 match-in-vrf`	Establishes static translation between an inside local address and an inside global address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 11	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside. **Note** NAT outside rules support the match-in-VRF functionality by default.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.
Step 13	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Dynamic NAT with Match-in-VRF

Perform the following task to configure a dynamic NAT translation with the same address pool and to enable NAT inside and outside traffic in the same VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source list** *access-list-number* **pool** *pool-name* [**vrf** *vrf-name* [**match-in-vrf**]]
4. **access-list** *access-list-number* **permit source** [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* [**match-in-vrf**]
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **ip vrf forwarding** *vrf-name*
10. **exit**
11. **interface** *type number*
12. **ip address** *ip-address mask*
13. **ip nat outside**
14. **ip vrf forwarding** *vrf-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source list** *access-list-number* **pool** *pool-name* [**vrf** *vrf-name* [**match-in-vrf**]] **Example:** `Router(config)# ip nat inside source list 1 pool shared-pool vrf vrf1 match-in-vrf`	Enables multiple dynamic mappings to be configured with the same address pool. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.
Step 4	**access-list** *access-list-number* **permit source** [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* [**match-in-vrf**] **Example:** `Router(config)# ip nat inside source list 1 pool shared-pool vrf vpn1`	Establishes dynamic source translation, specifying the access list defined in the previous step.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters interface configuration mode.
Step 12	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 13	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside. **Note** NAT outside rules support the match-in-VRF functionality by default.
Step 14	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface.
Step 15	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Configuration Examples for Match-in-VRF Support for NAT

Example: Configuring Static NAT with Match-in-VRF

The following example shows how to configure a static NAT translation between the local IP address 10.10.10.1 and the global IP address 172.16.131.1. The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.

```
Router# configure terminal
Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1 vrf vrf1 match-in-vrf
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 10.114.11.39 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat outside
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# end
```

Example: Configuring Dynamic NAT with Match-in-VRF

The following example shows how to configure dynamic NAT mappings with the same address pool. The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.

```
Router# configure terminal
Router(config)# ip nat inside source list 1 pool shared-pool vrf vrf1 match-in-vrf
Router(config)# access-list 1 permit 192.168.34.0 0.0.0.255
Router(config)# ip nat inside source list 1 pool shared-pool vrf vpn1
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat inside
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat outside
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# end
```

Additional References for Static NAT Mapping with HSRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
NAT commands: complete command syntax, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
IP Access List Sequence Numbering	*IP Access List Sequence Numbering* document
NAT configuration tasks	"Configuring NAT for IP Address Conservation" module
NAT maintenance	"Monitoring and Maintaining NAT" module
Using NAT with MPLS VPNs	"Integrating NAT with MPLS VPNs" module

Standards and RFCs

Standard/RFC	Title
RFC 903	*Reverse Address Resolution Protocol*
RFC 826	*Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*
RFC 1027	*Using ARP to implement transparent subnet gateways*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Match-in-VRF Support for NAT

Table 144: Feature Information for Match-in-VRF Support for NAT

Feature Name	Releases	Feature Information
Match-in-VRF Support for NAT	Cisco IOS XE Release 3.5S	The Match-in-VRF Support for NAT feature supports the NAT translation of packets that communicate between two hosts within the same VPN.

CHAPTER 103

Information About Stateless Static NAT

Static Network Address Translation (NAT) allows the user to configure one-to-one translations of the inside local addresses to the outside global addresses. It allows both IP addresses and port number translations from the inside to the outside traffic and the outside to the inside traffic.

Static NAT creates a fixed translation of private addresses to public addresses. Because static NAT assigns addresses on a one-to-one basis, you need an equal number of public addresses as private addresses. Because the public address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT enables hosts on the destination network to initiate traffic to a translated host if an access list exists that allows it .

In IOS XE Bengaluru 17.4.1a release, a new keyword **stateless** is introduced for the Cisco IOS XE static NAT configuration options. This option applies only to static NAT command. When the static mapping is set to stateless, no sessions are created for that traffic flow.

NAT Mappings and Translation Entry

If a stateless NAT mapping co-exists with other NAT mappings which are not stateless, a NAT flow entry is created in NAT translation table. Following table explains the flow creation possibilities when a flow is a match for two NAT mapping and also in redundancy and no redundancy scenario.

Table 145: NAT Mappings and Translation Entry

Mapping 1 with No Redundancy	Mapping 2 with No Redundancy	Mapping 1 with Redundancy	Mapping 2 with Redundancy	Flow Creation
Stateless	Stateful	NA	NA	Yes
Stateless	Stateless	NA	NA	No
NA	NA	Stateful	Stateless	On both active and standby

Mapping 1 with No Redundancy	Mapping 2 with No Redundancy	Mapping 1 with Redundancy	Mapping 2 with Redundancy	Flow Creation
NA	NA	Stateless	Stateless	Not on both active and standby

Restrictions for Stateless Static Network Address Translation

The following restrictions apply to the Stateless Static NAT:

- Stateless Static NAT is supported only on IPv4.
- Stateless Static NAT is supported only on default NAT mode. If you change the mode to CGN, it will fail as stateless mappings are already configured.
- Stateless Static NAT is not supported for static mapping with route-map.
- Stateless Static NAT does not support ALG processing for stateless static mappings.

Configuring Stateless Static NAT

You can cofigure the stateless static NAT on the following:

- Inside static NAT
- Outside static NAT
- Inside static NAT network
- Outside static NAT network
- Inside static NAT with PAT
- Outside static NAT with PAT

Configuring Stateless Static Inside and Outside NAT

Perform the following task to configure a static NAT translation with static mapping is set to stateless. When you set the static mapping to stateless, sessions are not created for that flow.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip* **stateless**
4. **ip nat outside source static** *global-ip local-ip* **stateless**
5. **exit**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **stateless** **Example:** `Router(config)# ip nat inside source static 10.1.1.1 100.1.1.1 stateless`	• Establishes static translation between an inside local address and an inside global address.
Step 4	**ip nat outside source static** *global-ip local-ip* **stateless** **Example:** `Router(config)# ip nat outside source static 100.1.1.1 10.1.1.1 stateless`	• Establishes static translation between an outside global address and inside local address.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Stateless Static NAT Port Forwarding

Perform the following task to configure a static NAT translation port forwarding with static mapping is set to stateless. When you set the static mapping to stateless, sessions are not created for that flow.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** {**tcp**|**udp**} *local-ip local-port global-ip global-port* **extendable Stateless**
4. **ip nat outside source static** {**tcp**|**udp**} *global-ip global-port local-ip local-port* **extendable Stateless**
5. **exit**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** {**tcp**|**udp**} *local-ip local-port global-ip global-port* **extendable Stateless** **Example:** `Router(config)# ip nat inside source static tcp 10.1.1.1 80 100.11.1.1 8080 extendable stateless`	• Establishes static translation between an inside local address and an inside global address.
Step 4	**ip nat outside source static** {**tcp**|**udp**} *global-ip global-port local-ip local-port* **extendable Stateless** **Example:** `Router(config)# ip nat outside source static tcp 100.1.1.1 8080 10.1.1.1 80 extendable stateless`	• Establishes static translation between an outside global address and inside local address.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Stateless Static NAT Network

Perform the following task to configure a static NAT translation network with static mapping is set to stateless. When you set the static mapping to stateless, sessions are not created for that flow.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static network** *local-network-mask global-network-mask* **Stateless**
4. **ip nat outside source static network** *global-network-mask local-network-mask* **Stateless**
5. **exit**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static network** *local-network-mask global-network-mask* **Stateless** **Example:** `Router(config)# ip nat inside source static network 10.0.0.0 100.1.1.0 /24 stateless`	• Establishes static translation between an inside local network and an inside global network.
Step 4	**ip nat outside source static network** *global-network-mask local-network-mask* **Stateless** **Example:** `Router(config)# ip nat outside source static network 100.0.0.0 10.1.1.0 /24 stateless`	• Establishes static translation between a outside global network and an inside local network.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Stateless Static NAT with VRF

Perform the following task to configure a static NAT translation with static mapping is set to stateless in VRF aware NAT scenario. When you set the static mapping to stateless, sessions are not created for that flow.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**match-in-vrf**]] **Stateless**
4. **ip nat outside source static** *global-ip local-ip* [**vrf** *vrf-name* [**match-in-vrf**]] **Stateless**
5. **exit**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**match-in-vrf**]] **Stateless** **Example:** `Router(config)# ip nat inside source static 10.1.1.1 100.11.1.1 vrf vrf1 match-in-vrf stateless`	Establishes static translation between an inside local address and an inside global address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF. • The **Stateless** keyword does not create the flow entries for static mapping.
Step 4	**ip nat outside source static** *global-ip local-ip* [**vrf** *vrf-name* [**match-in-vrf**]] **Stateless** **Example:** `Router(config)# ip nat outside source static 100.1.1.1 10.1.1.1 vrf vrf1 match-in-vrf stateless`	Establishes static translation between a outside global address and an inside local address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF. • The **Stateless** keyword does not create the flow entries for static mapping.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Stateless Static NAT with Static Stateless Static NAT Port Forwarding

Perform the following task to configure a static NAT port forwarding with VRF with static mapping is set to stateless. When you set the static mapping to stateless, sessions are not created for that flow.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip nat inside source static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**vrf** *vrf-name* [**match-in-vrf**]] **extendable stateless**
4. **ip nat outside source static** {**tcp** | **udp**} *global-ip global-port local-ip local-port* [**vrf** *vrf-name* [**match-in-vrf**]] **extendable stateless**
5. **exit**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**vrf** *vrf-name* [**match-in-vrf**]] **extendable stateless** **Example:** `Router(config)# ip nat inside source static tcp 10.1.1.1 80 100.11.1.1 8080 vrf 1 match-in-vrf extendable stateless`	Establishes static translation between an inside local address and an inside global address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF. • The **Stateless** keyword does not create the flow entries for static mapping.
Step 4	**ip nat outside source static** {**tcp** | **udp**} *global-ip global-port local-ip local-port* [**vrf** *vrf-name* [**match-in-vrf**]] **extendable stateless** **Example:** `Router(config)# ip nat outside source static tcp 100.1.1.1 8080 10.1.1.1 80 vrf 1 match-in-vrf extendable stateless`	Establishes static translation between a outside global address and an inside local address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF. • The **Stateless** keyword does not create the flow entries for static mapping.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Static Stateful NAT with Static Stateless NAT in Redundant Device

Perform the following task to configure a static NAT translation with static mapping is set to stateless. When you set the static mapping to stateless, sessions are not created for that flow. In this configuration, only on static mapping is set to stateless. A NAT translation entry is created when the flow matches to both mapping statements or if it matches to stateful mapping entry only. However, it will not be created if it matches to stateless entry only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**redundancy group name** [**match-in-vrf**]]] **stateless**
4. **ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**redundancy group name match-in-vrf**]]] **stateless**
5. **exit**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**redundancy group name** [**match-in-vrf**]]] **stateless** **Example:** `Router(config)# ip nat inside source static 10.180.4.4 10.236.214.218 vrf vrf1 redundancy 1 mapping-id 11 match-in-vrf stateless`	Establishes static translation between an inside local address and an inside global address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF. • The **Stateless** keyword does not create the flow entries for static mapping.
Step 4	**ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**redundancy group name match-in-vrf**]]] **stateless** **Example:** `Router(config)# ip nat outside source static 10.180.4.8 10.240.214.220 vrf vrf1 redundancy 1 mapping-id 10 match-in-vrf stateless`	Establishes static translation between an inside local address and an inside global address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF. • The **Stateless** keyword does not create the flow entries for static mapping.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Configuring Stateless Static NAT

Stateless Static NAT

The following example shows how to configure a stateless static inside and outside NAT translation between the local IP address 10.1.1.1 and the global IP address 100.1.1.1. The **Stateless** keyword does not create the flow entries for static mapping.

```
Router# configure terminal
Router(config)# ip nat inside source static 10.1.1.1 100.1.1.1 stateless
Router(config)# ip nat outside source static 100.1.1.1 10.1.1.1 stateless
```

Stateless Static NAT with Port Forwarding

The following example shows how to configure a stateless static NAT port forwarding translation between the local IP address 10.1.1.1 and the global IP address 100.1.1.1. The **Stateless** keyword does not create the flow entries for static mapping.

```
Router# configure terminal
Router(config)# ip nat inside source static tcp 10.1.1.1 80 100.11.1.1 8080 extendable
stateless
Router(config)# ip nat outside source static tcp 100.1.1.1 8080 10.1.1.1 80 extendable
stateless
```

Stateless Static NAT Network

The following example shows how to configure a stateless static NAT network between an inside local network and an inside global network. The **Stateless** keyword does not create the flow entries for static mapping.

```
Router# configure terminal
Router(config)# ip nat inside source static network 10.0.0.0 100.1.1.0 /24 stateless
Router(config)# ip nat outside source static network 100.0.0.0 10.1.1.0 /24 stateless
```

Static Stateless NAT with VRF

The following example shows how to configure a stateless static NAT translation between the local IP address 10.1.1.1 and the global IP address 100.1.1.1. The **match-in-vrf** keyword enables NAT

inside and outside traffic in the same VRF. The **Stateless** keyword does not create the flow entries for static mapping.

```
Router# configure terminal
Router(config)# ip nat inside source static 10.1.1.1 100.11.1.1 vrf vrf1 match-in-vrf
stateless
Router(config)# ip nat outside source static 100.1.1.1 10.1.1.1 vrf vrf1 match-in-vrf
stateless
Router(config)# Router(config-if)# end
```

Static Stateless NAT with Static Stateless Static NAT Port Forwarding

The following example shows how to configure a stateless static NAT translation between the local IP address 10.1.1.1 and the global IP address 100.1.1.1. The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF. The **Stateless** keyword does not create the flow entries for static mapping.

```
Router# configure terminal
Router(config)# ip nat inside source static tcp 10.1.1.1 80 100.11.1.1 8080 vrf 1 match-in-vrf
 extendable stateless
Router(config)# ip nat outside source static tcp 100.1.1.1 8080 10.1.1.1 80 vrf 1 match-in-vrf
 extendable stateless
Router(config)# Router(config-if)# end
```

Static Stateful NAT with Static Stateless NAT in Device-to-Device HA

The following example shows how to configure a stateless static NAT with static stateless NAT matching the flow with device-to-device redundancy enabled.

```
Router# configure terminal
ip nat inside source static 10.180.4.4 10.236.214.218 vrf vrf1 redundancy 1 mapping-id 11
match-in-vrf stateless
ip nat outside source static 10.180.4.8 10.240.214.220 vrf vrf1 redundancy 1 mapping-id 10
 match-in-vrf stateless
```

Feature Information for Statless Static NAT

Table 146: Feature Information for Statless Static NAT

Feature Name	Releases	Feature Information
Statless Static NAT	Cisco IOS XE Bengaluru 17.4	A new keyword **stateless** is introduced for IOS XE static NAT configuration.

CHAPTER 104

IP Multicast Dynamic NAT

The IP Multicast Dynamic Network Address Translation (NAT) feature supports the source address translation of multicast packets. You can use source address translation when you want to connect to the Internet, but not all your hosts have globally unique IP addresses. NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. The IP multicast dynamic translation establishes a one-to-one mapping between an inside local address and one of the addresses from the pool of outside global addresses.

Restrictions for IP Multicast Dynamic NAT

The IP Multicast Dynamic NAT feature does not support:

- IPv4-to-IPv6 address translation.
- Multicast destination address translation.
- Port Address Translation (PAT) overloading for multicast.
- Source and destination address translation.
- Unicast-to-multicast address translation.

Note To configure multicast ACL for a NAT inside interface, ensure that you configure the ACL to allow IP addresses before and after NAT translation. If you do not configure the ACL to permit IP addresses after NAT translation, the MFIB table does not contain (S,G) entry and this can cause issues in certain deployments.

Information About IP Multicast Dynamic NAT

How NAT Works

A device that is configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit device between a stub domain and the backbone. When a packet exits the domain, NAT translates the locally significant source address into a globally unique address. When a packet enters the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. If NAT cannot allocate an address because it has run out of addresses, it drops the packet. Then, NAT sends an Internet Control Message Protocol (ICMP) host unreachable packet to the destination.

Uses of NAT

NAT can be used for the following applications:

- When you want to connect to the Internet, but not all of your hosts have globally unique IP addresses. NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT is configured on the router at the border of a stub domain (referred to as the *inside network*) and a public network such as the Internet (referred to as the *outside network*). NAT translates internal local addresses to globally unique IP addresses before sending packets to the outside network. As a solution to the connectivity problem, NAT is practical only when relatively few hosts in a stub domain communicate outside of the domain at the same time. When this is the case, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses when outside communication is necessary, and these addresses can be reused when they are no longer in use.
- When you must change your internal addresses. Instead of changing the internal addresses, which can be a considerable amount of work, you can translate them by using NAT.
- When you want to do basic load sharing of TCP traffic. You can map a single global IP address to many local IP addresses by using the TCP load distribution feature.

NAT Inside and Outside Addresses

The term *inside* in a Network Address Translation (NAT) context refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network have addresses in one space (known as the *local* address space). These hosts appear to those users outside the network as being in another space (known as the *global* address space).

Similarly, the term *outside* refers to those networks to which the stub network connects, and which are not under the control of an organization. Also, hosts in outside networks can be subject to translation, and can thus have local and global addresses. NAT uses the following definitions:

- Inside local address—An IP address that is assigned to a host on the inside network. The address that the Network Information Center (NIC) or service provider assigns is probably not a legitimate IP address.
- Inside global address—A legitimate IP address assigned by the NIC or service provider that represents one or more inside local IP addresses to the outside world.

- Outside local address—The IP address of an outside host as it appears to the inside network. Not necessarily a legitimate address, it is allocated from the address space that is routable on the inside.
- Outside global address—The IP address that is assigned to a host on the outside network by the owner of the host. The address is allocated from a globally routable address or network space.

NAT supports the following VRFs:

Table 147: VRF NAT Support

NAT Inside Interface	NAT Outside Interface	Condition
Global VRF (also referred to as a non-VRF interface)	Global VRF (also referred to as a non-VRF interface)	Normal
VRF X	Global VRF (also referred to as a non-VRF interface)	When NAT is not configured for Match-in-VRF support. For more details, see the *Match-in-VRF Support for NAT* chapter.
VRF X	VRF X	When both inside and outside interfaces are in the same VRF, and NAT is configured with Match-in-VRF support.

This section describes the following topics:

- Inside Source Address Translation, on page 1008
- Overloading of Inside Global Addresses, on page 1010

Dynamic Translation of Addresses

Dynamic translation establishes a mapping between an inside local address and a pool of global addresses. Dynamic translation is useful when multiple users on a private network access the Internet. The dynamically configured pool IP address may be used as needed. It is released for use by other users when access to the Internet is no longer required.

Note When inside global or outside local addresses belong to a directly connected subnet on a NAT router, the router adds IP aliases for them. This action enables answer Address Resolution Protocol (ARP) requests. However, a situation can arise where the router itself answers packets that are not destined for it, possibly causing a security issue. This security issue can happen when an incoming Internet Control Message Protocol (ICMP) or UDP packet that is destined for one of the aliased addresses does not have a corresponding NAT translation in the NAT table. The router itself runs a corresponding service, for example, the Network Time Protocol (NTP). Such a situation might cause minor security risks.

How to Configure IP Multicast Dynamic NAT

Configuring IP Multicast Dynamic NAT

Note IP multicast dynamic translation establishes a one-to-one mapping between an inside local address and one of the addresses from the pool of outside global addresses

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} [**type** {**match-host** | **rotary**}]
4. **access-list** *access-list-number* **permit** *source-address wildcard-bits* [**any**]
5. **ip nat inside source list** *access-list-number* **pool** *name*
6. **ip multicast-routing distributed**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **ip pim sparse-mode**
10. **ip nat inside**
11. **exit**
12. **interface** *type number*
13. **ip address** *ip-address mask*
14. **ip pim sparse-mode**
15. **ip nat outside**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} [**type** {**match-host** | **rotary**}] **Example:** `Router(config)# ip nat pool mypool 10.41.10.1 10.41.10.23 netmask 255.255.255.0`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source-address wildcard-bits* [**any**] **Example:** `Router(config)# access-list 100 permit 10.3.2.0 0.0.0.255 any`	Defines a standard access list for the inside addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat inside source list 100 pool mypool`	Establishes dynamic source translation, specifying the access list defined in the prior step.
Step 6	**ip multicast-routing distributed** **Example:** `Router(config)# ip multicast-routing distributed`	Enables Multicast Distributed Switching (MDS).
Step 7	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 8	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 9	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode operation of Protocol Independent Multicast (PIM) on an interface.
Step 10	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Indicates that the interface is connected to the inside network (the network that is subject to NAT translation).
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 12	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 13	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.2.2.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 14	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode operation of PIM on an interface.
Step 15	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.
Step 16	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for IP Multicast Dynamic NAT

Example: Configuring IP Multicast Dynamic NAT

```
Router# configure terminal
Router(config)# ip nat pool mypool 10.41.10.1 10.41.10.23 netmask 255.255.255.0
Router(config)# access-list 100 permit 10.3.2.0 0.0.0.255 any
Router(config)# ip nat inside source list 100 pool mypool
Router(config)# ip multicast-routing distributed
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# ip pim sparse-mode
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 10.2.2.1 255.255.255.0
Router(config-if)# ip pim sparse-mode
Router(config-if)# ip nat outside
Router(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference
Configuring NAT for IP address conservation	Configuring NAT for IP Address Conservation module

Standards and RFCs

Standard/RFC	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP Multicast Dynamic NAT

Table 148: Feature Information for IP Multicast Dynamic NAT

Feature Name	Releases	Feature Information
IP Multicast Dynamic NAT	Cisco IOS XE Release 3.4S	The IP Multicast Dynamic Network Address Translation feature supports the source address translation of multicast packets. You can use source address translation when you want to connect to the Internet, but not all your hosts have globally unique IP addresses. NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. The IP multicast dynamic translation establishes a one-to-one mapping between an inside local address and one of the addresses from the pool of outside global addresses.

CHAPTER 105

PPTP Port Address Translation

The PPTP Port Address Translation feature supports the Point-to-Point Tunneling Protocol (PPTP) application layer gateway (ALG) for Port Address Translation (PAT) configuration. PAT configuration requires the PPTP ALG to parse PPTP packets. The PPTP ALG is enabled by default when Network Address Translation (NAT) is configured.

This module provides information about how to configure the PPTP ALG for PAT.

Restrictions for PPTP Port Address Translation

- The Point-to-Point Tunneling Protocol (PPTP) application layer gateway (ALG) does not support virtual TCP (vTCP) and TCP segments.
- The PPTP ALG will not work in Carrier Grade Network Address Translation (NAT) mode, when the NAT client and server use the same call ID.

Information About PPTP Port Address Translation

PPTP ALG Support Overview

The Point-to-Point Tunneling Protocol (PPTP) is a network protocol that enables the secure transfer of data from a remote client to an enterprise server by creating a VPN across TCP/IP-based data networks. PPTP encapsulates PPP packets into IP datagrams for transmission over the Internet or other public TCP/IP-based networks.

PPTP establishes a tunnel for each communicating PPTP network server (PNS)-PPTP Access Concentrator (PAC) pair. After the tunnel is set up, PPP packets are exchanged using enhanced generic routing encapsulation (GRE). A call ID present in the GRE header indicates the session to which a particular PPP packet belongs.

Network Address Translation (NAT) translates only the IP address and the port number of a PPTP message. Static and dynamic NAT configurations work with PPTP without the requirement of the PPTP application layer gateway (ALG). However, Port Address Translation (PAT) configuration requires the PPTP ALG to parse the PPTP header and facilitate the translation of call IDs in PPTP control packets. NAT then parses the GRE header and translates call IDs for PPTP data sessions. The PPTP ALG does not translate any embedded IP address in the PPTP payload. The PPTP ALG is enabled by default when NAT is configured.

NAT recognizes PPTP packets that arrive on the default TCP port, 1723, and invokes the PPTP ALG to parse control packets. NAT translates the call ID parsed by the PPTP ALG by assigning a global address or port number. Based on the client and server call IDs, NAT creates two doors based on the request of the PPTP ALG. (A door is created when there is insufficient information to create a complete NAT-session entry. A door contains information about the source IP address and the destination IP address and port.) Two NAT sessions are created (one with the server call ID and the other with the client call ID) for two-way data communication between the client and server. NAT translates the GRE packet header for data packets that complies with RFC 2673.

PPTP is a TCP-based protocol. Therefore, when NAT recognizes a TCP packet as a PPTP packet, it invokes the PPTP ALG parse-callback function. The PPTP ALG fetches the embedded call ID from the PPTP header and creates a translation token for the header. The PPTP ALG also creates data channels for related GRE tunnels. After ALG parsing, NAT processes the tokens created by the ALG.

PPTP Default Timer

The default timer for PPTP is 24 hours. This means that a generic routing encapsulation (GRE) session will live for 24 hours when deploying static and dynamic NAT. Based on your PPTP configuration and scaling requirement, you adjust the PPTP default timer.

Some PPTP clients and servers send keepalive messages to keep GRE sessions alive. You can adjust the NAT session timer for PPTP sessions by using the **ip nat translation pptp-timeout** command.

How to Configure PPTP Port Address Translation

Configuring PPTP ALG for Port Address Translation

The Point-to-Point Tunneling Protocol (PPTP) application layer gateway (ALG) is enabled by default when Network Address Translation (NAT) is configured. Use the **no ip nat service pptp** command to disable the PPTP ALG. Use the **ip nat service pptp** command to reenable PPTP ALG translation of applications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **interface** *type number*
7. **ip nat outside**
8. **exit**
9. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}

10. **ip nat inside source list** {*access-list-number* | *access-list-name*} **pool** *name* **overload**
11. **ip access-list standard** *access-list-name*
12. **permit** *host-ip*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Enables an interface and enters interface configuration mode.
Step 4	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Enables an interface and enters interface configuration mode.
Step 7	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool pptp-pool 192.168.0.1 192.168.0.234 prefix-length 24`	Defines a pool of IP addresses for NAT translations.

	Command or Action	Purpose
Step 10	**ip nat inside source list** {*access-list-number* | *access-list-name*} **pool** *name* **overload** **Example:** `Device(config)# ip nat inside source list pptp-acl pool pptp-pool overload`	Enables NAT of the inside source address. • When overloading is configured, the TCP or UDP port number of each inside host distinguishes between multiple conversations by using the same local IP address.
Step 11	**ip access-list standard** *access-list-name* **Example:** `Device(config)# ip access-list standard pptp-acl`	Defines a standard IP access list by name to enable packet filtering and enters standard access-list configuration mode.
Step 12	**permit** *host-ip* **Example:** `Device(config-std-nacl)# permit 10.1.1.1`	Sets conditions in named IP access lists that permit packets.
Step 13	**end** **Example:** `Device(config-std-nacl)# end`	Exits standard access-list configuration mode and enters privileged EXEC mode.

Configuration Examples for PPTP Port Address Translation

Example: Configuring PPTP ALG for Port Address Translation

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/1/0
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# ip nat pool pptp-pool 192.168.0.1 192.168.0.234 prefix-length 24
Device(config)# ip nat inside source list pptp-acl pool pptp-pool overload
Device(config)# ip access-list standard pptp-acl
Device(config-std-nacl)# permit 10.1.1.1
Device(config-std-nacl)# end
```

Additional References for PPTP Port Address Translation

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2637	*Point-to-Point Tunneling Protocol (PPTP)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for PPTP Port Address Translation

Table 149: Feature Information for PPTP Port Address Translation

Feature Name	Releases	Feature Information
PPTP Port Address Translation Support	Cisco IOS XE Release 3.9S	The PPTP Port Address Translation Support feature introduces the Point-to-Point Tunneling Protocol (PPTP) application layer gateway (ALG) for Port Address Translation (PAT) configuration. PAT configuration requires the PPTP ALG to parse PPTP packets. The PPTP ALG is enabled by default when Network Address Translation (NAT) is configured. The following commands were introduced or modified: **debug platform hardware qfp feature alg datapath pptp**, **ip nat service pptp**, **show platform hardware qfp feature alg statistics pptp**.

CHAPTER 106

NPTv6 Support

The NPTv6 feature supports translating IPv6 packet headers and source address prefixes in both directions, from inside to outside and vice versa. A router that implements an NPTv6 prefix translation function is referred to as an NPTv6 Translator.

To support inter-VRF communication, you can use VRF-Aware Software Infrastructure Scale feature. The VRF-Aware Software Infrastructure (VASI) Scale feature allows you to apply services such as access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls to MPLS traffic or IPv4 and IPv6 traffic that is flowing across two different Virtual Routing and Forwarding (VRF) instances. The VASI interfaces support redundancy of the Route Processor (RP) and Forwarding Processor (FP).

Information About NPTv6 support

The IPv6-to-IPv6 Network Prefix Translation (NPTv6) serves as a useful mechanism for implementing address independence in an IPv6 environment. A major benefit associated with NPTv6 is the fact that it avoids the requirement for an NPTv6 Translator to rewrite the transport layer headers which reduces the load on network devices. NPTv6 also does not interfere with encryption of the full IP payload.

The NPTv6 support allows for greater reliability as it provides support for load balancing and achieves the translation without breaking the end-to-end reachability at the network layer.

Interconnect Different Networks

The NPTv6 support allows you to redirect or forward packets from one network to another in an IPV6 environment. The NPTv6 support on is an algorithmic translation function which provides a 1:1 relationship between the addresses within the inside and outside network. When NPTv6 is used, you can interconnect different networks and support multihoming, load balancing, peer-to-peer networking.

Stateless Support

The NPTv6 does not create any state in the date plane and hence, can operate using minimal memory and supports High Availability (HA) by default.

Improved Support and Scaling

The NPTv6 supports prefix longer than 64 bits and supports static IPv6 host to host translations. You can configure IPv4 and IPv6 translations on the same interface using NPTv6 support and scaling is supported. The NPTv6 feature also supports packet tracing and conditional debugging.

Access to Services Hosted on a Global Network

Implementing VASI by using virtual interface pairs, where each of the interfaces in the pair is associated with a different VRF lets you access different services on the internet. The VASI virtual interface is the next hop interface for any packet that needs to be switched between these two VRFs. VASI interfaces provide the framework necessary to configure a firewall or a NAT between VRF instances.

Pairing of Interfaces

Each interface pair is associated with two different VRF instances. The two virtual interfaces, called vasileft and vasiright, in a pair are logically wired back-to-back and are completely symmetrical. Each interface has an index. The association of the pairing is done automatically based on the two interface indexes such that vasileft automatically gets paired to vasiright.

Static or Dynamic Routing

You can configure either static routing or dynamic routing with Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), or Open Shortest Path First (OSPF). BGP dynamic routing protocol restrictions and configuration are valid for BGP routing configurations between VASI interfaces.

Benefits of Using NPTv6 support

- When NPTv6 is used, you can interconnect different networks and support multihoming, load balancing,peer-to-peer networking. The NPTv6 does not create any state in the date plane and hence can operate using minimal memory and supports High Availability (HA) by default.
- You can configure IPv4 and IPv6 translations on the same interface using NPTv6 support and scaling is supported. The NPTv6 feature also supports Packet tracing and conditional debugging.

Restrictions for NPTv6 support

- Multicast is not supported.
- Firewall is not supported.
- High Speed Logging (HSL) and syslog is not supported..

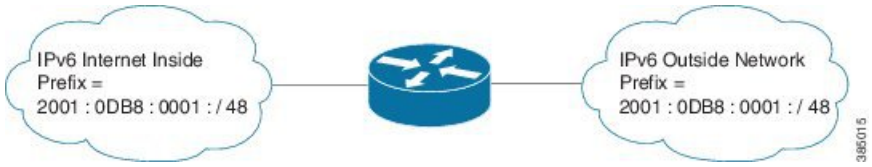
Deployment Scenarios for NPTv6 Support

Single Inside and Outside Network

You can use an NPTv6 Translator to interconnect two network links, one which is an internal network linked to a leaf network which is within a single administrative domain and the other which is external network with connectivity to a global network like the Internet. All hosts on the internal network use addresses from a single prefix which is routed locally. The addresses will be translated to and from the addresses in a globally routable prefix when the IP datagrams transit the NPTv6 Translator. The lengths of these two prefixes will be functionally the same and if the prefix lengths are different, the longer of the two prefixes limits the ability to use subnets in the shorter prefix.

The figure below illustrates NPTv6 deployment having a single inside and outside network.

Figure 98: NPTv6 using Single Inside and Outside Network

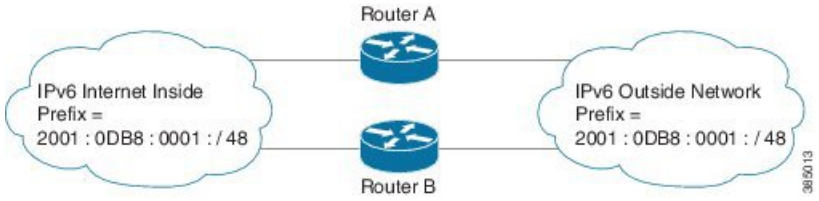


Redundancy and Load Sharing

When more than one NPTv6 Translator is attached to a network, the NPTv6 Translators are configured with the same internal and external prefixes. Since the translation is algorithmic, even though there are multiple translators, they map only one external address to the internal address.

The figure below illustrates NPTv6 deployment in redundancy and load-sharing network.

Figure 99: NPTv6 in Redundancy and Loadsharing Network

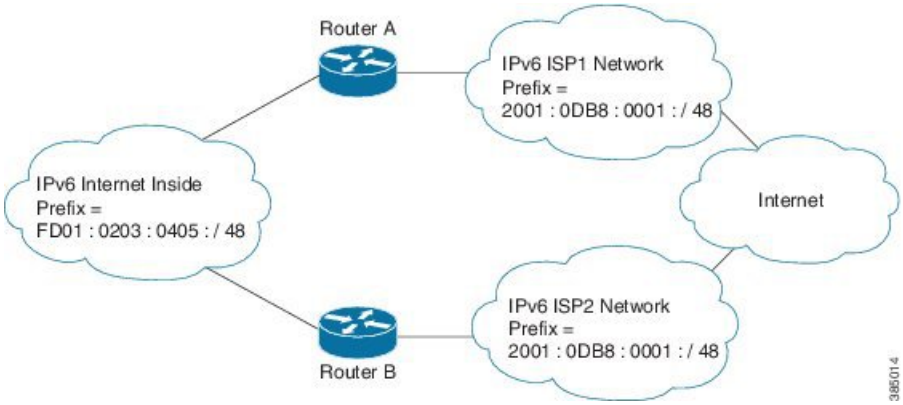


Multihoming

In a multihomed network the NPTv6 Translators are attached to an internal network, but are connected to different external networks. The NPTv6 Translators are configured with the same internal prefix but different external prefixes. Since there are multiple translations, the NPTv6 Translator maps multiple external addresses to the common internal address.

The figure below illustrates NPTv6 deployment in multihoming network.

Figure 100: NPTv6 in Multihoming Network



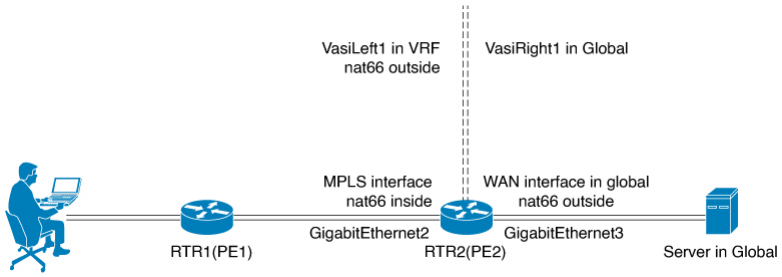
NPTv6 Support on VASI

VPN customers on 6vPE deployment could access services in global network like internet using NPTv6 translator on VASI interfaces (or by configuring NPTv6 on VASI interfaces). VASI allows applying NPTv6 translator to the traffic between VRFs/VPNs.

To support inter-VRF communication, you can use VRF-Aware Software Infrastructure Scale feature. The VRF-Aware Software Infrastructure (VASI) Scale feature allows you to apply services such as access control lists (ACLs), Network Address Translation (NAT), policing, and zone-based firewalls to MPLS traffic or IPv4 and IPv6 traffic that is flowing across two different Virtual Routing and Forwarding (VRF) instances. The VASI interfaces support redundancy of the Route Processor (RP) and Forwarding Processor (FP).

The figure below illustrates VPN customer in 6vPE deployment accessing services in global network using NPTv6 and VASI on PE2:

Figure 101: NPTv6 Support on VASI



Configuring NPTv6 Support on VASI

Configuring NPTv6 Support on VASI involves the following steps:

- Configure 6VPE for PE1
- Configure 6VPE for PE2
- Configure Virtual Interfaces on PE2
- Configure NPTv6 on PE2

Configure 6VPE for PE 1

To configure 6VPE for PE 1:

```
vrf definition client_vpn
rd 1:1
route-target export 1:1
route-target import 1:1
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
!
interface GigabitEthernet2
```

```
vrf forwarding client_vpn
ipv6 address 1001:1:2::2/64
!
interface GigabitEthernet3
ip address 10.2.0.2 255.255.255.0
!
interface Loopback 100
ip address 100.0.0.2 255.255.255.255
!
router ospf 1
network 100.0.0.2 0.0.0.0 area 0.1.0.0
network 10.2.0.2 0.0.0.0 area 0.1.0.0
!
interface GigabitEthernet3
ip ospf network point-to-point
!
mpls ldp router-id Loopback100 force
interface GigabitEthernet3
mpls ip
mpls label protocol ldp
!
router bgp 65002
bgp router-id 2.2.2.2
bgp log-neighbor-changes
neighbor 100.0.0.3 remote-as 65003
neighbor 100.0.0.3 ebgp-multihop 255
neighbor 100.0.0.3 update-source Loopback100
address-family ipv4
no neighbor 100.0.0.3 activate
exit-address-family
!
address-family vpnv6
neighbor 100.0.0.3 activate
neighbor 100.0.0.3 send-community both
exit-address-family
!
address-family ipv6 vrf client_vpn
redistribute connected
exit-address-family
!
```

Configure 6VPE for PE2

To Configure 6VPE for PE2:

```
ipv6 unicast-routing
vrf definition client_vpn
rd 1:1
route-target export 1:1
route-target import 1:1
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
!
interface GigabitEthernet2
ip address 10.2.0.3 255.255.255.0
!
interface GigabitEthernet3
ipv6 address 1001:3:4::3/64
!
```

```
interface Loopback 100
ip address 100.0.0.3 255.255.255.255
!
router ospf 1
network 100.0.0.3 0.0.0.0 area 0.1.0.0
network 10.2.0.3 0.0.0.0 area 0.1.0.0
!
interface GigabitEthernet2
ip ospf network point-to-point
!
mpls ldp router-id Loopback100 force
interface GigabitEthernet2
mpls ip
mpls label protocol ldp
!
router bgp 65003
bgp router-id 3.3.3.3
bgp log-neighbor-changes
neighbor 100.0.0.2 remote-as 65002
neighbor 100.0.0.2 ebgp-multihop 255
neighbor 100.0.0.2 update-source Loopback100
address-family ipv4
no neighbor 100.0.0.2 activate
exit-address-family
!
address-family vpnv6
neighbor 100.0.0.2 activate
neighbor 100.0.0.2 send-community both
exit-address-family
!
address-family ipv6 vrf client_vpn
exit-address-family
!
```

Configure Virtual Interfaces on PE2

To configure Virtual Interfaces on PE2:

```
interface vasileft1
vrf forwarding client_vpn
ipv6 address 1003:3:3::1/120
ipv6 address FE80:1:1:1::1 link-local
interface vasiright1
ipv6 address 1003:3:3::2/120
ipv6 address FE80:1:1:1::2 link-local
!
ipv6 prefix-list DENY_BGP_ROUTES_v6 deny 1001:1:2::/64
router bgp 65003
neighbor 1003:3:3::1 remote-as 60001
neighbor 1003:3:3::1 local-as 60002 no-prepend replace-as
neighbor 1003:3:3::1 description PEERING to the VASI left Interface
!
address-family ipv6
network 1001:3:4::/64
neighbor 1003:3:3::1 activate
neighbor 1003:3:3::1 send-community
neighbor 1003:3:3::1 next-hop-self
exit-address-family
!
address-family ipv6 vrf client_vpn
bgp router-id 5.5.5.5
neighbor 1003:3:3::2 remote-as 60002
neighbor 1003:3:3::2 local-as 60001 no-prepend replace-as
neighbor 1003:3:3::2 description Peer to VASI in Global
```

```
neighbor 1003:3:3::2 activate
neighbor 1003:3:3::2 send-community
neighbor 1003:3:3::2 prefix-list DENY_BGP_ROUTES_v6 out
exit-address-family
```

Configure NPTv6 on PE2

To configure NPTv6 on PE2:

```
interface GigabitEthernet2
nat66 inside
!
interface vasileft1
nat66 outside
!
interface GigabitEthernet3
nat66 outside
!
nat66 prefix inside 1001:1:2::/120 outside 2001:2001:2001::/120 vrf client_vpn
```

Verifying NPTv6 Configuration

To verify the various functions under the overall NPTv6 feature, refer to the below list of commands:

Command	Description

show nat66 prefix
Example:
`Device# show nat66 prefix`
`Prefixes configured: 1`
`NAT66 Prefixes`
`Id: 1 Inside 2002:AB01::/64 Outside 2002:AB02::/64`
	Verify stateless NAT66 prefix configuration.

show nat66 statistics
Example:
`Device# show nat66 statistics`
`NAT66 Statistics`
`Global Stats:`
`Packets translated (In -> Out)`
`: 7`
`Packets translated (Out -> In)`
`: 7`
	Verify NAT66 translation statistics.

show platform hardware qfp active feature nat66 datapath basecfg
Example:
`Device# show platform hardware qfp active feature`
`nat66 datapath basecfg`
`nat66 cfg_flags 0x00000001, dbg_flags 0x00000000`
`nat66_prefix_hash_table_entries 2048, nat66_prefix_hash_table 0x89628400`
`prefix hasht 0x89628400 max 2048 chunk 0x8c392bb0`
`hash_salt 719885386`
	Verify global stateless NPTv6 prefix in the data plane and other base configuration information.

show platform hardware qfp active feature nat66 datapath prefix
Example:
`Device# show platform hardware qfp active feature`
`nat66 datapath prefix`
`prefix hasht 0x89628400 max 2048 chunk 0x8c392bb0`
`hash_salt 719885386`
`NAT66 hash[1] id(1) len(64) vrf(0) in: 2002:ab01:0000:0000:0000:0000:0000:0000 out: 2002:ab02:0000:0000:0000:0000:0000:0000`
`in2out: 7`
`out2in: 7`
	Verify the stateless NPTv6 prefix configuration on passed interfaces.

show platform hardware qfp active feature nat66 datapath statistics
Example:
`Device# show platform hardware qfp act feat nat66`
`data statistics`
 	
`in2out xlated pkts 7 out2in xlated pkts 7`
`NAT66_DROP_SC_INVALID_PKT 0`
`NAT66_DROP_SC_BAD_DGLEN 0`
`NAT66_DROP_SC_PLU_FAIL 22786`
`NAT66_DROP_SC_PROCESS_V6_ERR 0`
`NAT66_DROP_SC_INVALID_EMBEDDED 0`
`NAT66_DROP_SC_SRC_RT 0`
`NAT66_DROP_SC_NOT_ENABLED 0`
`NAT66_DROP_SC_NO_GPM 0`
`NAT66_DROP_SC_LOOP 0`
`in2out_pkts 22768 out2in_pkts 22793`
`in2out_pkts_untrans 22761 out2in_pkts_untrans 22786`
`in2out_lookup_pass 7 out2in_lookup_pass 7`
`in2out_lookup_fail 0 out2in_lookup_fail 22786`
`mem_alloc_fail 0 prefix_fail 0`
`total prefix count 1`
	Verify global NPTv6 statistics.

Troubleshooting Tips

You must make sure that the inside and outside interfaces are configured.

Use the following debug commands if you have any configuration issues:

debug platform hardware qfp active feature nat66 datapath detailed	Provides detailed debugging information about the data plane layer.
debug platform hardware qfp active feature nat66 datapath all	Displays debugging information about the data plane layer.
debug platform condtion feature nat66 datapath submode detailed	Provides data plane layer debugging information using buginf_cond. ACL filter can be supplied via the debug condition infrastructure.

Additional References for NPTv6 support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IP Addressing Services commands	Cisco IOS IP Addressing Services Command Reference
VASI (VRF-Aware Software Infrastructure)	Configuring the VASI (VRF-Aware Software Infrastructure) Scale

Standards and RFCs

Standard/RFC	Title
RFC 6296	*IPv6-to-IPv6 Network Prefix Translation*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 107

NAT Stick Overview

The NAT Stick feature helps to route the packets back to the same input interface or another NAT stick interface. When a VM wants to communicate another VM in the same VRF (virtual router) and they are connected to the same physical router, NAT stick feature help the route back.

NAT helps when packet traverses from inside interface to outside interface and vice versa. NAT is required to configure both the interfaces (inside and outside).

Prerequisites for Configuring NAT Stick

Restrictions for Configuring NAT Stick

- ALGs are not supported on NAT stick.
- CGN mode is not supported on NAT stick
- Multicast packets are not supported on NAT stick.
- NAT stick does not support IPv6.
- Gate Keeper is not supported on NAT stick feature.
- Route maps are not supported on NAT stick. Only ACL will be supported.

Information About Configuring NAT Stick

Configuring NAT Stick

```
enable
configure terminal
interface GigabitEthernet2
 ip vrf forwarding vrf-30
```

```
 ip address 1.1.1.1 255.255.255.0
ip nat stick
end
```

Verifying NAT Stick Configuration

NAT Stick Configuration Example

CHAPTER 108

Initiating GARP for NAT Mapping

Restrictions

- The GARP retry feature provides customer support and minimizes control plane traffic impact, but only when utilized on BD-VIF interfaces.
- GARP for NAT mapping does not support the use of the same IP address in multiple VRFs within the same BD-VIF.
- GARP retry messages are sent only from the active box in a Box-to-Box High Availability (HA) configuration.

Information About Initiating GARP for NAT Mapping

Overview

Initiating Gratuitous Address Resolution Protocol (GARP) for Network Address Translation (NAT) Mapping is a feature that uses the Address Resolution Protocol (ARP) and GARP to map MAC addresses to IP addresses within a local network. This feature proactively updates and notifies devices about address changes in the network, ensuring accurate mapping between MAC and IP addresses.

By utilizing GARP, devices in the Application Centric Infrastructure (ACI) Fabric can efficiently discover and associate MAC addresses with IP addresses. This feature ensures that devices are promptly informed or notified about any changes in address assignments, allowing for seamless connectivity and efficient network operations.

Overall, GARP-based NAT Mapping in ACI Fabric simplifies the process of MAC-to-IP address resolution, enhances network efficiency, and ensures smooth communication between devices within the network.

Gratuitous ARP (GARP)

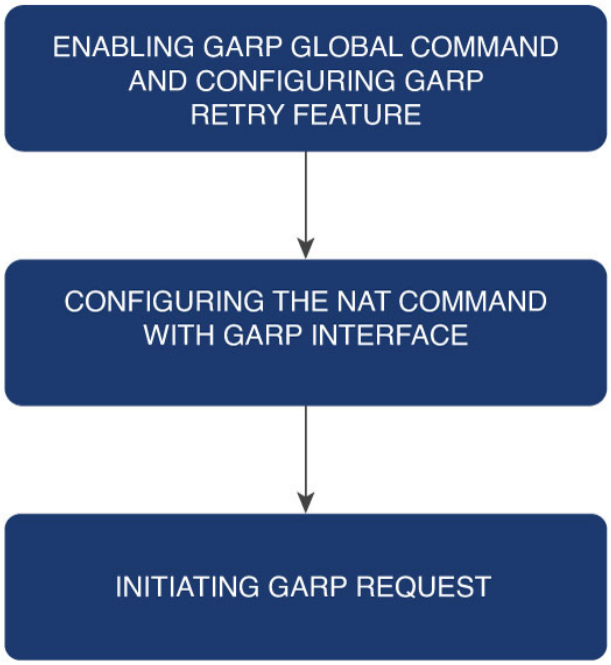
GARP is a part of the ARP that is primarily used to update host devices in a network about changes to IP to MAC address mappings. A network device sends a GARP request when the device's IP address changes, during failovers, or when the IP address initially becomes active on the network. This request ensures that other devices in the network update their ARP tables with the new mapping. In the context of NAT, GARP can be used to initiate or update NAT mappings across the network devices.

Initiating GARP for NAT Mapping in ACI Fabric

For a router connected to an ACI fabric in a cloud deployment, it takes a while to discover networks created due to changes in NAT mapping. With this functionality, whenever there is a change in NAT mapping, the router triggers a GARP message.

This message, in turn, enables the ACI fabric to discover the MAC address that corresponds to the IP address, thereby enabling seamless connectivity and efficient network operation.

Figure 102: Initiating GARP for NAT Mapping in ACI Fabric Process Flow



This feature is part of a broader configuration that brings together GARP and NAT mechanisms.

The process is initiated with the activation of the feature using the GARP global **ip arp nat-garp-retry feature enable** command. This command offers control over several optional parameters, including the number of NAT GARP retry messages, the interval between these messages, and the maximum number of GARP command executions.

Following this activation, the system requests GARP messages using the **garp-interface** option with the **ip nat inside source static** command. This happens on the BD-VIF interface during NAT mapping configuration.

How to Configure the Initiation of GARP for NAT Mapping

Configuring the Initiation of GARP for NAT Mapping

Perform the following steps to configure the initiation of GARP for NAT Mapping.

SUMMARY STEPS

1. **ip arp nat-garp-retry feature enable**
2. **ip arp nat-garp-retry retries** , **ip arp nat-garp-retry interval** , **ip arp nat-garp-retry entries**
3. **ip nat inside source static** {*local-ip*}{*global-ip*} **vrf** {*vrf-name*} **redundancy** {*redundancy-id*} **mapping-id** {*mapping-id*} **match-in-vrf garp-interface** {*interface-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip arp nat-garp-retry feature enable** **Example:** `Device(config)# ip arp nat-garp-retry feature enable`	Configure the GARP retry feature and the ARP retry database. Set the interval for ARP gratuitous retry. **Note** It is essential to enable GARP based NAT mapping at the system level.

Step 2	**ip arp nat-garp-retry retries** , **ip arp nat-garp-retry interval** , **ip arp nat-garp-retry entries**
	Example:
	`Device(config)# ip arp nat-garp-retry retries`
	`Device(config)# ip arp nat-garp-retry interval`
	`Device(config)# ip arp nat-garp-retry entries`
		These are optional arguments or keywords that provide further control over the **ip arp nat-garp-retry** command:
		• **retries:** Specifies the number of NAT GARP Retry messages. The default value is 2, and the permissible range is from 1 to 5. However, it is not recommended to set the value above 3.
		• **intervals:** Configures intervals between NAT GARP Retry messages. The default is 5 seconds, with a permissible range of 1 to 30 seconds
		• **entries:** Defines the number of NAT mappings to be supported. The maximum number of BD-VIF interfaces for GARP initiation is capped at 3000 to optimize control plane load.

Step 3	**ip nat inside source static** {*local-ip*}{*global-ip*} **vrf** {*vrf-name*} **redundancy** {*redundancy-id*} **mapping-id** {*mapping-id*} **match-in-vrf garp-interface** {*interface-name*} **Example:** `Device(config)# ip nat inside source static 192.168.1.1 203.0.113.1 vrf MYVRF redundancy 1 mapping-id 101 match-in-vrf garp-interface BD-VIF6000`	Configure NAT mapping with GARP requests.

Verifying NAT Mapping Configuration

To verify the initiation of GARP for NAT Mapping, use the **show running configuration | include garp-interface** command.

```
Router#sh running-config | inc garp-interface
ip nat inside source static 128.0.125.122 14.224.250.240 vrf ONE match-invrf redundancy 1
mapping-id 5555 garp-interface BD-VIF6000
```

This command will display the optional interface and indicate whether the GARP interface was configured successfully.

Configuration Examples for the Initiation of GARP for NAT Mapping

Configuring the Number of GARP Messages for Address Changes

```
active(config)#ip arp nat-garp-retry retries ?
<1-5> Specify the number of times an GARP is sent for static NAT
active(config)#
```

Configuring the Number of Intervals Between GARP Messages for Address Changes

```
active(config)#ip arp nat-garp-retry interval ?
<1-30> Specify the interval in seconds to send garp
active(config)#
```

Configuring the Number of Times to Initiate GARP

```
active(config)#ip arp nat-garp-retry entries ?
<1-3000> Specify the number of NAT static alias IP addresses to send garp
active(config)#
```

Configuring a GARP Request Using Static NAT Configuration

```
ip nat inside source static 10.180.137.182 155.20.1.112
vrf net50 redundancy 1 mapping-id 77 match-in-vrf garp-interface BD-VIF6000
```

PART IX

NHRP

CHAPTER 109

Configuring NHRP

The Next Hop Resolution Protocol (NHRP) is an Address Resolution Protocol (ARP)-like protocol that dynamically maps a Non-Broadcast Multi-Access (NBMA) network. With NHRP, systems attached to an NBMA network can dynamically learn the NBMA (physical) address of the other systems that are part of that network, allowing these systems to directly communicate.

NHRP is a client and server protocol where the hub is the Next Hop Server (NHS) and the spokes are the Next Hop Clients (NHCs). The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes to build direct tunnels.

Information About NHRP

How NHRP and NBMA Networks Interact

Most WAN networks are a collection of point-to-point links. Virtual tunnel networks (for example Generic Routing Encapsulation (GRE) tunnels) are also a collection of point-to-point links. To effectively scale the connectivity of these point-to-point links, they are usually grouped into a single or multilayer hub-and-spoke network. Multipoint interfaces (for example, GRE tunnel interfaces) can be used to reduce the configuration on a hub router in such a network. This resulting network is a Non-Broadcast Multi-Access (NBMA) network.

Because there are multiple tunnel endpoints reachable through the single multipoint interface, there needs to be a mapping from the logical tunnel endpoint IP address to the physical tunnel endpoint IP address in order to forward packets out the multipoint GRE (mGRE) tunnel interfaces over this NBMA network. This mapping could be statically configured, but it is preferable if the mapping can be discovered or learned dynamically.

NHRP is an ARP-like protocol that alleviates these NBMA network problems. With NHRP, systems attached to an NBMA network dynamically learn the NBMA address of the other systems that are part of that network, allowing these systems to directly communicate without requiring traffic to use an intermediate hop.

Routers, access servers, and hosts can use NHRP to discover the addresses of other routers and hosts connected to an NBMA network. Partially meshed NBMA networks typically have multiple logical networks behind

the NBMA network. In such configurations, packets traversing the NBMA network might have to make several hops over the NBMA network before arriving at the exit router (the router nearest the destination network). When NHRP is combined with IPsec, the NBMA network is basically a collection of point-to-point logical tunnel links over a physical IP network.

NHRP allows two functions to help support these NBMA networks:

1. NHRP Registration. NHRP allows Next Hop Clients (NHCs) to dynamically register with Next Hop Servers (NHSs). This registration function allows the NHCs to join the NBMA network without configuration changes on the NHSs, especially in cases where the NHC has a dynamic physical IP address or is behind a Network Address Translation (NAT) router that dynamically changes the physical IP address. In these cases, it would be impossible to preconfigure the logical virtual private network (VPN IP) to physical (NBMA IP) mapping for the NHC on the NHS. See the NHRP_Registration section for more information.
2. NHRP Resolution. NHRP allows one NHC (spoke) to dynamically discover the logical VPN IP to physical NBMA IP mapping for another NHC (spoke) within the same NBMA network. Without this discovery, IP packets traversing from hosts behind one spoke to hosts behind another spoke would have to traverse by way of the NHS (hub) router. This process would increase the utilization of the hub's physical bandwidth and CPU to process these packets that enter and exit the hub on the multipoint interface. With NHRP, systems attached to an NBMA network dynamically learn the NBMA address of the other systems that are part of that network, allowing these systems to directly communicate without requiring traffic to use an intermediate hop. This function alleviates the load on the intermediate hop (NHS) and can increase the overall bandwidth of the NBMA network to be greater than the bandwidth of the hub router.

Dynamically Built Hub-and-Spoke Networks

With NHRP, the NBMA network is initially laid out as a hub-and-spoke network that can be multiple hierarchical layers of NHCs as spokes and NHSs as hubs. The NHCs are configured with static mapping information to reach their NHSs and will connect to their NHS and send an NHRP registration to the NHS. This configuration allows the NHS to dynamically learn the mapping information for the spoke, reducing the configuration needed on the hub and allowing the spoke to obtain a dynamic NBMA (physical) IP address.

Once the base hub-and-spoke network is dynamically built, NHRP resolution requests and responses can be used to dynamically discover spoke-to-spoke mapping information, which allows spokes to bypass the hub and contact each other directly. This process allows a dynamic mesh of connections between spokes to be built based on data traffic patterns without requiring a preconfigured static fully meshed network. Using a dynamic-mesh network allows smaller spoke routers to participate up to their capability in a large NBMA network when these smaller spoke routers do not have the resources to participate in a full mesh on the same size network. The smaller spoke routers do not need to build out all possible spoke-to-spoke links; these routers need to build only the ones they are currently using.

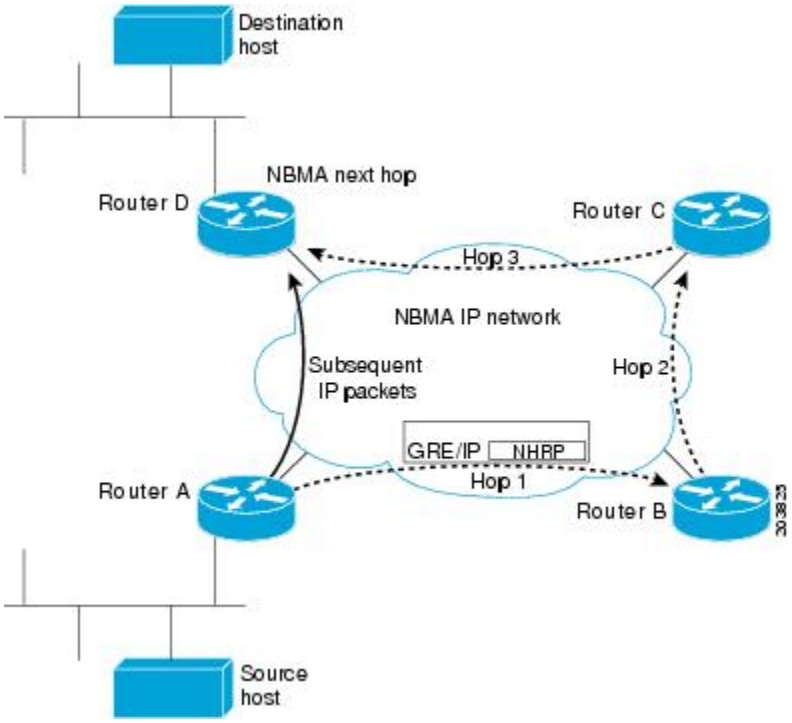
Next Hop Server Selection

NHRP resolution requests traverse one or more hops (hubs) within the base hub-and-spoke NBMA subnetwork before reaching the station that is expected to generate a response. Each station (including the source station) chooses a neighboring NHS to which it forwards the request. The NHS selection procedure typically involves performing a routing decision based upon the network layer destination address of the NHRP request. The NHRP resolution request eventually arrives at a station that generates an NHRP resolution reply. This responding station either serves the destination, or is the destination itself. The responding station generates a reply using the source address from within the NHRP packet to determine where the reply should be sent.

The Cisco implementation of NHRP also supports and extends the IETF RFC 2332, *NBMA Next Hop Resolution Protocol (NHRP)* .

The figure below illustrates four routers connected to an NBMA network. Within the network are IP routers necessary for the routers to communicate with each other by tunneling the IP data packets in GRE IP tunnel packets. The infrastructure layer routers support logical IP tunnel circuit connections represented by hops 1, 2, and 3. When router A attempts to forward an IP packet from the source host to the destination host, NHRP is triggered. On behalf of the source host, router A sends an NHRP resolution request packet encapsulated in a GRE IP packet, which takes three hops across the network to reach router D, connected to the destination host. After router A receives a positive NHRP resolution reply, router A determines that router D is the NBMA IP next hop, and router A sends subsequent data IP packets for the destination to router D in one GRE IP tunnel hop.

Figure 103: Next Hop Resolution Protocol



With NHRP, once the NBMA next hop is determined, the source either starts sending data packets to the destination (in a connectionless NBMA network such as GRE IP or SMDS) or establishes a virtual circuit (VC) connection to the destination. This connection is configured with the desired bandwidth and quality of service (QoS) characteristics for a connection-oriented NBMA network (such as Frame Relay or ATM) or with Dynamic Multipoint VPN (DMVPN) where an IPsec encryption peering must be established.

Other address resolution methods can be used while NHRP is deployed. IP hosts that rely upon the Logical IP Subnet (LIS) model might require ARP servers and services over the NBMA network, and deployed hosts might not implement NHRP, but might continue to support ARP variations. NHRP is designed to eliminate the suboptimal routing that results from the LIS model, and can be deployed with existing ARP services without interfering with them.

NHRP Registration

NHRP registrations are sent from NHCs to their configured NHSs every one-third of the NHRP holdtime (configured by the **ip nhrp holdtime** *value command*), unless the **ip nhrp registration timeout** *value* command is configured, in which case registrations are sent out according to the configured timeout value. If an NHRP registration reply is not received for an NHRP registration request, the NHRP registration request is retransmitted at timeouts of 1, 2, 4, 8, 16, and 32 seconds, then the sequence starts over again at 1.

The NHS is declared down if an NHRP registration reply is not received after three retransmission (7 seconds), and an NHRP resolution packets will no longer be sent to or by way of that NHS. NHRP registrations will continue to be sent at 1-, 2-, 4-, 8-, 16-, and 32-second intervals, probing the NHS until an NHRP registration reply is received. As soon as an NHRP registration reply is received the NHS is immediately declared up, the NHRP registration requests revert to being sent every one-third of NHRP holdtime or the value configured in the **ip nhrp registration timeout**command, and the NHS can again be sent NHRP resolution requests. The **show ip nhrp nhs detail** command can be used to check the state of the NHRP NHSs.

NHRP Used with a DMVPN

NHRP can be used to help build a VPN. In this context, a VPN consists of a virtual Layer 3 network that is built on top of an actual Layer 3 network. The topology you use over the VPN is largely independent of the underlying network, and the protocols you run over it are completely independent of it. The Dynamic Multipoint VPN (DMVPN) is based on GRE IP logical tunnels that can be protected by adding in IPsec to encrypt the GRE IP tunnels.

Dynamic Spoke-to-Spoke Tunnels

Spoke-to-spoke tunnels are designed to be dynamic, in that they are created only when there is data traffic to use the tunnel and they are removed when there is no longer any data traffic using the tunnel.

In addition to NHRP registration of NHCs with NHSs, NHRP provides the capability for NHCs (spokes) to find a shortcut path over the infrastructure of the network (IP network, SMDS) or build a shortcut switched virtual circuit (SVC) over a switched infrastructure network (Frame Relay and ATM) directly to another NHC (spoke), bypassing hops through the NHSs (hubs). This capability allows the building of very large NHRP NBMA networks. In this way, the bandwidth and CPU limitations of the hub do not limit the overall bandwidth of the NHRP NBMA network. This capability effectively creates a full-mesh-capable network without having to discover all possible connections beforehand. This type of network is called a dynamic-mesh network, where there is a base hub-and-spoke network of NHCs and NHSs for transporting NHRP and dynamic routing protocol information (and data traffic) and dynamic direct spoke-to-spoke links that are built when there is data traffic to use the link and torn down when the data traffic stops.

The dynamic-mesh network allows individual spoke routers to directly connect to anywhere in the NBMA network, even though they are capable of connecting only to a limited number at the same time. This functionality allows each spoke in the network to participate in the whole network up to its capabilities without limiting another spoke from participating up to its capability. If a full-mesh network were to be built, then all spokes would have to be sized to handle all possible tunnels at the same time.

For example, in a network of 1000 nodes, a full-mesh spoke would need to be large and powerful because it must always support 999 tunnels (one to every other node). In a dynamic-mesh network, a spoke needs to support only a limited number of tunnels to its NHSs (hubs) plus any currently active tunnels to other spokes. Also, if a spoke cannot build more spoke-to-spoke tunnels, then it will send its data traffic by way of the spoke-hub-spoke path. This design ensures that connectivity is always preserved, even when the preferred single hop path is not available.

Developmental Phases of DMVPN and NHRP

The developmental phases described in this section are actually DMVPN phases combining mGRE plus NHRP and IPsec. Phase 2 is important because it provides the functionality needed to support dynamic spoke-to-spoke tunnels.

- Phase 1 is the hub-and-spoke capability only. This phase will not be discussed here because phase 1 does not support spoke-to-spoke tunnels.
- Phase 2 adds spoke-to-spoke capability.

NHRP gathers the information that it needs to build spoke-to-spoke tunnels by using NHRP resolution request and reply packets that are sent via the spoke-hub-spoke path through the NBMA network. NHRP also has to be triggered (or know when) to collect this information for building the spoke-to-spoke tunnels, because it brings up the spoke-to-spoke tunnel only when there is data traffic to use it. The two ways that NHRP does this are described the following sections.

NHRP gathers the information that it needs to build spoke-to-spoke tunnels by using NHRP resolution request and reply packets that are sent via the spoke-hub-spoke path through the NBMA network. NHRP also has to be triggered (or know when) to collect this information for building the spoke-to-spoke tunnels, because it brings up the spoke-to-spoke tunnel only when there is data traffic to use it.

The IP routing table and the routes learned by way of the hub are important when building spoke-to-spoke tunnels. Therefore, the availability of the NHSs (hubs) is critical for the functioning of an NHRP-based network. When there is only one hub and that hub goes down, the spoke removes the routes that it learned from the hub from its routing table, because it lost the hub as its routing neighbor. However, the spoke does not delete any of the spoke-to-spoke tunnels (NHRP mappings) that are now up. Even though the spoke-to-spoke tunnel is still there the spoke will not be able to use the tunnel because its routing table no longer has a route to the destination network. The spoke has a path (spoke-to-spoke tunnel), but does not know to use it (because there is no routing table entry).

In addition, when the routing entries are removed there is no trigger into NHRP for NHRP to remove NHRP mapping entries. Eventually NHRP will time out the current dynamic NHRP mapping entries that it had when the hub went down because they are not being used. Only at that time does NHRP remove the mapping entry.

In phase 2, if there still happened to be a route in the routing table (could be a static route) with the correct IP next hop, then the spoke could still use the spoke-to-spoke tunnel even when the hub is down. NHRP will not be able to refresh the mapping entry because the NHRP resolution request or response would need to go through the hub.

If you have two (or more) NHS hubs within a single NBMA network (single mGRE, Frame Relay, or ATM interface), then when the first (primary) hub goes down, the spoke router will still remove the routes from the routing table that it learned from this hub, but it will also be learning the same routes (higher metric) from the second (backup) hub, so it will immediately install these routes. Therefore the spoke-to-spoke traffic would continue going over the spoke-to-spoke tunnel and be unaffected by the primary hub outage.

In phase 2, NHRP brings up the NHC-to-NHS tunnel and a dynamic routing protocol is used to distribute routing information about all of the networks that are available behind the hub and all of the other spokes. Included in this information is the IP next hop of the destination spoke that is supporting a particular destination network.

When a data packet is forwarded, it obtains the outbound interface and the IP next hop from the matching routing table network entry. If the NHRP interface is the outbound interface, it looks for an NHRP mapping entry for that IP next hop. If there is no matching of an NHRP mapping entry, then NHRP is triggered to send an NHRP resolution request to get the mapping information (IP next-hop address to physical layer address). The NHRP registration reply packet contains this mapping information. When this information is received,

the spoke has enough information to correctly encapsulate the data packet to go directly to the remote spoke, taking one hop across the infrastructure network. One of the disadvantages to this technique is that each spoke must have all of the individual routes in its routing table for all possible destination networks behind the hub and other spokes. Keeping this routing information distributed and up to date can put a significant load on the routing protocol running over the VPN.

Spoke Refresh Mechanism for Spoke-to-Spoke Tunnels

Spoke-to-spoke tunnels are designed to be dynamic, in that they are created only when there is data traffic to use the tunnel and they are removed when there is no longer any data traffic using the tunnel. This section describes the mechanism to refresh the spoke-to-spoke tunnel when it is still being used (no packet loss) and to detect and remove the spoke-to-spoke tunnel when it is no longer being used.

Process Switching

Each time a data packet is switched using an NHRP mapping entry, the “used” flag is set on the mapping entry. Then when the NHRP background process runs (every 60 seconds) the following actions occur:

- If the expire time is >135 seconds and the “used” flag is set, then the “used” flag is cleared.
- If the expire time is <= 135 seconds and the “used” flag is set, then the entry is refreshed.
- If the expire time is <= 135 seconds and the “used” flag is not set, then nothing is done.

CEF Switching

NHRP has no knowledge about when a packet is Cisco Express Forwarding (CEF) switched through the spoke-to-spoke tunnel.

When the NHRP background process runs, the following actions occur:

- If the expire time is > 135 seconds, then nothing is done.
- If the expire time is <= 135 seconds, then the corresponding CEF adjacency is marked “stale”. If the CEF adjacency is then used to switch a packet, CEF will mark the adjacency “fresh” and trigger NHRP to refresh the mapping entry.

In both the process and CEF switching cases, refreshed means that another NHRP resolution request is sent and response is needed to keep the entry from expiring. If the expiration time goes to 0 then the NHRP mapping entry is deleted. Also, if this entry is the last mapping entry with this NBMA address and if the router is CEF switching, then the CEF adjacency will be cleared and marked incomplete.

If the IPsec **tunnel protection ipsec profile** *name* command is used on an NHRP mGRE interface, then the following actions also occur:

1. The corresponding crypto socket entry is deleted.
2. The corresponding crypto map entry is deleted.
3. The corresponding IPsec security associations (SAs) and Internet Security Association and Key Management Protocol (ISAKMP) SAs are deleted.
4. Just prior to removing the ISAKMP SA, phase 2 and phase 1 delete notify messages are sent to the ISAKMP peer.
5. The ISAKMP peer deletes the corresponding IPsec SAs and ISAKMP SAs.

6. Via the crypto socket, the ISAKMP peer's NHRP mapping entry sets its expire time set to 5 seconds, unless it is a static NHRP mapping entry.
7. When the NHRP mapping entry expires and if it is the last mapping entry with this NBMA address, then the ISAKMP peer also performs items 1 through 5.

How to Configure NHRP

Configuring a GRE Tunnel for Multipoint Operation

Perform this task to configure a GRE tunnel for multipoint (NMBA) operation.

You can enable a GRE tunnel to operate in multipoint fashion. A tunnel network of multipoint tunnel interfaces can be thought of as an NBMA network. When multiple GRE tunnels are configured on the same router, they must either have unique tunnel ID keys or unique tunnel source addresses. NHRP is required on mGRE tunnel interfaces because it provides the VPN-layer-IP to NBMA-layer-IP address mappings for forwarding IP data packets over the mGRE tunnel.

IThe tunnel ID key is carried in each GRE packet, it is not carried in any NHRP messages. We do not recommend relying on this key for security purposes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel mode gre multipoint**
5. **tunnel key** *key-number*
6. **ip nhrp network-id** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**tunnel mode gre multipoint** **Example:** `Router(config-if)# tunnel mode gre multipoint`	Enables a GRE tunnel to be used in multipoint NBMA mode.
Step 5	**tunnel key** *key-number* **Example:** `Router(config-if)# tunnel key 3`	(Optional) Sets the tunnel ID key
Step 6	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 1`	Enables NHRP on the interface.

Enabling NHRP on an Interface

Perform this task to enable NHRP for an interface on a router. In general, all NHRP stations within a logical NBMA network should be configured with the same network identifier.

The NHRP network ID is used to define the NHRP domain for an NHRP interface and differentiate between multiple NHRP domains or networks, when two or more NHRP domains (GRE tunnel interfaces) are available on the same NHRP node (router). The NHRP network ID is used to help keep two NHRP networks (clouds) separate from each other when both are configured on the same router.

The NHRP network ID is a local only parameter. It is significant only to the local router and is not transmitted in NHRP packets to other NHRP nodes. For this reason the actual value of the NHRP network ID configured on a router need not match the same NHRP network ID on another router where both of these routers are in the same NHRP domain. As NHRP packets arrive on a GRE interface, they are assigned to the local NHRP domain in the NHRP network ID that is configured on that interface.

Note This method of assigning a network ID is similar to the Open Shortest Path First (OSPF) concept of process ID in the **router ospf** *process- id*command. If more than one OSPF process is configured, then the OSPF neighbors and any routing data that they provide is assigned to the OSPF process (domain) by which interfaces map to the *network* arguments under the different **router ospf** *process-id* configuration blocks.

We recommend that the same NHRP network ID be used on the GRE interfaces on all routers that are in the same NHRP network. It is then easier to track which GRE interfaces are members of which NHRP network.

NHRP domains (network IDs) can be unique on each GRE tunnel interface on a router. This is required when running DMVPN phase 1 or phase 2 or when using a tunnel key on the GRE interfaces. These unique IDs place each GRE interface into a different NHRP domain, which is equivalent to each being in a unique DMVPN.

NHRP domains can span across GRE tunnel interfaces on a route. This option is available when running DMVPN phase 3 and not using a tunnel key on the GRE tunnel interfaces. In this case the effect of using the same NHRP network ID on the GRE tunnel interfaces is to merge the two GRE interfaces into a single NHRP network (DMVPN network).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address network-mask*
5. **ip nhrp network-id** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address network-mask* **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Enables IP and gives the interface an IP address.
Step 5	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 1`	Enables NHRP on the interface.
Step 6	**end** **Example:** `Router(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Static IP-to-NBMA Address Mapping on a Station

Perform this task to configure static IP-to-NBMA address mapping on a station (host or router). To enable IP multicast and broadcast packets to be sent to the statically configured station, use the **ip nhrp map multicast**

*nbma-address*command. This command is required on multipoint GRE tunnels and not required on point-point RE tunnels.

To participate in NHRP, a station connected to an NBMA network must be configured with the IP and NBMA addresses of its NHSs. The format of the NBMA address depends on the medium you are using. For example, GRE uses a network service access point (NSAP) address, Ethernet uses a MAC address, and SMDS uses an E.164 address.

These NHSs may also be the default or peer routers of the station, so their addresses can be obtained from the network layer forwarding table of the station.

If the station is attached to several link layer networks (including logical NBMA networks), the station should also be configured to receive routing information from its NHSs and peer routers so that it can determine which IP networks are reachable through which link layer networks.

Perform this task to configure static IP-to-NBMA address mapping on a station (host or router). To enable IP multicast and broadcast packets to be sent to the statically configured station, use the **ip nhrp map multicast** *nbma-address*command. This step is required on multipoint GRE tunnels and not required on point-point RE tunnels.

Note The IGP routing protocol uses IP multicast or broadcast, so the **ip nhrp map multicast** command, though optional, is often required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp map** *ip-address nbma-address*
5. **ip nhrp map multicast** *nbma-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip nhrp map** *ip-address nbma-address* **Example:** `Router(config-if)# ip nhrp map 10.0.0.2 172.16.1.2`	Configures static IP-to-NBMA address mapping on the station.
Step 5	**ip nhrp map multicast** *nbma-address* **Example:** `Router(config-if)# ip nhrp map multicast 172.16.1.12`	(Optional) Adds an NBMA address to receive multicast or broadcast packets sent out the interface. **Note** This command is not required on point-to-point GRE tunnels.

Statically Configuring a Next Hop Server

Perform this task to statically configure a Next Hop Server.

An NHS normally uses the network layer forwarding table to determine where to forward NHRP packets and to find the egress point from an NBMA network. An NHS may also be statically configured with a set of IP address prefixes that correspond to the IP addresses of the stations it serves, and their logical NBMA network identifiers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp nhs** *nhs-address* [*net-address* [*netmask*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp nhs** *nhs-address* [*net-address* [*netmask*]]	Statically configures a Next Hop Server.

	Command or Action	Purpose
	Example: `Router(config-if)# ip nhrp nhs 10.0.0.2`	• To configure multiple networks that the Next Hop Server serves, repeat the **ip nhrp nhs** command with the same Next Hop Server address, but different IP network addresses. • To configure additional Next Hop Servers, repeat the **ip nhrp nhs** command.

Changing the Length of Time NBMA Addresses Are Advertised as Valid

Perform this task to change the length of time that NBMA addresses are advertised as valid in positive NHRP responses. In this context, *advertised* means how long the Cisco IOS XE software tells other routers to keep the address mappings it is providing in NHRP responses. The default length of time is 7200 seconds (2 hours).

This configuration controls how long a spoke-to-spoke shortcut path will stay up after it is no longer used or how often the spoke-to-spoke short-cut path mapping entry will be refreshed if it is still being used. We recommend that a value from 300 to 600 seconds be used.

The **ip nhrp holdtime**command controls how often the NHRP NHC will send NHRP registration requests to its configured NHRP NHSs. Effective with Cisco IOS XE 16.2.1 Release, the default value to send NHRP registrations is every two-third the NHRP holdtime value (default = 600 seconds (10 minutes)).

Note For the devices prior to Cisco IOS XE 16.2.1 Release, the NHRP default holdtime is 2400 seconds.

The optional **ip nhrp registration timeout** *value* command can be used to set the interval for sending NHRP registration requests independently from the NHRP holdtime.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp holdtime** *seconds*
5. **ip nhrp registration timeout** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp holdtime** *seconds* **Example:** `Router(config-if)# ip nhrp holdtime 600`	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in positive NHRP responses. • In this example, NHRP NBMA addresses are advertised as valid in positive NHRP responses for 10 minutes. **Note** The recommended NHRP hold time value ranges from 300 to 600 seconds. Although a higher value can be used when required, we recommend that you do not use a value less than 300 seconds, and if used, it should be used with extreme caution.
Step 5	**ip nhrp registration timeout** *seconds* **Example:** `Router(config-if)# ip nhrp registration timeout 100`	(Optional) Changes the interval that NHRP NHCs send NHRP registration requests to configured NHRP NHSs. • In this example, NHRP registration requests are now sent every 100 seconds (default value is one third NHRP holdtime value).

Specifying the NHRP Authentication String

Perform this task to specify the authentication string for NHRP on an interface.

Configuring an authentication string ensures that only routers configured with the same string can communicate using NHRP. Therefore, if the authentication scheme is to be used, the same string must be configured in all devices configured for NHRP on a fabric.

Note We recommend using an NHRP authentication string, especially to help keep multiple NHRP domains separate from each other. The NHRP authentication string is not encrypted, so it cannot be used as a true authentication for an NHRP node trying to enter the NHRP network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp authentication** *string*

5. **exit**
6. **show ip nhrp** [**dynamic** | **static**] [*type number*]
7. **show ip nhrp traffic**
8. **show ip nhrp nhs** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication specialxx`	Specifies an authentication string. • All routers configured with NHRP within one logical NBMA network must share the same authentication string.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nhrp** [**dynamic**	**static**] [*type number*] **Example:** `Router# show ip nhrp`	Displays the IP NHRP cache, which can be limited to dynamic or static cache entries for a specific interface.
Step 7	**show ip nhrp traffic** **Example:** `Router# show ip nhrp traffic`	Displays NHRP traffic statistics.
Step 8	**show ip nhrp nhs** [**detail**] **Example:** `Router# show ip nhrp nhs detail`	Displays NHRP holdtime details.

Configuring NHRP Server-Only Mode

Perform this task to configure NHRP server-only mode.

You can configure an interface so that it will not initiate or respond to an attempt to establish an NHRP shortcut SVCs. Configure NHRP server-only mode on routers that you do not want building NHRP shortcut SVCs.

Configuring the router in NHRP server-only mode stops a router from initiating NHRP resolution requests and also from responding to an NHRP resolution request for any prefix where this router is the exit point from the NBMA network for the prefix in the request. However, this will not stop the router from forwarding NHRP resolution requests and responses that would be or have been answered by other nodes.

If an interface is placed in NHRP server-only mode, you have the option to specify the **ip nhrp server-only** [**non-caching**] command keyword. In this case, NHRP does not store mapping information in the NHRP cache, such as NHRP responses that go through the router. To save memory and block building of NHRP shortcuts, the non-caching option is generally used on a router located between two other NHRP routers (NHRP hubs).

Perform this task to configure NHRP server-only mode.

Note When the **ip nhrp server-only** command is applied on Cisco ASR 1000 Series Aggregation Services Routers, any data IP packets that are being forwarded out of the tunnel interface to a destination IP that does not have a current NHRP mapping for the next-hop IP address, are dropped. For this reason, it is recommend that the **ip nhrp server-only** command is configured on Cisco ASR 1000 Series Aggregation Services Routers only if the router is used as a hub node (NHS) in the NBMA network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp server-only** [**non-caching**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface tunnel 100`	
Step 4	**ip nhrp server-only** [**non-caching**] **Example:** `Router(config-if)# ip nhrp server-only non-caching`	Configures NHRP server-only mode.

Controlling the Triggering of NHRP

There are two ways to control when NHRP is triggered on any platform. These methods are described in the following sections:

Triggering NHRP on a Per-Destination Basis

Perform the following task to trigger NHRP on a per-destination basis.

You can specify an IP access list that is used to decide which IP packets can trigger the sending of NHRP resolution requests. By default, all non-NHRP packets trigger NHRP resolution requests. To limit which IP packets trigger NHRP resolution requests, define an access list and then apply it to the interface.

Note NHRP resolution requests are used to build direct paths between two NHRP nodes. Even though certain traffic is excluded from triggering the building of this path, if the path is already built then this "excluded" traffic will use the direct path.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **access-list** *access-list-number* {**deny** | **permit**} *source*[*source-wildcard*]
   - **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]
4. **interface** *type number*
5. **ip nhrp interest** *access-list-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **access-list** *access-list-number* {**deny** | **permit**} *source*[*source-wildcard*]
	• **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]
	Example:
	`Router(config)# access-list 101 permit ip any any`
	Example:
	`Router(config)# access-list 101 deny ip any 10.3.0.0 0.0.255.255`
		Defines a standard or extended IP access list.

Step 4	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 5	**ip nhrp interest** *access-list-number* **Example:** `Router(config-if)# ip nhrp interest 101`	Specifies an IP access list that controls NHRP requests. • In this example, only the packets that pass extended access list 101 are subject to the default SVC triggering and teardown rates.

Triggering NHRP on a Packet Count Basis

By default, when the software attempts to send a data packet to a destination for which it has determined that NHRP can be used, it sends an NHRP request for that destination. Perform this task to configure the system to wait until a specified number of data packets have been sent to a particular destination before NHRP is attempted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp use** *usage-count*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp use** *usage-count* **Example:** `Router(config-if)# ip nhrp use 5`	Specifies how many data packets are sent to a destination before NHRP is attempted. • In this example, if in the first minute five packets are sent to the first destination and five packets are sent to a second destination, then a single NHRP request is generated for the second destination. • If in the second minute the same traffic is generated and no NHRP responses have been received, then the system resends its request for the second destination.

Triggering NHRP Based on Traffic Thresholds

NHRP can run on Cisco Express Forwarding platforms when NHRP runs with Border Gateway Protocol (BGP). You can configure NHRP to initiate SVCs once a configured traffic rate is reached. Similarly, SVCs can be torn down when traffic falls to another configured rate.

You can configure the traffic rate that must be reached before NHRP sets up or tears down an SVC. Because SVCs are created only for burst traffic, you can conserve resources.

To configure the NHRP triggering and teardown of SVCs based on traffic rate, perform the following tasks. The first task is required; the second and third tasks are optional.

Changing the Rate for Triggering SVCs

Perform this task to change the number of kilobits per second (kbps) at which NHRP sets up or tears down the SVC to this destination.

When NHRP runs with BGP, there is a way to control the triggering of NHRP packets. This method consists of SVCs being initiated based on the input traffic rate to a given BGP next hop.

When BGP discovers a BGP next hop and enters this BGP route into the routing table, an NHRP request is sent to the BGP next hop. When an NHRP reply is received, a subsequent route is put in the NHRP cache that directly corresponds to the BGP next hop.

A new NHRP request is sent to the same BGP next hop to repopulate the NHRP cache. When an NHRP cache entry is generated, a subsequent map statement to the same BGP next hop is also created.

Aggregate traffic to each BGP next hop is measured and monitored. Once the aggregate traffic has met or exceeded the configured trigger rate, NHRP creates an SVC and sends traffic directly to that destination router. The router tears down the SVC to the specified destinations when the aggregate traffic rate falls to or below the configured teardown rate.

By default, NHRP will set up an SVC for a destination when aggregate traffic for that destination is more than 1 kbps over a running average of 30 seconds. Similarly, NHRP will tear down the SVC when the traffic for that destination drops to 0 kbps over a running average of 30 seconds. There are several ways to change the rate at which SVC setup or teardown occurs. You can change the number of kbps thresholds, or the load interval, or both.

Before you begin

Before you configure the feature whereby NHRP initiation is based on traffic rate, the following conditions must exist in the router:

- GRE must be configured.
- CEF switching or distributed CEF (dCEF) switching must be enabled.
- BGP must be configured on all routers in the network where these enhancements are running.

If your network has CEF switching or dCEF switching and you want NHRP to work (whether with default values or changed values), configure the **ip cef accounting non-recursive**command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp trigger-svc** *trigger-threshold teardown-threshold*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp trigger-svc** *trigger-threshold teardown-threshold* **Example:** `Router(config-if)# ip nhrp trigger-svc 100 5`	Changes the rate at which NHRP sets up or tears down SVCs. • In this example, the triggering and teardown thresholds are set to 100 kbps and 5 kbps, respectively.

Changing the Sampling Time Period and Sampling Rate

You can change the length of time over which the average trigger rate or teardown rate is calculated. By default, the period is 30 seconds; the range is from 30 to 300 seconds in 30-second increments. This period is for calculations of aggregate traffic rate internal to Cisco IOS XE software only, and it represents a worst-case time period for taking action. In some cases, the software will act sooner, depending on the ramp-up and fall-off rate of the traffic.

If your Cisco hardware has a Virtual Interface Processor, version 2 adapter, you must perform this task to change the sampling time. By default, the port adapter sends the traffic statistics to the Route Processor every 10 seconds. If you are using NHRP in dCEF switching mode, you must change this update rate to 5 seconds.

Perform this task to change the sampling time period and the sampling rate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef traffic-statistics** [**load-interval** *seconds*]
4. **ip cef traffic-statistics** [**update-rate** *seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef traffic-statistics** [**load-interval** *seconds*] **Example:**	Changes the length of time in a sampling period during which trigger and teardown thresholds are averaged.

	Command or Action	Purpose
	`Router(config)# ip cef traffic-statistics load-interval 120`	• In this example, the triggering and teardown thresholds are calculated based on an average over 120 seconds.
Step 4	**ip cef traffic-statistics** [**update-rate** *seconds*] **Example:** `Router(config)# ip cef traffic-statistics update-rate 5`	Specifies the frequency that the port adapter sends the accounting statistics to the RP. • When using NHRP in distributed CEF switching mode, this value must be set to 5 seconds. The default value is 10 seconds.

Applying the Triggering and Teardown Rates to Specific Destinations

Perform this task to impose the triggering and teardown rates on certain destinations. By default, all destinations are measured and monitored for NHRP triggering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **access-list** *access-list-number* {**deny** | **permit**} *source*[*source-wildcard*]
   - **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**log**]
4. **interface** *type number*
5. **ip nhrp interest** *access-list-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **access-list** *access-list-number* {**deny** | **permit**} *source*[*source-wildcard*] • **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**log**]	Defines a standard or extended IP access list. • In the example an extended access list is defined.

	Command or Action	Purpose

	Example:
	`Router(config)# access-list 101 permit ip any any`
	Example:
	`Router(config)# access-list 101 deny ip any 10.3.0.0 0.0.255.255`	

Step 4	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 5	**ip nhrp interest** *access-list-number* **Example:** `Router(config-if)# ip nhrp interest 101`	Specifies an IP access list that controls NHRP requests. • In this example, only the packets that pass extended access list 101 are subject to the default SVC triggering and teardown rates.

Controlling the NHRP Packet Rate

Perform this task to change the maximum rate at which NRHP packets will be handled.

There is the maximum value (max-send interval) for the number of NHRP messages that the local NHRP process can handle within a set period of time. This limit protects the router against events like a runaway NHRP process sending NHRP requests or an application (worm) that is doing an IP address scan that is triggering many spoke-to-spoke tunnels.

The larger the max-send interval the more NHRP packets the system can process and send. These messages do not use much memory and the CPU usage is not very large per message; however, excessive messages causing excessive CPU usage can degrade system performance.

To set a reasonable max-send-interval, consider the following information:

- Number of spoke routers being handled by this hub and how often they send NHRP registration requests. To support this load you would need:

Number of spokes/registration timeout * max-send interval

For example, 500 spokes with a 100-second registration timeout would equate as follows:

max-send interval = 500/100*10 = 50

- The maximum number of spoke-to-spoke tunnels that are expected to be up at any one time across the NBMA network:

spoke-to-spoke tunnels/NHRP holdtime * max-send interval

This would cover spoke-to-spoke tunnel creation and the refreshing of spoke-to-spoke tunnels that are used for longer periods of time.

Then add these values together and multiply the result by 1.5 or 2.0 to give a buffer.

- The max-send interval can be used to keep the long-term average number of NHRP messages allowed to be sent constant, but allow greater peaks.

By default, the maximum rate at which the software sends NHRP packets is five packets per 10 seconds. The software maintains a per-interface quota of NHRP packets (whether generated locally or forwarded) that can be sent.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp max-send** *pkt-count* **every** *interval*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**ip nhrp max-send** *pkt-count* **every** *interval* **Example:** `Router(config-if)# ip nhrp max-send 10 every 10`	In this example, ten NHRP packets can be sent from the interface every 10 seconds (twice the default rate).

Suppressing Forward and Reverse Record Options

To dynamically detect link layer filtering in NBMA networks (for example, SMDS address screens), and to provide loop detection and diagnostic capabilities, NHRP incorporates a Route Record in request and reply packets. The Route Record options contain the network (and link layer) addresses of all intermediate Next Hop Servers between the source and destination (in the forward direction) and between the destination and source (in the reverse direction).

By default, Forward Record options and Reverse Record options are included in NHRP request and reply packets. Perform this task to suppress forward and reverse record options.

Note Forward and Reverse Record information is required for the proper operation of NHRP, especially in a DMVPN network. Therefore you must not configure suppression of this information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip nhrp record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 100`	Configures an interface and enters interface configuration mode.
Step 4	**no ip nhrp record** **Example:** `Router(config-if)# no ip nhrp record`	Suppresses Forward and Reverse Record options.

Specifying the NHRP Responder IP Address

An NHRP requester that wants to know which Next Hop Server generates an NHRP reply packet can include the responder address option in its NHRP request packet. The Next Hop Server that generates the NHRP reply packet then complies by inserting its own IP address in the NHRP reply. The Next Hop Server uses the primary IP address of the specified interface.

Perform this task to specify which interface the Next Hop Server uses for the NHRP responder IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **ip nhrp responder** *type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface serial 0`	Configures a serial interface and enters interface configuration mode.
Step 4	**ip nhrp responder** *type number* **Example:** `Router(config-if)# ip nhrp responder serial 0`	Specifies which interface the Next Hop Server uses for the NHRP responder IP address. • In this example, any NHRP requests for the Responder Address will cause this router acting as a next-hop server to supply the primary IP address of serial interface 0 in the NHRP reply packet. • If an NHRP reply packet being forwarded by a Next Hop Server contains the IP address of that server, the Next Hop Server generates an error indication of type "NHRP Loop Detected" and discards the reply.

Clearing the NHRP Cache

The NHRP cache can contain entries of statically configured NHRP mappings and dynamic entries caused by the Cisco IOS XE software learning addresses from NHRP packets. To clear statically configured entries, use the **no ip nhrp map** command in interface configuration mode.

Perform the following task to clear the NHRP cache.

SUMMARY STEPS

1. **enable**
2. **clear ip nhrp** [*ip-address*] [*ip-mask*]

DETAILED STEPS

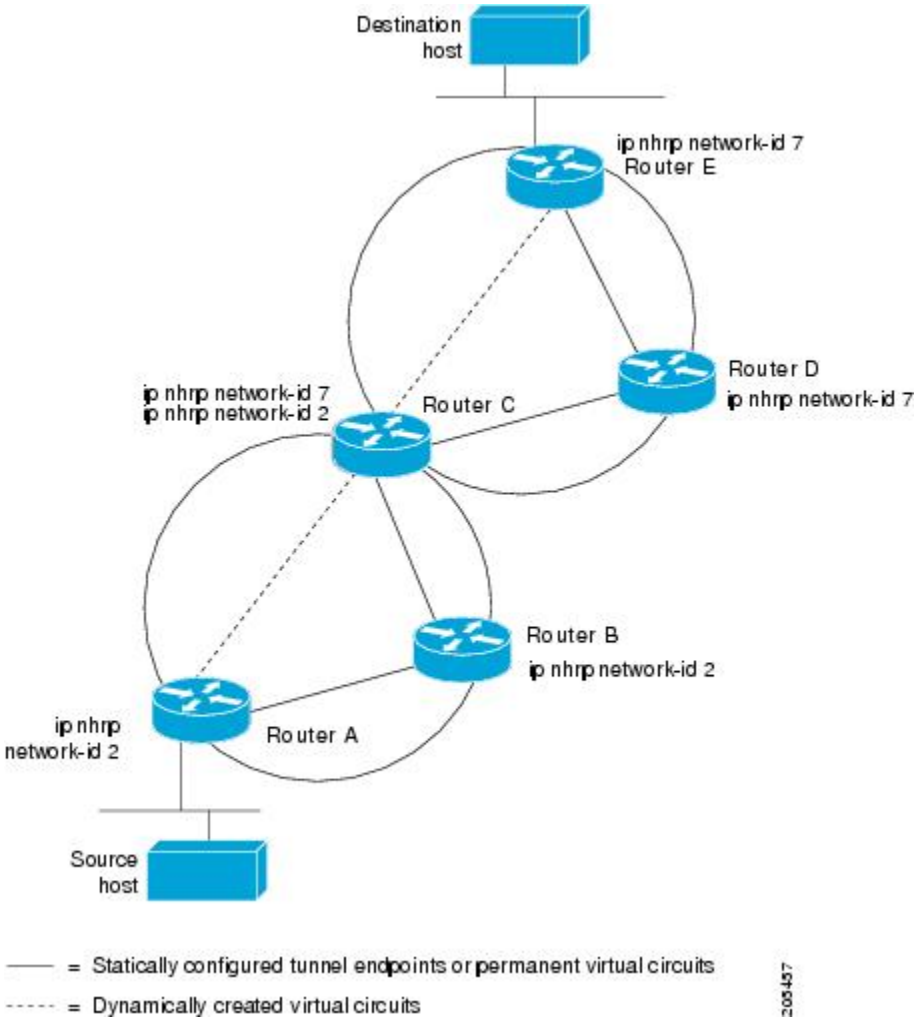
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip nhrp** [*ip-address*] [*ip-mask*] **Example:** `Router# clear ip nhrp`	Clears the IP NHRP cache of dynamic entries. • This command does not clear any static (configured) IP to NBMA address mappings from the NHRP cache.

Configuration Examples for NHRP

Physical Network Designs for Logical NBMA Examples

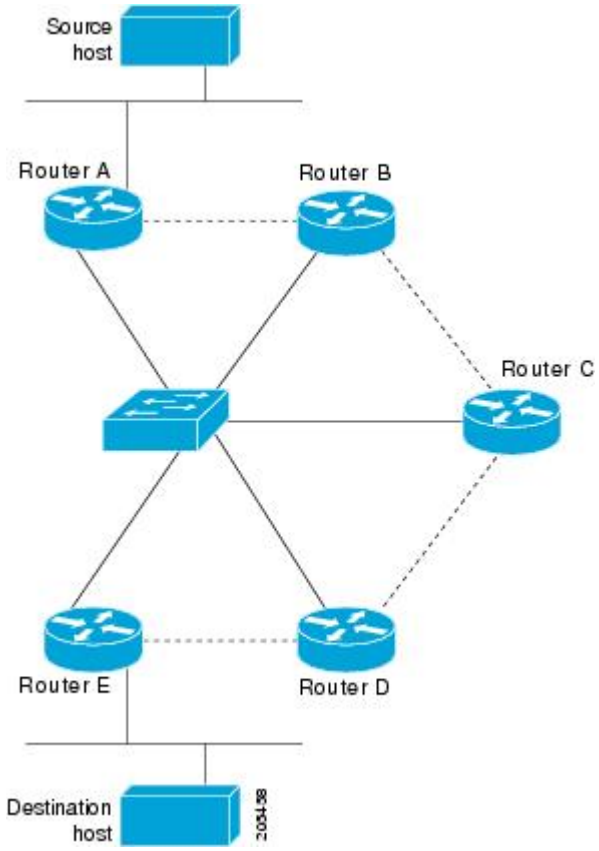
A logical NBMA network is considered the group of interfaces and hosts participating in NHRP and having the same network identifier. The figure below illustrates two logical NBMA networks (shown as circles) configured over a single physical NBMA network. Router A can communicate with routers B and C because they share network identifier (2). Router C can also communicate with routers D and E because they share network identifier 7. After address resolution is complete, router A can send IP packets to router C in one hop, and router C can send them to router E in one hop, as shown by the dotted lines.

Figure 104: Two Logical NBMA Networks over One Physical NBMA Network



The physical configuration of the five routers in the figure above might actually be that shown in the figure below. The source host is connected to router A and the destination host is connected to router E. The same switch serves all five routers, making one physical NBMA network.

Figure 105: Physical Configuration of a Sample NBMA Network



Refer again to the first figure above. Initially, before NHRP has resolved any NBMA addresses, IP packets from the source host to the destination host travel through all five routers connected to the switch before reaching the destination. When router A first forwards the IP packet toward the destination host, router A also generates an NHRP request for the IP address of the destination host. The request is forwarded to router C, whereupon a reply is generated. Router C replies because it is the egress router between the two logical NBMA networks.

Similarly, router C generates an NHRP request of its own, to which router E replies. In this example, subsequent IP traffic between the source and the destination still requires two hops to traverse the NBMA network, because the IP traffic must be forwarded between the two logical NBMA networks. Only one hop would be required if the NBMA network were not logically divided.

Applying NHRP Rates to Specific Destinations Example

In the following example, only the packets that pass extended access list 101 are subject to the default SVC triggering and teardown rates:

```
interface tunnel 100
 ip nhrp interest 101
!
access-list 101 permit ip any any
access-list 101 deny ip any 10.3.0.0 0.0.255.255
```

NHRP on a Multipoint Tunnel Example

With multipoint tunnels, a single tunnel interface may be connected to multiple neighboring routers. Unlike point-to-point tunnels, a tunnel destination need not be configured. In fact, if configured, the tunnel destination must correspond to an IP multicast address. Broadcast or multicast packets to be sent over the tunnel interface can then be sent by sending the GRE packet to the multicast address configured as the tunnel destination.

Multipoint tunnels require that you configure a tunnel key. Otherwise, unexpected GRE traffic could easily be received by the tunnel interface. For simplicity, we recommend that the tunnel key correspond to the NHRP network identifier.

In the following example, routers A and B share a GigabitEthernet segment. Minimal connectivity over the multipoint tunnel network is configured, thus creating a network that can be treated as a partially meshed NBMA network.

The significant portions of the configurations for routers A and B follow:

Router A Configuration

```
interface tunnel 1
 ip address 10.1.1.1 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp network-id 123
 no ip split-horizon eigrp 100
 tunnel source GigabitEthernet 0/0/7
 tunnel mode gre multipoint
 tunnel key 123
 tunnel protection ipsec profile DMVPN
interface GigabitEthernet 0/0/7
 ip address 10.1.2.1 255.255.255.0
```

Router B Configuration

```
interface tunnel 1
 ip address 10.1.1.2 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp map multicast 10.1.2.1
 ip nhrp map 10.1.1.1 10.1.2.1
 ip nhrp network-id 123
 ip nhrp nhs 10.1.1.1
 tunnel source GigabitEthernet 0/1
 tunnel mode gre multipoint
 tunnel key 123
 tunnel protection ipsec profile DMVPN
interface GigabitEthernet 0/1
 ip address 10.1.2.2 255.255.255.0
```

Show NHRP Examples

The following is sample output from the **show ip nhrp** command:

```
Router# show ip nhrp
```

```
10.1.1.2/32 via 10.1.1.2, Tunnel1 created created 22:59:16, expire 01:35:31
  Type: dynamic, Flags: unique registered
  NBMA address: 10.1.2.2
10.1.1.3/32 via 10.1.1.3, Tunnel1 created 21:59:16, expire 01:20:44
  Type: dynamic, Flags: unique registered
  NBMA address: 10.1.1.2
```

The fields in the sample display are as follows:

- The IP address and its network mask in the IP-to-NBMA address cache. The mask is always 255.255.255.255 (/32) because Cisco does not support aggregation of NBMA information through NHRP.
- The interface type and number and how long ago it was created (hours:minutes:seconds).
- The time in which the positive and negative authoritative NBMA address will expire (hours:minutes:seconds). This value is based on the **ip nhrp holdtime** command.
- Type of interface:
  - dynamic--NBMA address was obtained from the NHRP Request packet.
  - static--NBMA address was statically configured.
- Flags:
  - authoritative--Indicates that the NHRP information was obtained from the Next Hop Server or router that maintains the NBMA-to-IP address mapping for a particular destination.
  - implicit--Indicates that the information was learned from the source mapping information of an NHRP resolution request received by the local router, or from an NHRP resolution packet being forwarded through the local router.
  - negative--For negative caching; indicates that the requested NBMA mapping could not be obtained.
  - unique--Indicates that this NHRP mapping entry must be unique; it cannot be overwritten with a mapping entry that has the same IP address but a different NBMA address.
  - registered--Indicates the NHRP mapping entry was created by an NHRP registration request.
  - used--Indicates the NHRP mapping was used to forward data packets within the last 60 seconds.
  - router--Indicates an NHRP mapping entry that is from a remote router that is providing access to a network or host behind the remote router.
  - local--Indicates an NHRP mapping entry for networks local to this router for which this router has answered an NHRP resolution request.
  - (no socket)--Indicates an NHRP mapping entry for which IPsec socket (for encryption) has not been triggered. These mapping entries are not used to forward data packets.
  - nat--Indicates an NHRP mapping entry for which IPsec socket (for encryption) has not been triggered. These mapping entries are not used to forward data packets.
  - NBMA address--Nonbroadcast multiaccess address. The address format is appropriate for the type of network being used (for example, GRE, Ethernet, SMDS, or multipoint tunnel

The following example shows output for a specific tunnel, tunnel7:

Router# **show ip nhrp traffic interface tunnel0**

```
Tunnel0: Max-send limit:100Pkts/10Sec, Usage:0%
   Sent: Total 79
         18 Resolution Request  10 Resolution Reply  42 Registration Request
         0 Registration Reply  3 Purge Request  6 Purge Reply
         0 Error Indication  0 Traffic Indication
   Rcvd: Total 69
```

```
10 Resolution Request  15 Resolution Reply  0 Registration Request
36 Registration Reply  6 Purge Request  2 Purge Reply
0 Error Indication  0 Traffic Indication
```

The fields shown in the sample display are as follows:

- Tunnel0--Interface type and number.
- Max-send limit--Maximum number of NHRP messages that can be sent by this station in the given interval.
- Resolution Request--Number of NHRP resolution request packets originated from or received by this station.
- Resolution Reply--Number of NHRP resolution reply packets originated from or received by this station.
- Registration Request--Number of NHRP resolution reply packets originated from or received by this station.
- Registration Reply--Number of NHRP registration reply packets originated from or received by this station.
- Purge Request--Number of NHRP reply packets received by this station.
- Purge Reply--Number of NHRP register packets originated from this station. Routers and access servers do not send register packets, so this value is 0.
- Error Indication--Number of NHRP error packets originated from or received by this station.
- Traffic Indication--Number of NHRP traffic indication packets (redirects) originated or received from this station.

Additional References

The following sections provide references related to configuring NHRP.

Related Documents

Related Topic	Document Title
The DMVPN feature allows users to better scale large and small IP Security (IPsec) Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP).	"Dynamic Multipoint VPN" module
NRHP commands	*Cisco IOS IP Addressing Services Command Reference*

RFCs

RFC	Title
RFC 2332	NBMA Next Hop Resolution Protocol (NHRP)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring NHRP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 150: Feature Information for NHRP

Feature Name	Releases	Feature Configuration Information
Next Hop Resolution Protocol	Cisco IOS XE Release 2.1	NHRP is an Address Resolution Protocol (ARP)-like protocol that dynamically maps an NBMA network. With NHRP, systems attached to an NBMA network can dynamically learn the NBMA (physical) address of the other systems that are part of that network, allowing these systems to directly communicate. NHRP is a client and server protocol where the hub is the Next Hop Server (NHS) and the spokes are the Next Hop Clients (NHCs). The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes to build direct tunnels.

CHAPTER 110

Shortcut Switching Enhancements for NHRP in DMVPN Networks

Routers in a Dynamic Multipoint VPN (DMVPN) Phase 3 network use Next Hop Resolution Protocol (NHRP) Shortcut Switching to discover shorter paths to a destination network after receiving an NHRP redirect message from the hub. This allows the routers to communicate directly with each other without the need for an intermediate hop.

Information About Shortcut Switching Enhancements for NHRP

DMVPN Phase 3 Networks Overview

In a DMVPN Phase 3 network, separate regional DMVPN networks are connected together into a single hierarchical DMVPN network. Spokes in different regions use NHRP to build direct spoke-to-spoke tunnels with each other, bypassing both the regional and the central hubs. When building spoke-to-spoke tunnels within a region, only the regional hubs are involved in the tunnel setup. When building spoke-to-spoke tunnels between regions, the regional and the central hubs are involved in the tunnel setup.

DMVPN Phase 3 provides improvements over a DMVPN Phase 2 network. For a DMVPN spoke-to-spoke network, the main improvements from Phase 2 are in the increased flexibility in laying out the base DMVPN network. DMVPN Phase 3 allows a hierarchical hub design whereas DMVPN Phase 2 relies on “daisy-chaining” of hubs for scaling the network. DMVPN Phase 3 also removes some of the restrictions on the routing protocols required by Phase 2 (OSPF broadcast mode and non split-tunneling). DMVPN Phase 3 is not expected to change the number of spokes that a single DMVPN hub can support but it may reduce the CPU load of the routing protocol on the hub.

Benefits of NHRP Shortcut Switching Enhancements

Cisco has developed NHRP shortcut switching model enhancements that allow for more scalable DMVPN implementations. This model provides the following benefits:

- Allows summarization of routing protocol updates from hub to spokes. The spokes no longer need to have an individual route with an IP next hop of the tunnel IP address of the remote spoke for the networks behind all the other spokes. The spoke can use summarized routes with an IP next hop of the tunnel IP address of the hub and still be able to build spoke-to-spoke tunnels. It can reduce the load on the routing protocol running on the hub router. You can reduce the load because, when you can summarize the networks behind the spokes to a few summary routes or even one summary route, the hub routing protocol only has to advertise the few or one summary route to each spoke rather than all of the individual spoke routes. For example, with 1000 spokes and one router per spoke, the hub receives 1000 routes but only has to advertise one summary route to each spoke (equivalent to 1000 advertisements, one per spoke) instead of the 1,000,000 advertisements it had to process in the prior implementation of DMVPN.
- Provides better alternatives to static daisy-chaining of hubs for expanding DMVPN spoke-to-spoke networks. The hubs must still be interconnected, but they are not restricted to just a daisy-chain pattern. The routing table is used to forward data packets and NHRP control packets between the hubs. The routing table allows efficient forwarding of packets to the correct hub rather than having request and reply packets traversing through all of the hub routers.
- Allows for expansion of DMVPN spoke-to-spoke networks with OSPF as the routing protocol beyond two hubs. Because the spokes can use routes with the IP next-hop set to the hub router (not the remote spoke router as before), you can configure OSPF to use point-multipoint network mode rather than broadcast network mode. Configuring OSPF to use point-multipoint network mode removes the DR and BDR requirements that restricted the DMVPN network to just two hubs. When using OSPF, each spoke still has all individual routes, because the DMVPN network must be in a single OSPF area but you cannot summarize routes within an OSPF area.
- Allows routing protocols such as ODR to be used and still retain the ability to build dynamic spoke-to-spoke tunnels.
- Allows for hierarchical (greater than one level) and more complex tree-based DMVPN network topologies. Tree-based topologies allow the capability to build DMVPN networks with regional hubs that are spokes of central hubs. This architecture allows the regional hub to handle the data and NHRP control traffic for its regional spokes, but still allows spoke-to-spoke tunnels to be built between any spokes within the DMVPN network, whether they are in the same region or not.
- Enables the use of Cisco Express Forwarding to switch data packets along the routed path until a spoke-to-spoke tunnel is established.

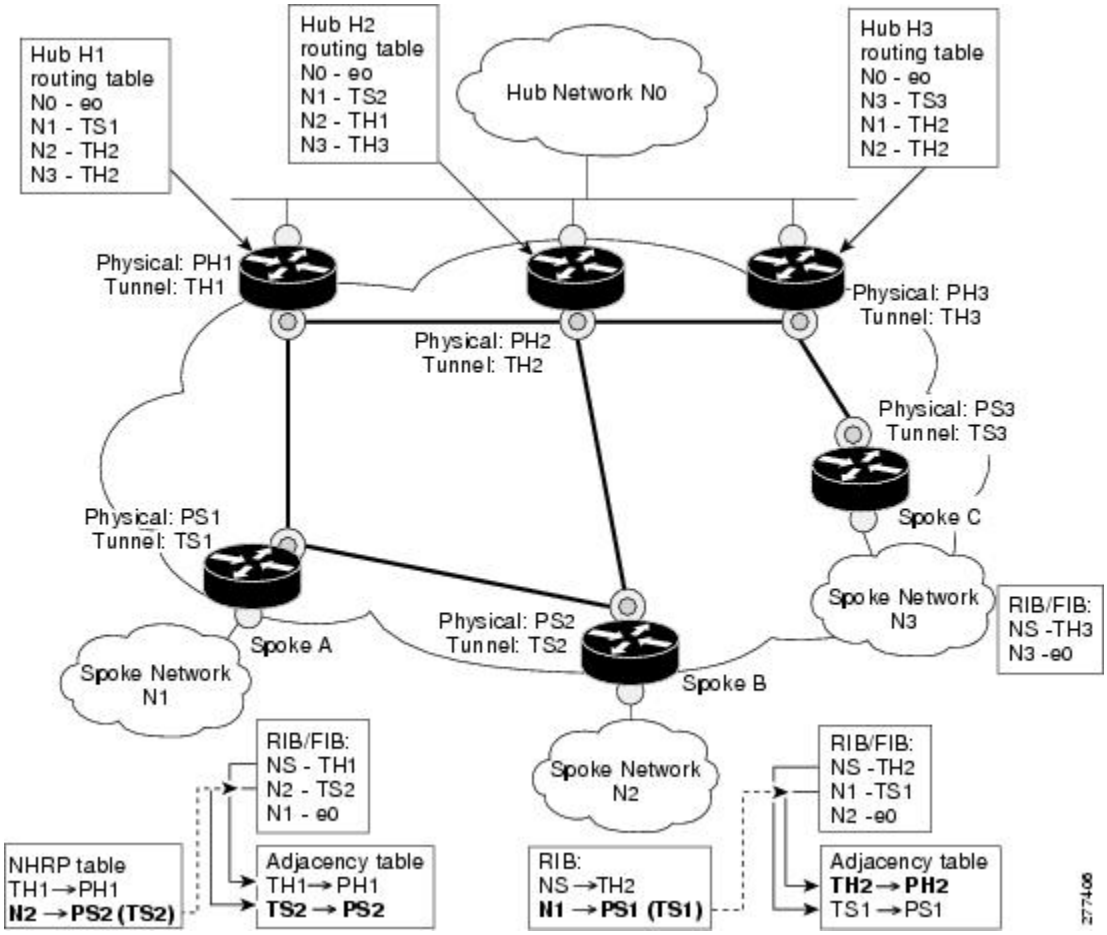
NHRP as a Route Source

To implement shortcut switching, NHRP works as a route source and installs shortcut paths, as NHRP routes, directly into the Routing Information Base (RIB). This means that shortcut paths appear as routes in the routing table and NHRP works in lieu of the routing protocol (for example, RIP, OSPF or EIGRP). The shortcut routes in the RIB are distributed into the Fowarding Information Base (FIB). When a spoke discovers a shortcut path, it adds the path as an NHRP route to its routing table. The RIB and FIB have no special behaviour for shortcut switching and shortcut routes are treated like any other route.

NHRP acts as a route producer to the RIB, but it does not function as a full routing protocol. NHRP manages the route registration, resolution, and purge messages but it does not discover or maintain NHRP neighbors, advertise NHRP routing messages, or inform the network of any network topology changes.

Consider Spoke A in the figure below. It discovers a shortcut path to N2 via Spoke 2's tunnel (overlay) address TS2. It installs the shortcut path in its NHRP mapping table via the entry N2-PS2 (TS2) and it also adds the route to the RIB. The new route in the RIB is then distributed into the FIB and the FIB installs the corresponding adjacency TS2-PS2 in the adjacency table. The new route TS2-PS2 can now be used for forwarding. Note the consistency between the RIB, the FIB, and the adjacency table.

Figure 106: NHRP As A Route Source



Next Hop Overrides

If an NHRP route in the RIB is identical to another route (owned by another protocol) in the RIB then NHRP overrides the other protocol's next hop entries by installing shortcut next hops in the RIB. NHRP installs shortcut paths into the routing table, not as NHRP routes but as local forwarding paths. The other routing protocols continue to function as normal managing route redistribution and advertisement. NHRP only overrides local forwarding decisions by installing alternate or backup next hops into the routing table.

NHRP Route Watch Infrastructure

In a DMVPN full-mesh design, the hub creates summary routes to each of the spokes (Interior Gateway Protocol (IGP) routes). Specific NHRP shortcuts are installed at the spokes by NHRP as and when required. These shortcuts can be viewed as a refinement of the route summaries because they deal with a specific subnet while the summary routes represent super-nets. If the summary route is absent, NHRP cannot discover a shortcut path.

The summary route, or “covering prefix”, governs the existence of the NHRP route in the RIB. The removal of a covering prefix in the RIB would lead to the removal of the all the corresponding NHRP routes, that were learnt via this covering prefix, from the RIB. The tracking of covering prefixes is done via the Route Watch infrastructure.

A "watched prefix" is a route that immediately precedes an NHRP route. For example, if an NHRP route is 172.16.3.0/24, then the watch-prefix corresponding to it would be 172.16.2.0/23. Each “watched prefix” and its associated “covering prefixes” are tracked by the Route Watch service. A “covering prefix” is defined as the longest matching IGP route in the RIB which is less specific than the “watched prefix”. The validity of each NHRP shortcut is determined by the following events:

- If a “covering prefix” is removed so that there is no other IGP route in the RIB "covering" the watched prefix, (the watched prefix is unreachable), then the corresponding NHRP shortcut route is removed.
- If a new IGP route, which is more specific than the covering prefix but less specific than watched prefix, is installed in RIB, then it will become the covering prefix for the watched prefix. If the new covering prefix has a different next hop associated with it, the original shortcut is removed.

In summary, the validity of an NHRP route in the RIB is determined by the less specific, longest match IGP route present in the RIB. NHRP shortcuts are refinements to the routing topology, so shortcut paths are added to the RIB without modifying the routing topology.

From Cisco IOS XE Release 17.10.1a, NHRP supports the Route Watch attribute for the next hop of a cache entry that has a destination of a different subnet. This behavior is automatically triggered and does not depend on whether Route Watch is enabled or not. Therefore, a cache entry can now have two Route Watch instances: One Route Watch instance for the prefix entry and another Route Watch instance for the next hop.

Supporting the Route Watch attribute for next hop entries ensures that cache entries are cleaned up and monitored when route details disappear or are modified therefore preventing traffic drops.

NHRP Purge Request Reply

When an NHRP hub replies to a resolution request, it creates a local NHRP mapping entry. The local mapping entry is a network entry for which NHRP has sent a reply. The local mapping entry maintains a list of requesters. When a network entry is modified or deleted in the routing table, NHRP is notified of the event. NHRP finds the local cache entry for the network and sends a purge request to the requesters that the network to which it previously replied has changed. The receivers of the purge message delete the corresponding NHRP mapping entry from its table and send a purge reply indicating that the purge message was processed successfully.

How to Configure Shortcut Switching for NHRP

NHRP Smart Defaults

NHRP Smart default commands are:

- **ip|ipv6 nhrp map multicast dynamic**
- **ip|ipv6 nhrp registration no-unique**
- **ip|ipv6 nhrp holdtime 600**—default hold time is 6 mins and registrations are sent every 2 mins
- **ip|ipv6 nhrp shortcut**—enabled or disabled by default according to whether or not the interface is multipoint or p2p
- **ip|ipv6 nhrp network-id** —enabled by default where ID is the tunnel key or the tunnel interface number (in the absence of a tunnel key)
- **ip|ipv6 nhrp path preference** —the preference is 255 by default, meaning spoke-spoke routes are always ECMP irrespective of the spoke-hub cost ratio (unless the preference ratio is configured to match the IGP metric ratio). NHRP cache entries are created with a preference that is received in the packet. The preference that is sent in the packet is based on what is configured on the interface using **ip|ipv6 nhrp path preference <1-255>**. The ratio of preferences for cache entries created for the same prefix also decides the ratio of metric of NHRP routes (ratio of metric is the inverse ratio of preference). Hence, CEF load balances traffic over multiple paths in the ratio of the corresponding cache preferences. This can be used for egress load-balancing (equal or unequal cost) or ingress traffic engineering over a dynamic spoke-spoke tunnel. The default value of the cache preference is changed to 255 from 0.

Note The default values do not display when you use the **show run** command but are displayed when you use **show run all** command. However, user configured values override default values.

Enabling NHRP Shortcut Switching on an Interface

Perform this task to enable shortcut switching for NHRP for an interface on a router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nhrp shortcut**
5. **end**
6. **show ip nhrp shortcut**
7. **show ip route nhrp**
8. **show ip route next-hop-override**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Tunnel 0`	Enters interface configuration mode.
Step 4	**ip nhrp shortcut** **Example:** `Router(config-if)# ip nhrp shortcut`	Enables NHRP shortcut switching on an interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Ends the configuration session.
Step 6	**show ip nhrp shortcut** **Example:** `Router# show ip nhrp shortcut`	(Optional) Displays only the NHRP cache entries that have an NHRP route or an NHRP next-hop override associated with them.
Step 7	**show ip route nhrp** **Example:** `Router# show ip route nhrp`	(Optional) Displays the routes added to the routing table by NHRP.
Step 8	**show ip route next-hop-override** **Example:** `Router# show ip route next-hop-override`	(Optional) Displays the NHRP next-hop overrides associated with a particular route, along with the corresponding default next hops.

Clearing NHRP Cache Entries on an Interface

Perform this optional task to clear NHRP cache entries that have associated NHRP routes and next-hop overrides on an interface on a router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clear ip nhrp shortcut** *interface-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**clear ip nhrp shortcut** *interface-name* **Example:** `Router(config)# clear ip nhrp shortcut Tunnel0`	Clears NHRP cache entries on an interface.
Step 4	**end** **Example:** `Router(config)# end`	Ends the configuration session.

Configuration Examples for Shortcut Switching Enhancements for NHRP

Configuring NHRP Shortcut Switching Example

The following example configures NHRP shortcut switching on tunnel interface 1:

```
Router(config)#
interface Tunnel 1
Router(config-if)#
ip nhrp shortcut
```

The following example shows the output of the **show ip route** and **show ip route nhrp** commands. These commands can be used to show the current state of the routing table. NHRP entries are flagged “H”.

```
Router#
show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
Gateway of last resort is not set
```

```
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Tunnel0
C       172.16.22.0 is directly connected, Ethernet1/0
H       172.16.99.0 [250/1] via 1.1.1.99, 00:11:43, Tunnel0
     10.2.2.0/24 is subnetted, 1 subnets
C       10.11.11.0 is directly connected, Ethernet0/0
Router#
show ip route nhrp
H       172.16.99.0 [250/1] via 10.1.1.99, 00:11:43, Tunnel0
```

The following sample output displays the NHRP next-hop overrides associated with a particular route and the corresponding default next hops, when the following next-hop override is added:

- IP address: 10.50.10.0
- Mask: 255.255.255.0
- Gateway: 10.1.1.1
- Interface: Tunnel0

```
Router#
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
       + - replicated route
Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.2.1.0/24 is directly connected, Loopback1
L        10.2.1.1/32 is directly connected, Loopback1
      10.50.0.0/24 is subnetted, 1 subnets
% S        10.50.10.0 is directly connected, Tunnel0
      10.30.0.0/24 is subnetted, 1 subnets
S        10.30.11.0 is directly connected, Ethernet0/0
Router#
show ip route next-hop-override
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
       + - replicated route
Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.2.1.0/24 is directly connected, Loopback1
L        10.2.1.1/32 is directly connected, Loopback1
      10.50.0.0/24 is subnetted, 1 subnets
% S        10.50.10.0 is directly connected, Tunnel0
                    [NHO][1/0] via 10.1.1.1, Tunnel0
      10.30.0.0/24 is subnetted, 1 subnets
S        10.30.11.0 is directly connected, Ethernet0/0
Router#
show ip cef
Prefix              Next Hop            Interface
10.2.1.255/32         receive             Loopback110.10.10.0/24
```

```
10.50.10.0/24       10.1.1.1              Tunnel0
10.30.11.0/24       attached            Ethernet0/0
127.0.0.0/8           drop
```

The following example displays the output of the **show ip route** and **show ip route next-hop-override** commands after the following next-hop override is deleted:

- IP address: 10.50.10.0
- Mask: 255.255.255.0
- Gateway: 10.1.1.1
- Interface: Tunnel0

```
Router#
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
       + - replicated route
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.2.1.0/24 is directly connected, Loopback1
L        10.2.1.1/32 is directly connected, Loopback1
      10.50.0.0/24 is subnetted, 1 subnets
% S        10.50.10.0 is directly connected, Tunnel0
      10.30.0.0/24 is subnetted, 1 subnets
S        10.30.11.0 is directly connected, Ethernet0/0
Router#
show ip route next-hop-override
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
       + - replicated route

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.2.1.0/24 is directly connected, Loopback1
L        10.2.1.1/32 is directly connected, Loopback1
      10.50.0.0/24 is subnetted, 1 subnets
S        10.50.10.0 is directly connected, Tunnel0
      10.30.0.0/24 is subnetted, 1 subnets
S        10.30.11.0 is directly connected, Ethernet0/0
Router#
show ip cef
Prefix              Next Hop            Interface
10.2.1.255/32        receive             Loopback110.10.10.0/24
10.50.10.0/24       attached            Tunnel0
10.30.11.0/24       attached            Ethernet0/0
127.0.0.0/8         drop
```

The following sample output shows the information displayed by the **show ip nhrp** command when a cache entry has an associated NHRP next-hop override in the RIB. Note that the flags for the entry are displayed as "router rib" and not "router candidate".

```
Router#
show ip nhrp
10.1.1.22/32 via 10.1.1.22
   Tunnel0 created 00:00:06, expire 00:02:23
   Type: dynamic, Flags: router implicit
   NBMA address: 10.11.11.22
10.1.1.99/32 via 10.1.1.99
   Tunnel0 created 4d04h, never expire
   Type: static, Flags: used
   NBMA address: 10.11.11.99
172.16.11.0/24 via 10.1.1.11
   Tunnel0 created 00:00:06, expire 00:02:23
   Type: dynamic, Flags: router unique local
   NBMA address: 10.11.11.11
    (no-socket)
172.16.22.0/24 via 10.1.1.22
   Tunnel0 created 00:00:05, expire 00:02:24
   Type: dynamic, Flags: router rib
   NBMA address: 10.11.11.22
```

The following example shows the output displayed by the **show ip nhrp** command when a cache entry has an NHRP next-hop override added to the RIB. If the corresponding cache entry has an associated NHRP next-hop override in the RIB, the flags are displayed as "router rib nho".

```
Router#
show ip nhrp
10.1.1.22/32 via 10.1.1.22
   Tunnel0 created 00:00:06, expire 00:02:23
   Type: dynamic, Flags: router implicit
   NBMA address: 10.11.11.22
10.1.1.99/32 via 10.1.1.99
   Tunnel0 created 4d04h, never expire
   Type: static, Flags: used
   NBMA address: 10.11.11.99
172.16.11.0/24 via 10.1.1.11
   Tunnel0 created 00:00:06, expire 00:02:23
   Type: dynamic, Flags: router unique local
   NBMA address: 10.11.11.11
    (no-socket)
172.16.22.0/24 via 10.1.1.22
   Tunnel0 created 00:00:05, expire 00:02:24
   Type: dynamic, Flags: router rib nho
   NBMA address: 10.11.11.22
```

The following example shows the output displayed by the **show ip nhrp shortcut** command. This command displays only the NHRP cache entries that have an associated NHRP route or NHRP next-hop override.

```
Router#
show ip nhrp shortcut
172.16.22.0/24 via 10.1.1.22
   Tunnel0 created 00:00:05, expire 00:02:24
   Type: dynamic, Flags: router rib
   NBMA address: 10.11.11.22
172.16.22.0/24 via 10.1.1.22
   Tunnel0 created 00:00:05, expire 00:02:24
   Type: dynamic, Flags: router rib nho
   NBMA address: 10.11.11.22
```

The following example shows the output displayed by the **show dmvpn** command. The output indicates a route installation in the attributes section of the command output.

```
Router#
show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
N - NATed, L - Local, X - No Socket, T1 - Route Installed,
T2 - Nexthop-override
# Ent --> Number of NHRP entries with same NBMA peer
NHS Status: E --> Expecting Replies, R --> Responding
UpDn Time --> Up or Down Time for a Tunnel
==========================================================================
Interface: Tunnel0, IPv4 NHRP Details
IPv4 Registration Timer: 60 seconds
IPv4 NHS: 10.1.1.99 RE
Type:Spoke, Total NBMA Peers (v4/v6): 2
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    2    10.11.11.22        192.1.1.22    UP 00:10:11    D        192.1.1.22/32
    0    10.11.11.22        173.1.1.22    UP 00:10:11   DT1     172.16.22.0/24
    1    10.11.11.99        173.1.1.99    UP 02:18:29    S        173.1.1.99/32
```

The example shows how to clear NHRP cache entries on tunnel interface 1 that have associated NHRP routes or nexthop overrides:

```
Router(config)# clear ip nhrp shortcut Tunnel1
```

Additional References

The following sections provide references related to NHRP and DMVPN.

Related Documents

Related Topic	Document Title
NHRP information and configuration tasks	"Configuring NHRP" module of the *Cisco IOS XE IP Addressing Services Configuration Guide* .
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
NHRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
Dynamic Multipoint VPN	"Dynamic Multipoint VPN" module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Shortcut Switching Enhancements for NHRP in DMVPN Networks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 151: Feature Information for Shortcut Switching Enhancements for NHRP in DMVPN Networks

Feature Name	Releases	Feature Information
Next Hop Resolution Protocol (NHRP)-CEF Rewrite for DMVPN Phase 3 Networks.	Cisco IOS XE Release 2.5 Cisco IOS XE Release 3.9S	Routers in a Dynamic Multipoint VPN (DMVPN) Phase 3 network use Next Hop Resolution Protocol (NHRP) Shortcut Switching to discover shorter paths to a destination network after receiving an NHRP redirect message from the hub. This allows the routers to communicate directly with each other without the need for an intermediate hop. The following commands were introduced or modified: **clear ip nhrp shortcut, debug dmvpn, debug nhrp routing, ip nhrp shortcut**, **show dmvpn**, **show ip nhrp, show ip nhrp shortcut**, **show ip route**, **show ip route next-hop-override**.

PART X

Easy Virtual Network

CHAPTER 111

Overview of Easy Virtual Network

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end virtualization of two or more Layer-3 networks. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other.

EVN builds on the existing IP-based virtualization mechanism known as VRF-Lite. EVN provides enhancements in path isolation, simplified configuration and management, and improved shared service support. EVN is backward compatible with VRF-Lite to enable seamless network migration from VRF-Lite to EVN.

EVN supports IPv4, static routes, Open Shortest Path First version 2 (OSPFv2), and Enhanced Interior Gateway Routing Protocol (EIGRP) for unicast routing, and Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP) for IPv4 Multicast routing. EVN also supports Cisco Express Forwarding (CEF) and Simple Network Management Protocol (SNMP).

Prerequisites for Configuring EVN

- Implementing EVN in a network requires a single IP infrastructure that you want to virtualize into two or more logical networks or L3VPNs. EVN provides path isolation for the traffic on the different virtual networks.
- You must have a functioning campus design in place before adding virtualization to a network.
- You should understand virtual routing and forwarding (VRF) instances and how they are used to maintain traffic separation across the network.

Restrictions for EVN

- An EVN trunk is allowed on any interface that supports 802.1q encapsulation, such as Fast Ethernet, Gigabit Ethernet, and port channels.

- There are additional platform and line-card restrictions for an EVN trunk. Check Cisco Feature Navigator, www.cisco.com/go/cfn for supported platforms and line cards.
- A single IP infrastructure can be virtualized to provide up to 32 virtual networks end-to-end.
- If an EVN trunk is configured on an interface, you cannot configure VRF-Lite on the same interface.
- OSPFv3 is not supported; OSPFv2 is supported.
- The following are not supported by EVN:
  - IS-IS
  - RIP
  - Route replication is not supported with BGP
  - Certain SNMP set operations
- The following are not supported on an EVN trunk:
  - Access control lists (ACLs)
  - BGP interface commands are not inherited
  - IPv6, except on vnet global
  - Network address translation (NAT)
  - NetFlow
  - Web Cache Communication Protocol (WCCP)

Information About EVN

Benefits of EVN

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end virtualization over Layer-3 networks. Network virtualization can be used to secure a network and to reduce network expenses by utilizing the same network infrastructure for multiple virtual networks. You can leverage the same physical infrastructure multiple times by supporting multiple groups, each with their own logical network and unique routing and forwarding tables.

Prior to network virtualization, path isolation can be achieved by:

- Separating pathis using dedicated routers which is more expensive than virtual networks.
- Using access control lists (ACLs), but ACLs do not support unique routing and forwarding tables, can be expensive to maintain, and more prone to error than virtual networks.

EVN provides the following benefits:

- Reduced capital expenditures by not having to maintain separate physical infrastructures to keep traffic isolated. One IP network has two or more virtual networks with traffic path isolation thereby saving the expense of additional hardware.
- Increased business flexibility, due to the ease of network integration for mergers, acquisitions, and business partners.
- Reduced network complexity due to a decrease in the infrastructure requirements for maintaining traffic separation through the core of the network.

- Build on the existing mechanism known as Multi-VRF (VRF-Lite). EVN is compatible with VRF-Lite. See the EVN Compatibility with VRF-Lite section. EVN is recommended over VRF-Lite because EVN provides enhancements in path isolation, simplified configuration and management, and improved shared service support.

  In addition to maintaining traffic separation between business units within a company, there are other scenarios in which path isolation is beneficial, including the following:

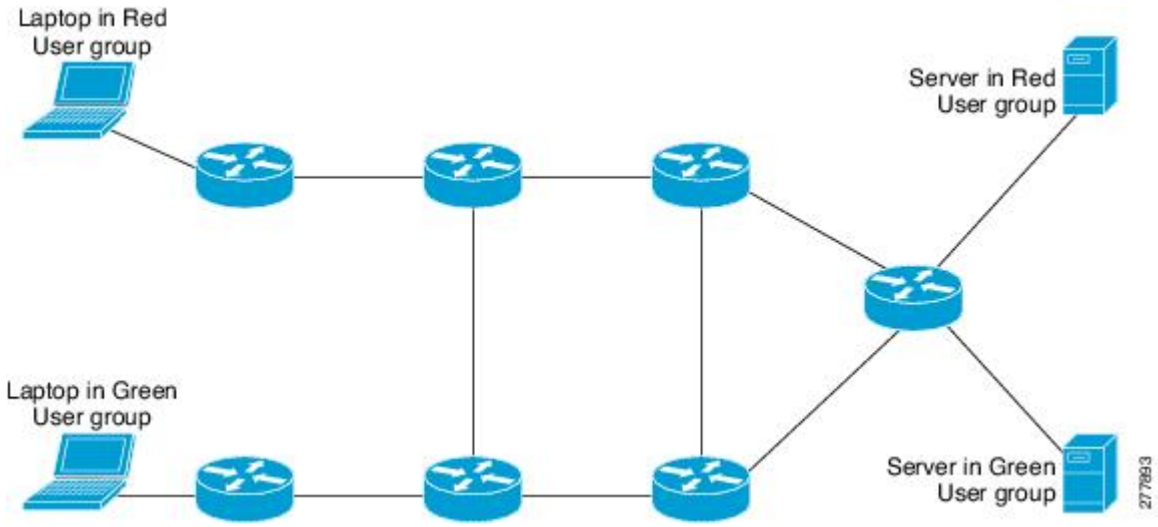
  - Guest access to the Internet—Restricting a guest's network access to the Internet, using a predetermined data path through the customer's network, and being able to define a unique default route for guest traffic.
  - Network Admission Control (NAC) isolation—Isolating the traffic sourced from a noncompliant desktop.
  - Partner access—Restricting partners and contractors to access a network's shared services, such as the Internet, e-mail, DNS, DHCP, or an application server.
  - Application and device isolation—Securing services and devices by "forcing" traffic to a centralized firewall where the traffic is subject to inspection.
  - Outsourcing services—Separating data traffic of various clients from each other.
  - Scalable network—Restricting a portion of the network to traffic that requires a very strict service level, which can lower costs by providing those requirements only where needed.
  - Subsidiaries/mergers/acquisitions—Consolidating companies or networks in stages, while enabling them to share services, when required.
  - Enterprise acting as a service provider—Requiring a separate network under a single authority for autonomous groups. An example is an airport authority supporting a virtual network per airline.

Virtual Network Tags Provide Path Isolation

It is not uncommon to have different user groups running on the same IP infrastructure. Various business reasons require traffic isolation between different groups. The figure below shows two user groups, Red and Green, running on the same network. Prior to network virtualization, there is no separation of traffic between the two groups. Users in the Red user group can access the server in the Green user group, and vice versa.

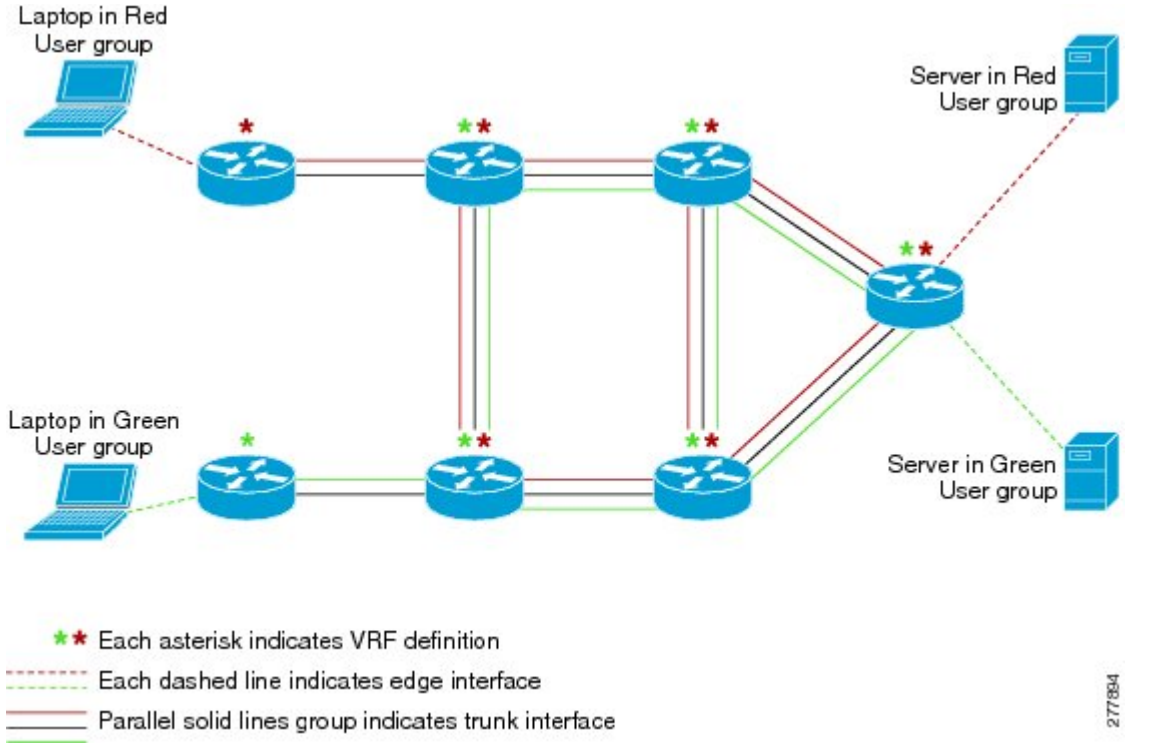
Without network virtualization, path isolation can be achieved by using access control, which is expensive to maintain, prone to error and does not support unique routing and forwarding tables per network.

Figure 107: Network without Virtualization



Virtual networks provide a coarse-grained segmentation of different user groups on one physical network. By configuring virtual networks, you can virtualize a single IP infrastructure to provide a number of virtual networks end to end. In the figure below, a single IP infrastructure is virtualized into two VPNs by creating two VRFs, Red and Green.

Figure 108: Network with Virtualization



In addition to utilizing VRFs to provide device-level separation, each virtual network has path isolation from the other. Path isolation is achieved by tagging the traffic so it carries the same tag value throughout the same

virtual network. Each network device along the path uses the tags to provide separation among different VRFs. A single tag number ties VRF red, for example, on one router to VRF red on another router.

Virtual Network Tag

Each VPN and associated EVN has a tag value that you assign during configuration. The tag value is global, meaning that on each router, the same EVN must be assigned the same numerical tag value. Tag values range from 2 to 4094.

Note When configuring EVN on a Cisco Catalyst 6500 Family networking device, we recommend you assign a vnet tag in the range 2 to1000. Beginning with Cisco IOS Release 15.1(1)SY, on the Sup2T platform of the Cisco Catalyst 6000 product lines, if the **vlan internal allocation policy descending** command is configured, the **vnet tag** range is from 2 to 3900.

An EVN is allowed on any interface that supports 802.1q encapsulation, such as Fast Ethernet, Gigabit Ethernet, and port channels. To allow for backward compatibility with the VRF-Lite solution, the vLAN ID field in the 802.1q frame is used to carry the virtual network tag.

Traffic that carries a virtual network tag is called tagged traffic. Traffic that does not carry a virtual network tag is called untagged traffic.

Tags are illustrated in the following configuration with two VRFs, red and green:

```
! Define two VRFs, red and green.
vrf definition red
 vnet tag 101
!
 address-family ipv4
 exit-address-family
!
vrf definition green
 vnet tag 102
!
 address-family ipv4
 exit-address-family
!
```

A virtual network is defined as a VRF instance with a virtual network tag assigned.

vnet Global

A predefined EVN known as “vnet global” is on the device. It refers to the global routing context and it corresponds to the default RIB. In figure 2 and figure 3, vnet global is represented by a black line connecting routers. The vnet global carries untagged traffic. By default, interfaces belong to the vnet global. Furthermore, vnet global is always running on trunk interfaces. The vnet global is also known as the default routing table.

Note IPv6 traffic is supported in vnet global only.

Edge Interfaces and EVN Trunk Interfaces

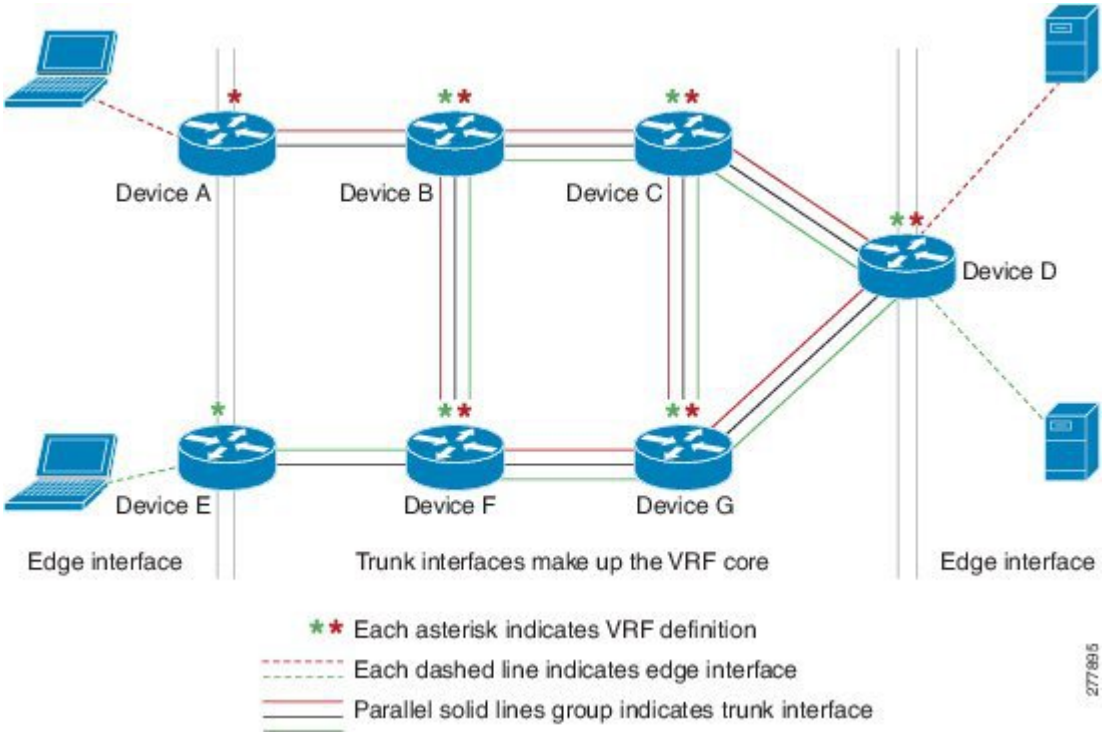
User devices are connected to a Layer 2 switch port, which is assigned to a VLAN. A VLAN can be thought of as a Layer 2 VPN. Customers will group all of the devices that need to be supported in a common Layer 3 VPN in a single VLAN. The point where data traffic is handed off between a VLAN and VRF is called an edge interface.

- An edge interface connects a user device to the EVN and in effect defines the boundary of the EVN. Edge interfaces connect end devices such as hosts and servers that are not VRF-aware. Traffic carried over the edge interface is untagged. The edge interface classifies which EVN the received traffic belongs to. Each edge interface is configured to belong to only one EVN.
- An EVN trunk interface connects VRF-aware routers together and provides the core with a means to transport traffic for multiple EVNs. Trunk interfaces carry tagged traffic. The tag is used to de-multiplex the packet into the corresponding EVN. A trunk interface has one subinterface for each EVN. The **vnet trunk** command is used to define an interface as an EVN trunk interface.

An EVN interface uses two types of interfaces: edge interfaces and trunk interfaces. An interface can be an edge or trunk interface, but not both. Figure 3 illustrates Routers A and D, which have edge interfaces that belong to VRF Red. Routers D and E have edge interfaces that belong to VRF Green.

Routers B, C, D, F, and G have trunk interfaces that make up the EVN core. These five routers have interfaces that belong to both VRF Red and VRF Green.

Figure 109: EVN Edge and EVN Trunk Interfaces



Identifying Trunk Interfaces in Display Output

Because a trunk interface carries multiple EVNs, sometimes it is not sufficient to display only the trunk interface name. When it is necessary to indicate that display output pertains to a particular EVN running on the trunk interface, the convention used is append a period and the virtual network tag, making the format *interface.virtual-network-tag*. Examples are gigabitethernet1/1/1.101 and gigabitethernet1/1/1.102.

By default, when a trunk interface is configured, all of the EVNs and associated virtual network tags are configured, and a virtual network subinterface is automatically created. As stated above, a period and the virtual network tag number are appended to the interface number.

In the following example, VRF red is defined with virtual network tag 3. Hence, the system created Fast Ethernet 0/0/0.3 (in VRF red).

```
Router# show running-config vrf red

Building configuration...
Current configuration : 1072 bytes
vrf definition red
 vnet tag 3
 !
 address-family ipv4
 exit-address-family
 !
```

You can display this hidden interface with the **show derived-config** command and see that all of the commands entered on Fast Ethernet 0/0/0 have been inherited by Fast Ethernet 0/0/0.3:

```
Router# show derived-config interface fastethernet0/0/0.3

Derived configuration : 478 bytes
!
interface FastEthernet0/0/0.3
 description Subinterface for VRF NG red
 vrf forwarding red
 encapsulation dot1Q 3
 ip address 10.1.1.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 ip summary-address eigrp 1 10.0.0.0 255.0.0.0
end
```

Single IP Address on Trunk Interfaces

A trunk interface can carry traffic for multiple EVNs. To simplify the configuration process, all the subinterfaces and associated EVNs have the same IP address assigned. In other words, a trunk interface is identified by the same IP address in different EVN contexts. This is because each EVN has a unique routing and forwarding table, thereby enabling support for overlapping IP addresses across multiple EVNs.

Relationship Between VRFs Defined and VRFs Running on a Trunk Interface

By default, the trunk interfaces on a router will carry traffic for all VRFs defined by the **vrf definition** command. For example, in the following configuration, every VRF defined on the router is included on the interface:

```
interface FastEthernet 1/0/0
 vnet trunk
 ip address 10.1.1.1 255.255.255.0
```

However, you might want to enable only a subset of VRFs over a certain trunk interface for traffic separation purposes. This is achieved by creating a VRF list, which is referenced in the **vnet trunk** command. When a trunk interface is enabled with a VRF list, only VRFs on the list are enabled on the interface. The exception is that **vnet global** is always enabled on the trunk interface.

In the following example, only the two specified VRFs on the list (red and green) are enabled on the interface:

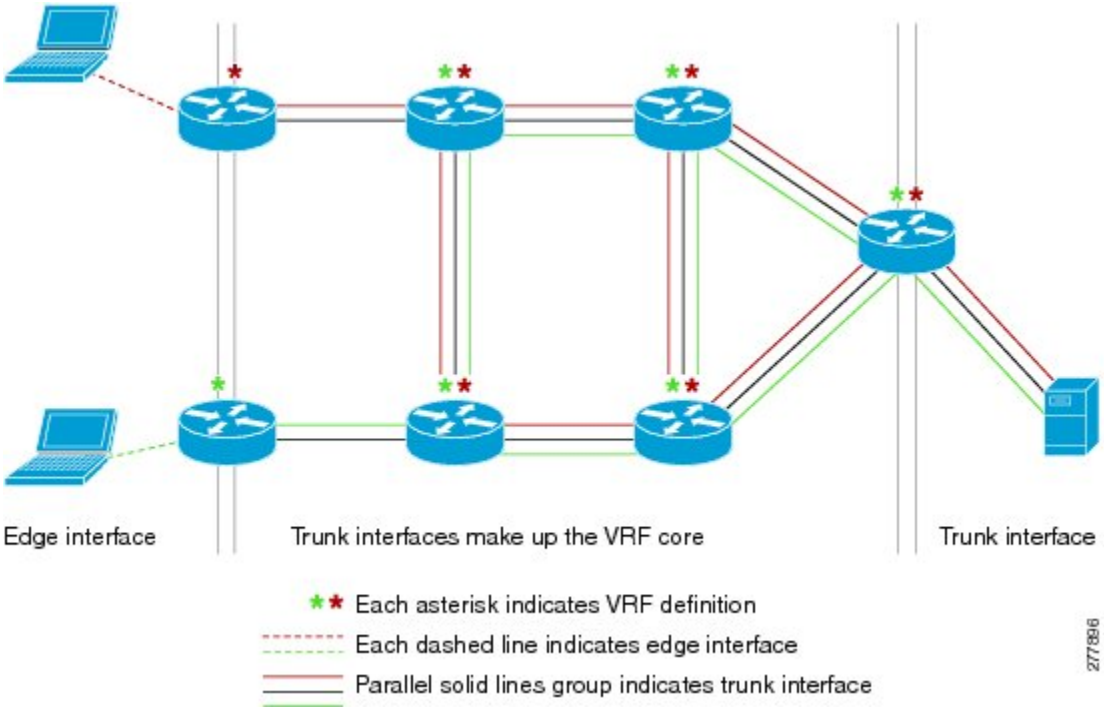
```
vrf list mylist
 member red
 member green
!
interface FastEthernet 1/0/0
 vnet trunk list mylist
 ip address 10.1.1.1 255.255.255.0
```

VRF Awareness

A device connected to a virtual network may not understand virtual network tags and can send and receive only untagged traffic. Such a device is referred to as VRF unaware. For example, a laptop computer is usually VRF unaware.

By contrast, a device that can send and receive tagged traffic and therefore takes the tag value into account when processing such traffic is known as VRF aware. For example, a VRF-aware server shared among different EVNs could use the virtual network tag to distinguish requests received and send responses. A VRF-aware device is connected to the EVN using a trunk interface, as shown in figure 4.

Figure 110: VRF Aware Server



The term “VRF aware” can also be used to describe a software component running on the router. A software component is VRF aware if it can operate on different EVNs. For example, ping is VRF aware because it allows you to choose which EVN to send the ping packet over.

Routing Protocols Supported by EVN

Each EVN runs a separate instance of a routing protocol. This allows each EVN to fine-tune its routing separately and also limits fate sharing. Different virtual networks may run different routing protocols concurrently.

EVN supports static routes, OSPFv2, and EIGRP for unicast routing, and PIM, MSDP, and IGMP for multicast routing.

Packet Flow in a Virtual Network

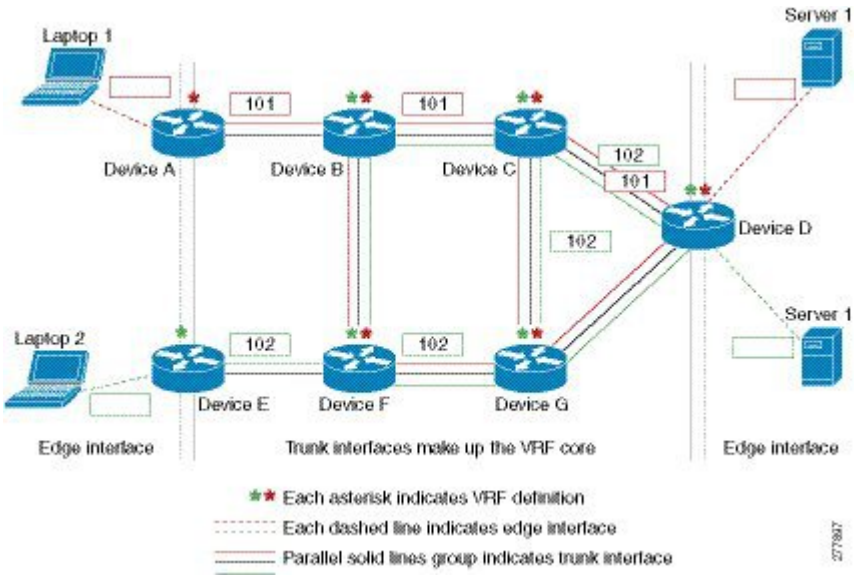
Packets enter an EVN through an edge interface, traverse multiple trunk interfaces, and exit the virtual network through another edge interface. At the ingress edge interface, packets are mapped from a VLAN into a particular EVN. Once the packet is mapped to an EVN, it is tagged with the associated virtual network tag. The virtual network tag allows the trunk interface to carry packets for multiple EVNs. The packets remain tagged until they exit the EVN through the egress edge interface.

On the edge interface, the EVN associated with the interface is used for route lookup. On the trunk interface, the virtual network tag carried in the packet is used to locate the corresponding EVN for routing the packets.

If the egress interface is an edge interface, the packet is forwarded untagged. However, if the egress interface is a trunk interface, the packet is forwarded with the tag of the ingress EVN.

The figure below illustrates how traffic from two VRFs, red and green, can coexist on the same IP infrastructure, using the tags 101 and 102.

Figure 111: Packet Flow in a Virtual Network



The packet flow from Laptop 1 to Server 1 in VRF red occurs as follows:

1. Laptop 1 send an untagged packet to Server 1.
2. Router A receives the packet over an edge interface, which is associated with VRF red.
   a. Router A does route lookup in VRF red and sees that the next hop is Router B through a trunk interface.
   b. Router A encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
3. Router B receives the packet over a trunk interface. Seeing virtual network tag 101, Router B identifies that the packet belongs to VRF red.
   a. Router B does route lookup in VRF red and sees that the next hop is Router C through a trunk interface.
   b. Router B encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
4. Router C receives the packet over a trunk interface. Using virtual network tag 101, Router C identifies that the packet belongs to VRF red.
   a. Router C does route lookup in VRF red and sees that the next hop is Router D through a trunk interface.
   b. Router C encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
5. Router D receives the packet over a trunk interface. Using virtual network tag 101, Router D identifies that the packet belongs to VRF red.
   a. Router D does route lookup in VRF red and sees that the next hop is through an edge interface.
   b. Router D sends the untagged packet over the edge interface to Server 1.
6. Server 1 receives the untagged packet originated from Laptop 1.

Command Inheritance on EVN Trunk Interfaces

One of the benefits of EVN is the ability to easily configure multiple EVNs on a common trunk interface without the need to configure each interface associated with an EVN individually. An EVN trunk interface takes advantage of the fact that the configuration requirements for different EVNs will be similar over a single trunk interface. When specific commands are configured on the trunk interface, they define default values that are inherited by all EVNs running over the same interface, including **vnet global**. If you feel that the settings are acceptable for all of the EVNs sharing an interface, then no individual configuration is necessary.

For example, the OSPF hello interval can be set for all EVNs over the trunk interface with one line of configuration, as follows:

```
interface gigabitethernet1/1/1
 vnet trunk
 ip address 10.1.2.1 255.255.255.0
 ! set OSPF hello interval for all VRFs on this interface.
 ip ospf hello-interval 20
```

The list of commands configured on the trunk interface whose values are inherited by all EVNs running on the same interface is provided in the table in "Commands Whose Values Can be Inherited Or Overridden by a Virtual Network on an Interface" section.

For more examples of command inheritance, see the configuration examples in the *Configuring Easy Virtual Networks* module.

Overriding Command Inheritance Virtual Network Interface Mode

You might want some EVNs on the same trunk interface to have different configurations. An alternative to command inheritance is to selectively override inherited values by using specific commands in virtual network interface mode for individual EVNs. In this mode, the command's settings override the Cisco default value or the value you set in interface configuration mode.

In interface configuration mode, entering the **vnet name** command causes the system to enter virtual network interface mode. The system prompt for this mode is Router(config-if-vnet)#.

The list of commands whose inherited values can be overridden is provided in the table in the "Commands Whose Values Can be Inherited Or Overridden by a Virtual Network on an Interface" section in this module.

Example: Overriding Command Inheritance

In the following example, the OSPF cost of 30 for VRF blue overrides the OSPF cost of 20 for the other VRFs on the interface:

```
interface gigabitethernet 2/0/0
 vnet trunk
 ip address 10.1.1.1 255.255.255.0
 ! Set OSPF cost for all VRFs on this interface to 20.
 ip ospf cost 20
vnet name blue
 description Subinterface for VRF NG blue
 ! Set OSPF cost for blue to 30.
 ip ospf cost 30
```

The **show derived** command indicates the subinterface changed to a cost of 30:

```
Router(config-if-vnet)# do show derived | s interface GigabitEthernet2/0/0
```

```
interface GigabitEthernet2/0/0
vnet trunk
ip address 10.1.1.1 255.255.255.0
ip ospf cost 20
interface GigabitEthernet2/0/0.200
description Subinterface for VRF NG blue
vrf forwarding blue
ip address 10.1.1.1 255.255.255.0
ip ospf cost 30
Router(config-if-vnet)#
```

Example: Enabling an Attribute to vnet Global Only

Similarly, you might want to enable an attribute to vnet global only. To do so, use the **vnet global** interface submode, as follows:

```
interface gigabitethernet1/1/1
 vnet trunk
 ip address 10.1.2.1 255.255.255.0
 vnet global
  ! Set OSPF cost for global to 40.
  ip ospf cost 40
```

In this example, a user wants an EIGRP interface attribute set for all EVNs except vnet global. All EVNs inherit a hold time of 20 seconds, except vnet global, which overrides 20 with a hold time of 40 seconds.

```
interface fastethernet 1/0/0
 vnet trunk
 ip address 10.1.3.1 255.255.255.0
 ip hold-time eigrp 1 20
 vnet global
 ip hold-time eigrp 1 40
```

Removing Overrides and Restoring Values Inherited from EVN Trunk

The **no** and **default** keywords result in different outcomes, depending on whether they are used for a trunk interface or in virtual network interface mode. This section describes the different outcomes.

- When the **no** or **default** keyword is entered before a command on a trunk interface, the trunk is restored to the system's default value for that command. (This is standard behavior resulting for the **no** or **default** keyword).
- When the **default** keyword is entered before a command in virtual network interface mode, the override value is removed and the value that is inherited from the trunk is restored. The override value for the specific EVN is no longer in effect.

In the following example, the trunk interface is configured with an OSPF cost of 20, but VRF blue overrides that value with an OSPF cost of 30:

```
interface gigabitethernet 2/0/0
 vnet trunk
 ip address 10.1.1.1 255.255.255.0
 ! Set OSPF cost for all VRFs on this interface to 20.
 ip ospf cost 20
vnet name blue
 ! Set OSPF cost for blue to 30.
 ip ospf cost 30
```

When the following commands are entered, the OSPF cost value is restored to 20, which is the cost inherited from the trunk interface. (Note that 20 is not the default value of the **ip ospf cost** command.)

```
Router(config-if)# vnet name blue
Router(config-if-vnet)# default ip ospf cost
```

The **default** keyword entered before a command in virtual network interface mode restores the default state, but the **no** keyword does not always do that. In the following example, **no ip dampening-change eigrp 1** disables dampening change.

```
interface Ethernet1/1
 vnet trunk
 ip dampening-change eigrp 1 50
 shutdown
 vnet name red
  no ip dampening-change eigrp 1
! Make sure vnet red does NOT have dampening change enabled, regardless of trunk setting.
!
```

Determining if No Form of Command Appears in Configuration File

If a command is the type of command that switches a feature on or off, the **no** form of the command will appear in the configuration file when configured. That is, nonvolatile generation (NVGEN) overrides the setting from the EVN trunk, as shown in the following example:

```
interface gigabitethernet 2/0/0
 vnet trunk
 ip access-group 1 in
 vnet name red
  no ip pim sparse-mode
  no ip route-cache cef
  no ip access-group in
 vnet global
  ip ospf cost 100
```

If a command takes an argument in its syntax, such as **ip ospf cost** *cost* , the **no** form of the command will remove the configuration, but does not appear in the configuration file. That is, it will not be NVGEN'ed because the user could enter **ip ospf cost** *default-value* to override the inherited value in a more direct way.

EXEC Commands Routing Context

There may be occasions when you want to issue several EXEC commands to apply to a single EVN. In order to reduce the repetitive entering of VRF names for multiple EXEC commands, the **routing-context vrf** command allows you to set the VRF context of EXEC commands once, and then proceed using EXEC commands.

The table below shows four EXEC commands without routing context and in routing context. Note that in the left column, each EXEC command must identify the VRF. In the right column, the VRF content is identified once and the prompt changes to reflect that VRF; there is no need to identify the VRF in each command.

Table 152: EXEC Commands Routing Context

EXEC Commands Without Routing Context	EXEC Commands Routing Context
—	`Router# routing-context vrf red` `Router%red#`
`Router# show ip route vrf red` [Routing table output for VRF red]	`Router%red# show ip route` [Routing table output for VRF red]
`Router# ping vrf red 10.1.1.1` [Ping result using VRF red]	`Router%red# ping 10.1.1.1` [Ping result using VRF red]
`Router# telnet 10.1.1.1 /vrf red` [Telnet to 10.1.1.1 in VRF red]	`Router%red# telnet 10.1.1.1` [Telnet to 10.1.1.1 in VRF red]
`Router# traceroute vrf red 10.1.1.1` [Traceroute output in VRF red]	`Router%red# traceroute 10.1.1.1` [Traceroute output in VRF red]

EVN Compatibility with VRF-Lite

EVN is wire compatible with VRF-Lite. In other words, on the outside, 802.1q, SNMP MIBs, and all the EVN infrastructure will look exactly the same as VRF-Lite.

In the figure below, both routers have VRFs defined. The router on the left uses VRF-Lite, and the router on the right uses an EVN trunk with tags. The two configurations follow the figure.

VRF-Lite Subinterface Configuration

```
interface TenGigabitEthernet1/1/1
 ip address 10.122.5.31 255.255.255.254
 ip pim query-interval 333 msec
 ip pim sparse-mode
 logging event link-status
interface TenGigabitEthernet1/1/1.101
 description Subinterface for Red VRF
 encapsulation dot1Q 101
 ip vrf forwarding Red
 ip address 10.122.5.31 255.255.255.254
 ip pim query-interval 333 msec
 ip pim sparse-mode
```

EVN Trunk Configuration

```
interface TenGigabitEthernet 1/1/1
 vnet trunk
 ip address 10.122.5.32 255.255.255.254
 pim sparse-mode
 logging event link-status
Global Configuration:
vrf definition red
 vnet tag 101

vrf definition green
 vnet tag 102
```

```
 logging event subif-link-status
interface TenGigabitEthernet1/1/1.102
 description Subinterface for Green VRF
 encapsulation dot1Q 102
 ip vrf forwarding Green
 ip address 10.122.5.31 255.255.255.254
 ip pim query-interval 333 msec
 ip pim sparse-mode
 logging event subif-link-status
```

Multiaddress Family VRF Structure

Prior to Cisco IOS Releases 12.2(33)SB and 15.0(1)M, the CLI for a VRF applied to only one address family at a time. For example, the **ip vrf blue** command applies only to the IPv4 address family.

In Cisco IOS Releases 12.2(33)SB and 15.0(1)M, the CLI for a VRF applies to multiple address families under the same VRF. This is known as multiprotocol VRF. For example, the **vrf definition blue** command applies to IPv4 and IPv6 VPNs at the same time, but the routing tables for the two protocols are still different.

Note In Cisco IOS XE Release 3.2S, virtual networks do not support IPv6 except in **vnet global**.

QoS Functionality with EVN

Quality of Service (QoS) configurations are applied to the main physical interface on an EVN trunk. The QoS policy affects all traffic that flows out the physical interface in all the VRFs at the same time. In other words, QoS and network virtualization are mutually independent. For example, traffic marked with the DSCP value specified for voice will be put into the voice queue if the packet is from the red VRF, blue VRF, or green VRF. The traffic for all the VRFs will be queued together.

Commands Whose Values Can be Inherited Or Overridden by a Virtual Network on an Interface

As explained in the "Command Inheritance on EVN Trunk Interfaces" section, there are interface commands that are defined once for a trunk interface, and the value is inherited by each EVN sharing the interface. These commands are sometimes referred to as trunk commands.

A subset of the trunk commands are commands whose values can be overridden by specifying the command in virtual network interface mode. This is explained in the "Overriding Command Inheritance Virtual Network Interface Mode" section.

The table below lists interface commands and indicates whether the values are inherited by the EVNs on the interface and whether the commands can be overridden for a specific EVN.

Table 153: Interface Command Values Inherited or Overridden by a Virtual Network on an Interface

	Values Inherited by EVNs on Interface?	Values Can Be Overriden in Virtual Network Interface Mode?
IP Commands		

	Values Inherited by EVNs on Interface?	Values Can Be Overriden in Virtual Network Interface Mode?
ip accounting	Yes	No
ip address	Yes	No
ip broadcast-address	Yes	No
ip directed broadcast	Yes	No
ip information-reply	Yes	No
ip irdp	Yes	No
ip load-sharing	Yes	No
ip mask-reply	Yes	No
ip mtu	Yes	No
ip proxy-arp	Yes	No
ip redirects	Yes	No
ip unnumbered	Yes	No
ip unreachables	Yes	No
EIGRP Commands		
ip authentication key-chain eigrp	Yes	Yes
ip authentication mode eigrp	Yes	Yes
ip bandwidth-percent eigrp	Yes	Yes
ip dampening-change eigrp	Yes	Yes
ip dampening-interval eigrp	Yes	Yes
ip hello-interval eigrp	Yes	Yes
ip hold-time eigrp	Yes	Yes
ip next-hop-self eigrp	Yes	Yes
ip split-horizon eigrp	Yes	Yes
ip summary-address eigrp	Yes	Yes
Commands that Affect how EIGRP Determines Cost for an Interface		
bandwidth (interface)	Yes	Yes

	Values Inherited by EVNs on Interface?	Values Can Be Overriden in Virtual Network Interface Mode?
delay (interface)	Yes	Yes
OSPF Commands		
ip ospf *process-id* **area**	No	Yes
ip ospf authentication	Yes	Yes
ip ospf authentication-key	Yes	Yes
ip ospf bfd	Yes	Yes
ip ospf cost	Yes	Yes
ip ospf database-filter	Yes	Yes
ip ospf dead-interval	Yes	Yes
ip ospf demand-circuit	Yes	Yes
ip ospf flood-reduction	Yes	Yes
ip ospf hello-interval	Yes	Yes
ip ospf lls	Yes	Yes
ip ospf message-digest-key	Yes	Yes
ip ospf mtu-ignore	Yes	Yes
ip ospf network	Yes	Yes
ip ospf priority	Yes	Yes
ip ospf resync-timeout	Yes	Yes
ip ospf shutdown	Yes	Yes
ip ospf transmit-delay	Yes	Yes
ip ospf transmit-interval	Yes	Yes
ip ospf ttl-security	Yes	Yes
ip ospf vnet area	No	No
IP Multicast Commands		
ip igmp access-group	Yes	Yes
ip igmp explicit-tracking	Yes	Yes

	Values Inherited by EVNs on Interface?	Values Can Be Overriden in Virtual Network Interface Mode?
ip igmp helper-address	Yes	Yes
ip igmp immediate-leave	Yes	Yes
ip igmp last-member-query-count	Yes	Yes
ip igmp last-member-query-interval	Yes	Yes
ip igmp limit	Yes	Yes
ip igmp mroute-proxy	Yes	Yes
ip igmp proxy-service	Yes	Yes
ip igmp querier-timeout	Yes	Yes
ip igmp query-interval	Yes	Yes
ip igmp query-max-response-time	Yes	Yes
ip igmp tcn	Yes	Yes
ip igmp unidirectional-link	Yes	Yes
ip igmp v3lite	Yes	Yes
ip igmp version	Yes	Yes
ip multicast boundary	Yes	Yes
ip pim bidir-neighbor-filter	Yes	Yes
ip pim bsr-border	Yes	Yes
ip pim dense-mode	Yes	Yes
ip pim dr-priority	Yes	Yes
ip pim nbma-mode	Yes	Yes
ip pim neighbor-filter	Yes	Yes
ip pim passive	Yes	Yes
ip pim query-interval	Yes	Yes
ip pim sparse-dense-mode	Yes	Yes
ip pim sparse-mode	Yes	Yes
ip pim state-refresh	Yes	Yes

	Values Inherited by EVNs on Interface?	Values Can Be Overriden in Virtual Network Interface Mode?
Multicast Forwarding Information Base (MFIB) Commands		
ip mfib cef	Yes	Yes
ip mfib forwarding	Yes	Yes

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Easy Virtual Network commands	Easy Virtual Network Command Reference
Configuring Easy Virtual Network	"Configuring Easy Virtual Network" module in the *Easy Virtual Network Configuration Guide*
Configuring Easy Virtual Network shared services and route replication	"Configuring Easy Virtual Network Shared Services" module in the *Easy Virtual Network Configuration Guide*
Easy Virtual Network management and troubleshooting	"Easy Virtual Network Management and Troubleshooting" module in the *Easy Virtual Network Configuration Guide*

MIBs

MIB	MIBs Link
Any MIB that gives VRF information will continue to work with Easy Virtual Network. VRF-independent MIBs report information on every VRF in a system. • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Easy Virtual Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 154: Feature Information for Overview of Easy VIrtual Network

Feature Name	Releases	Feature Information
EVN VNET Trunk	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	Easy Virtual Network is an IP-based virtualization technology that provides end-to-end virtualization of the network. You can use a single IP infrastructure to provide separate virtual networks with isolated traffic paths.

CHAPTER 112

Configuring Easy Virtual Network

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end network virtualization. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other. Configure Easy Virtual Network to configure two or more virtual IP networks.

Prerequisites for Configuring EVN

- Implementing EVN in a network requires a single IP infrastructure that you use to create two or more virtual networks. You want path isolation for traffic on the different virtual networks.
- You should understand the concepts in the “Overview of Easy Virtual Network” module.
- We recommend that you draw your network topology, indicating the interfaces on each router that belong to the EVNs. The diagram facilitates tracking the interfaces you are configuring as edge interfaces and the interfaces you are configuring as trunk interfaces.

How to Configure EVN

Configuring an Easy Virtual Network Trunk Interface

Perform this task to configure an EVN trunk interface, which connects routers to provide the core to transport traffic for multiple virtual networks. Traffic carried over a trunk interface is tagged. This task illustrates how to configure a trunk interface with a base virtual routing and forwarding (VRF) and two named VRFs: VRF red and VRF blue.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **vrf definition** *vrf-name*
4. **vnet tag** *number*
5. **description** *string*
6. **address-family ipv4**
7. **exit-address-family**
8. **exit**
9. **vrf definition** *vrf-name*
10. **vnet tag** *number*
11. **description** *string*
12. **address-family ipv4**
13. **exit-address-family**
14. **exit**
15. **interface** *type number*
16. **ip address** *ip-address mask*
17. **vnet trunk** [**list** *vrf-list-name*]
18. **vnet name** *vrf-name*
19. **exit-if-vnet**
20. **no shutdown**
21. **exit**
22. **router ospf** *process-id*
23. **network** *ip-address wildcard* **area** *area-id*
24. **exit**
25. **router ospf** *process-id* **vrf** *vrf-name*
26. **network** *ip-address wildcard* **area** *area-id*
27. **exit**
28. **router ospf** *process-id* **vrf** *vrf-name*
29. **network** *ip-address wildcard* **area** *area-id*
30. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition red`	Configures a VRF routing table instance and enters VRF configuration mode.

	Command or Action	Purpose
Step 4	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 100`	Specifies the global numeric tag for the VRF. • The same tag number must be configured for the same virtual network on each edge and trunk interface. • When configuring EVN on a Cisco Catalyst 6500 family networking device, we recommend you assign a vnet tag number in the range 2 to 1000.
Step 5	**description** *string* **Example:** `Router(config-vrf)# description guest access`	(Optional) Describes a VRF to help a network administrator review the configuration files.
Step 6	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IP version 4 address prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.
Step 9	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition blue`	Configures a VRF routing table instance and enters VRF configuration mode.
Step 10	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 200`	Specifies the global numeric tag for the VRF. • The same tag number must be configured for the same VRF on each edge and trunk interface.
Step 11	**description** *string* **Example:** `Router(config-vrf) description Finance`	(Optional) Describes a VRF to help a network administrator review configuration files.
Step 12	**address-family ipv4** **Example:** `Router(config-vrf) address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.

	Command or Action	Purpose
Step 13	**exit-address-family** **Example:** `Router(config-vrf-af) exit-address-family`	Exits address family configuration mode.
Step 14	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.
Step 15	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/1/1`	Configures an interface type and enters interface configuration mode.
Step 16	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for the interface.
Step 17	**vnet trunk** [**list** *vrf-list-name*] **Example:** `Router(config-if)# vnet trunk`	Defines a trunk interface. • By default, all VRFs defined with the **vrf definition** command run on all trunk interfaces on the router. Therefore, VRF red and VRF blue are now running on this interface. • Use the **list** *vrf-list-name* command elements to restrict VRFs running on a trunk interface.
Step 18	**vnet name** *vrf-name* **Example:** `Router(config-if)# vnet name red`	(Optional) Enters virtual network interface mode to configure features that apply to a specified VRF to override global VRF values. • This step is not necessary if the global settings are acceptable for all of the VRFs on the interface. • After this step, you configure one or more eligible commands, such as **ip ospf cost**. (Not shown in this task.) For the list of commands that are used to override global VRF values, see Overview of Easy Virtual Network module, Table 2.
Step 19	**exit-if-vnet** **Example:** `Router(config-if-vnet) exit-if-vnet`	Exits VRF interface configuration mode and enters interface configuration mode.

	Command or Action	Purpose
Step 20	**no shutdown** **Example:** `Router(config-if) no shutdown`	Restarts an interface.
Step 21	**exit** **Example:** `Router(config-if) exit`	Exits to global configuration mode.
Step 22	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an Open Shortest Path First (OSPF) routing process and associates it with a VRF. • This OSPF instance has no VRF, so it is **vnet global**.
Step 23	**network** *ip-address wildcard* **area** *area-id* **Example:** `Router(config-router) network 10.0.0.0 255.255.255.0 area 0`	Defines the interfaces and associated area IDs on which OSPF runs.
Step 24	**exit** **Example:** `Router(config-router) exit`	Exits to global configuration mode.
Step 25	**router ospf** *process-id* **vrf** *vrf-name* **Example:** `Router(config)# router ospf 2 vrf red`	Configures an OSPF routing process and associates it with a VRF. • Specifies a different *process-id* for each VRF because they each need their own OSPF instance.
Step 26	**network** *ip-address wildcard* **area** *area-id* **Example:** `Router(config-router) network 10.0.0.0 255.255.255.0 area 0`	Defines the interfaces and associated area IDs on which OSPF runs and the area ID for those interfaces.
Step 27	**exit** **Example:** `Router(config-router) exit`	Exits to global configuration mode.
Step 28	**router ospf** *process-id* **vrf** *vrf-name* **Example:** `Router(config)# router ospf 3 vrf blue`	Configures an OSPF routing process and associates it with a VRF. • Specifies a different *process-id* for each VRF because they each need their own OSPF instance.

	Command or Action	Purpose
Step 29	**network** *ip-address wildcard* **area** *area-id* **Example:** `Router(config-router) network 10.0.0.0 255.255.255.0 area 2`	Defines the interfaces and associated area IDs on which OSPF runs and the area ID for those interfaces.
Step 30	**end** **Example:** `Router(config-vrf) end`	Ends the configuration session and returns to privileged EXEC mode.

Enabling a Subset of VRFs over a Trunk Interface

The prior task, "Configuring an Easy Virtual Network Trunk Interface," shows how to configure a trunk interface with two VRFs. By default, the trunk interfaces on a router can carry traffic for each VRF defined by the **vrf definition** command. However, you might want to enable only a subset of VRFs over a trunk interface, which is done by creating a VRF list. A maximum of 32 VRF lists can exist on a router. Perform the following task to create a VRF list. This task presumes that the VRF has already been configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf list** *vrf-list-name*
4. **member** *vrf-name*
5. Repeat Step 4 to add other VRFs to the list.
6. **exit-vrf-list**
7. **interface** *type number*
8. **vnet trunk list** *vrf-list-name*
9. **ip address** *ip-address mask*
10. **end**
11. **show vrf list** [*vrf-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**vrf list** *vrf-list-name* **Example:** `Router(config)# vrf list External`	Defines a list of VRFs and enters VRF list configuration mode. • The *vrf-list-name* argument may contain up to 32 characters. Quotation marks, spaces, and * are not allowed.
Step 4	**member** *vrf-name* **Example:** `Router(config-vrf-list)# member blue`	Specifies an existing VRF as a member of a VRF list. • The VRF must be defined before it can be added to a list.
Step 5	Repeat Step 4 to add other VRFs to the list.	(Optional) If you want a trunk interface with one VRF, your list only needs one VRF.
Step 6	**exit-vrf-list** **Example:** `Router(config-vrf-list)# exit-vrf-list`	Exits VRF list configuration mode.
Step 7	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/1/1`	Configures an interface and enters interface configuration mode.
Step 8	**vnet trunk list** *vrf-list-name* **Example:** `Router(config-if)# vnet trunk list mylist`	Defines a trunk interface and enables the VRFs that are in the VRF list. • Use the *vrf-list-name* defined in Step 3.
Step 9	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.3.1 255.255.255.0`	Sets a primary IP address for the interface.
Step 10	**end** **Example:** `Router(config-if) end`	Ends the configuration session and returns to privileged EXEC mode.
Step 11	**show vrf list** [*vrf-list-name*] **Example:** `Router# show vrf list mylist`	Displays information about a VRF list.

Configuring an EVN Edge Interface

Perform this task to configure an edge interface, which connects a user device to a virtual network. Traffic carried over an edge interface is untagged. The edge interface determines which virtual network the received traffic belongs to. Each edge interface is mapped to only one virtual network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding red`	Defines an edge interface and determines the VRF that the incoming traffic belongs to. • The *vrf-name* must already be defined by a **vrf definition** command. • In this example, incoming traffic belongs to VRF red. **Note** Make sure you are not on the trunk interface when you are trying to configure an edge interface.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for the interface.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if) end`	Ends the configuration session and returns to privileged EXEC mode.

What to Do Next

After you have configured an edge interface and a trunk interface, refer to your network diagram and log on to a different router. If it is has an edge interface, configure that interface. If it has a trunk interface, configure that interface with the appropriate VRFs. Continue configuring each of the routers and interfaces that belong to each VRF.

Configure other protocol features you want running in your VRFs. See the appropriate IP Routing configuration guide.

Verifying EVN Configurations

Perform any of the following steps in this task to verify your configuration. Because a virtual network is a VRF, all the existing VRF **show** commands are supported for virtual networks. If a router has a mix of VRFs and virtual networks, the various **show vrf** commands will include both VRFs and virtual networks in the output.

SUMMARY STEPS

1. **enable**
2. **show vnet tag**
3. **show running-config** [**vrf** | **vnet**] [*vrf-name*]
4. **show vrf list** [*vrf-list-name*]
5. **show** {**vrf** | **vnet**} [**ipv4** | **ipv6**] [**interface** | **brief** | **detail** | **lock**] [*vrf-name*]
6. **show** {**vrf** | **vnet**} **counters**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show vnet tag** **Example:** `Router# show vnet tag`	(Optional) Displays where each tag has been configured or used.

	Command or Action	Purpose
Step 3	**show running-config** [**vrf** | **vnet**] [*vrf-name*] **Example:** `Router# show running-config vrf green`	(Optional) Displays the VRFs in the running configuration, displays the interfaces in the VRFs, and displays the protocol configurations for Multi-VRF.
Step 4	**show vrf list** [*vrf-list-name*] **Example:** `Router# show vrf list`	(Optional) Displays information about VRF lists, such as the VRFs in each list.
Step 5	**show** {**vrf** | **vnet**} [**ipv4** | **ipv6**] [**interface** | **brief** | **detail** | **lock**] [*vrf-name*] **Example:** `Router# show vnet detail`	(Optional) Displays information about the VRFs.
Step 6	**show** {**vrf** | **vnet**} **counters** **Example:** `Router# show vnet counters`	(Optional) Displays information about the number of VRFs or virtual networks supported and configured.

Configuration Examples for Configuring EVN

Example: Virtual Networks Using OSPF with network Commands

In this example, **network** commands associate a shared VRF interface with a base VRF and two named VRFs, red and blue. There are three OSPF instances because each VRF needs its own OSPF instance. OSPF 1 has no VRF, so it is **vnet global**.

```
vrf definition red
 vnet tag 100
 address-family ipv4
 exit-address-family
!
vrf definition blue
 vnet tag 200
 address-family ipv4
 exit-address-family
!
interface gigabitethernet 0/0/0
 ip address 10.0.0.1 255.255.255.0
 vnet trunk
 vnet name red
  ip ospf cost 100
!
router ospf 1
 log-adjacency-changes detail
 network 10.0.0.0 255.255.255.0 area 0
router ospf 2 vrf red
```

```
 log-adjacency-changes
 network 10.0.0.0 255.255.255.0 area 0
router ospf 3 vrf blue
 log-adjacency-changes
 network 10.0.0.0 255.255.255.0 area 2
```

Example: Virtual Networks Using OSPF with ip ospf vnet area Command

This example differs from the prior example regarding the association between OSPF instances and a particular interface. In this example, OSPF is running on all of the virtual networks of a trunk interface. The **ip ospf vnet area** command associates the GigabitEthernet 0/0/0 interface with the three OSPF instances.

```
vrf definition red
 vnet tag 100
 address-family ipv4
 exit-address-family
!
vrf definition blue
 vnet tag 200
 address-family ipv4
 exit-address-family
!
interface gigabitethernet 0/0/0
 ip address 10.0.0.1 255.255.255.0
 vnet trunk
 ip ospf vnet area 0
 vnet name red
  ip ospf cost 100
 vnet name blue
  ip ospf 3 area 2
!
router ospf 1
 log-adjacency-changes detail
router ospf 2 vrf red
 log-adjacency-changes
router ospf 3 vrf blue
 log-adjacency-changes
```

Example: Command Inheritance and Virtual Network Interface Mode Override in an EIGRP Environment

This example shows a GigabitEthernet interface configured with various EIGRP commands:

```
interface gigabitethernet0/0/0
 vnet trunk
 ip address 10.0.0.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 ip summary-address eigrp 1 1.0.0.0 255.0.0.0
end
```

Because a trunk is configured, a VRF subinterface is automatically created and the commands on the main interface are inherited by the VRF subinterface (g0/0/0.3, where the number 3 is the tag number from vnet tag 3 .)

```
R1# show running-config vrf red
Building configuration...
Current configuration : 1072 bytes
vrf definition red
 vnet tag 3
 !
 address-family ipv4
 exit-address-family
 !
```

If you display that hidden subinterface with the **show derived-config** command, you'll see that all of the commands entered on GigabitEthernet 0/0/0 have been inherited by GigabitEthernet 0/0/0.3:

```
R1# show derived-config interface gigabitethernet0/0/0.3
Building configuration...
Derived configuration : 478 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET red
 vrf forwarding red
 encapsulation dot1Q 3
 ip address 10.0.0.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 ip summary-address eigrp 1 1.0.0.0 255.0.0.0
end
```

You can override those commands by using virtual network interface mode (under the **vnet name** command). For example:

```
R1(config)# interface gigabitethernet0/0/0
R1(config-if)# vnet name red
R1(config-if-vnet)# no ip authentication mode eigrp 1 md5
 ! disable authen for e0/0.3 only
R1(config-if-vnet)# ip authentication key-chain eigrp 1 y
 ! different key-chain
R1(config-if-vnet)# ip band eigrp 1 99
 ! higher bandwidth-percent
R1(config-if-vnet)# no ip dampening-change eigrp 1
 ! disable dampening-change
R1(config-if-vnet)# ip hello eigrp 1 7
R1(config-if-vnet)# ip hold eigrp 1 21
R1(config-if-vnet)# ip next-hop-self eigrp 1
 ! enable next-hop-self for e0/0.3
R1(config-if-vnet)# ip split-horizon eigrp 1
 ! enable split-horizon
R1(config-if-vnet)# no ip summary-address eigrp 1 10.0.0.1 255.0.0.0
 ! do not summarize on e0/0.3

R1(config-if-vnet)# do show running-config interface gigabitethernet0/0/0
```

```
Building configuration...
Current configuration : 731 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 1.1.1.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 ip summary-address eigrp 1 1.0.0.0 255.0.0.0
 vnet name red
  ip split-horizon eigrp 1
  no ip summary-address eigrp 1 1.0.0.0 255.0.0.0
  no ip authentication mode eigrp 1 md5
  ip authentication key-chain eigrp 1 y
  ip bandwidth-percent eigrp 1 99
  no ip dampening-change eigrp 1
  ip hello-interval eigrp 1 7
  ip hold-time eigrp 1 21
  ip next-hop-self eigrp 1
 !
end
```

Notice that g0/0.3 is now using the override settings:

```
R1(config-if-vnet)# do show derived-config interface g0/0.3

Building configuration...
Derived configuration : 479 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET red
 vrf forwarding red
 encapsulation dot1Q 3
 ip address 1.1.1.1 255.255.255.0
 no ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 y
 ip bandwidth-percent eigrp 1 99
 no ip dampening-change eigrp 1
 ip hello-interval eigrp 1 7
 ip hold-time eigrp 1 21
 ip next-hop-self eigrp 1
 ip split-horizon eigrp 1
 no ip summary-address eigrp 1 1.0.0.0 255.0.0.0
end
```

Commands entered in **vnet name** submode are sticky. That is, when you enter a command in **vnet name** submode, it will nvgen, regardless of whether it is set to the same value as the default value. For example, the default hello value is 5. When the **ip hello eigrp** command is entered in **vnet name** submode, it will nvgen; it does not do that in any other mode.

```
R1(config-if)# interface gigabitethernet0/0/2
R1(config-if)# vnet trunk
R1(config-if)# ip bandwidth-percent eigrp 1 50         <---<< this will NOT nvgen
R1(config-if)# ip hello eigrp 1 5              <---<< this will NOT nvgen
R1(config-if)# no ip authentication mode eigrp 1 md5    <---<< this will NOT nvgen
```

```
R1(config-if)# vnet name red
R1(config-if-vnet)# ip bandwidth-percent eigrp 1 50   <---<< this will nvgen
R1(config-if-vnet)# ip hello eigrp 1 5   <---<< this will nvgen
R1(config-if-vnet)# no ip authentication mode eigrp 1 md5  <---<< this will nvgen
R1(config-if-vnet)# do show running-config interface gigabitethernet0/0/2

Building configuration...
Current configuration : 104 bytes
!
interface GigabitEthernet0/0/2
 vnet trunk
 no ip address
 vnet name red
  ip bandwidth-percent eigrp 1 50
  ip hello-interval eigrp 1 5
  no ip authentication mode eigrp 1 md5
 !
```

Because of this sticky factor, to remove a configuration entry in **vnet name** submode, you typically must use the default form of that command. Some commands can also be removed using the **no** form; it depends on the command. Some commands use the **no** form to disable the command instead, such as the **authentication** and **summary-address** commands.

```
R1(config-if-vnet)# default ip authentication mode eigrp 1 md5
R1(config-if-vnet)# no ip bandwidth-percent eigrp 1
R1(config-if-vnet)# no ip hello eigrp 1

R1(config-if-vnet)# do show running-config interface g0/2

Building configuration...
Current configuration : 138 bytes
!
interface GigabitEthernet0/0/2
 vnet trunk
 no ip address
 vnet name red
 !
end
```

Example: Command Inheritance and Virtual Network Interface Mode Override in a Multicast Environment

The following example illustrates command inheritance and virtual network interface mode override in a multicast network. A trunk interface leverages the fact that configuration requirements from different VRFs will be similar over the same trunk interface. Eligible commands configured on the trunk interface are inherited by all VRFs running over the same interface.

In this example, IP multicast (PIM sparse mode) is configured on the trunk interface, which has several VRFs:

```
vrf definition red
 vnet tag 13
 !
 address-family ipv4
 exit-address-family
!
ip multicast-routing
ip multicast-routing vrf red
interface GigabitEthernet0/1/0
 vnet trunk
```

```
 ip address 125.1.15.18 255.255.255.0
 ip pim sparse-mode
```

The user decides that he does not want IP multicast configured for VRF red on GigabitEthernet 0/1/0, so he uses the virtual network interface mode override. IP Multicast is disabled for VRF red only. The **no ip pim** command disables all modes of Protocol Independent Multicast (PIM), including sparse mode, dense mode, and sparse-dense mode, for VRF red.

```
interface GigabitEthernet0/1/0
 vnet trunk
 ip address 125.1.15.18 255.255.255.0
 ip pim sparse-mode
 vnet name red
 no ip pim
```

Example: EVN Using IP Multicast

The following example configures PIM sparse mode and leverages Anycast RP for RP redundancy. In this example, only one VRF is configured.

The example shows how to enable multicast routing globally and on each L3 interface. The black text indicates the group of commands configuring the global table; the red text indicates the group of commands configuring VRF red.

```
ip multicast-routing
interface GigabitEthernet 1/1/1
 description GigabitEthernet to core (Global)                    GLOBAL TABLE
 ip pim sparse-mode
vrf definition red
 vnet tag 100
!
 address-family ipv4
 exit-address-family
!
ip multicast-routing vrf red                                      VRF RED
!
interface gigabitethernet1/1/1.100
 description GigabitEthernet to core (VRF red)
 vrf forwarding red
 ip pim sparse-mode
```

Configure the RP in the VRF using Anycast RP.

```
interface loopback0
 description Anycast RP Global
 ip address 10.122.5.200 255.255.255.255
 ip pim sparse-mode
!
interface loopback1
 description MDSP Peering interface
 ip address 10.122.5.250 255.255.255.255                       GLOBAL TABLE
 ip pim sparse-mode
!
ip msdp peer 10.122.5.251 connect-source loopback 1
ip msdp originator-id loopback 1
ip pim rp-address 10.122.5.200
access-list 10 permit 239.0.0.0 0.255.255.255
!
!
interface loopback 10
```

```
 description Anycast RP VRF Red
 vrf forwarding red
 ip address 10.122.15.200 255.255.255.255
 ip pim sparse-mode
interface loopback 11
 description MSDP Peering interface VRF red                    VRF RED
 vrf forwarding red
 ip address 10.122.15.250 255.255.255.255
 ip pim sparse-mode
!
ip msdp vrf red peer 10.122.15.251 connect-source loopback 11
ip msdp vrf red originator-id loopback 11
!
ip pim vrf red rp-address 10.122.15.200
access-list 11 permit 239.192.0.0 0.0.255.255
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Easy Virtual Network commands	Easy Virtual Network Command Reference
Information about Easy Virtual Network configuration tasks	"Overview of Easy Virtual Networks" module in the *Easy Virtual Network Configuration Guide*
Easy Virtual Network shared services and route replication configuration tasks	"Configuring Easy Virtual Network Shared Services" module in the *Easy Virtual Network Configuration Guide*
Easy Virtual Network management and troubleshooting	"Easy Virtual Network Management and Troubleshooting" module in the *Easy Virtual Network Configuration Guide*

MIBs

MIB	MIBs Link
Any MIB that gives VRF information will continue to work with EVN. VRF-independent MIBs report information on every VRF in a system. • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Easy Virtual Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 155: Feature Information for Configuring Easy Virtual Network

Feature Name	Releases	Feature Information
EVN VNET Trunk	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	This module describes how to configure virtual IP networks. An EVN is an IP-based virtualization technology that provides end-to-end virtualization of the network. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other. The following commands were modified: **vrf definition** , **vrf forwarding** . The following commands were introduced: **description** (vrf definition submode), **exit-if-vnet** , **exit-vrf-list** , **member** (vrf list), **routing-context** , **show running-config vnet** , **show vnet** , **show vnet counters** , **show vnet tag** , **show vrf counters** , **show vrf list** , **vnet** , **vnet tag** , **vnet trunk** , **vrf list** .
EVN OSPF	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	EVN OSPF provides Easy Virtual Network support for OSPF. The following commands were modified: **ip ospf database-filter all out** , **ip ospf demand-circuit** , **ip ospf flood-reduction** , **ip ospf mtu-ignore** , **ip ospf shutdown** . The following command was introduced: **ip ospf vnet area** .

Feature Name	Releases	Feature Information
EVN EIGRP	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	EVN EIGRP provides Easy Virtual Network support for EIGRP. The following commands were modified: **ip summary-address eigrp** , **summary-metric** .
EVN Multicast	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	EVN Multicast provides Easy Virtual Network support for IP Multicast.

CHAPTER 113

Easy Virtual Network Management and Troubleshooting

This module describes how to manage and troubleshoot Easy Virtual Network (EVN).

Prerequisites for EVN Management and Troubleshooting

- Read the "Overview of Easy Virtual Network" section and the "Configuring Easy Virtual Network" section, and implement EVN.

Information About EVN Management and Troubleshooting

Routing Context for EXEC Mode Reduces Repetitive VRF Specification

There may be occasions when you want to issue several EXEC commands to apply to a single virtual network. In order to reduce the repetitive entering of virtual routing and forwarding (VRF) names for multiple EXEC commands, the **routing-context vrf** command allows you to set the VRF context of such EXEC commands once, and then proceed using EXEC commands.

The table below shows four EXEC commands in Cisco IOS XE software without routing context and in routing context. Note that in the left column, each EXEC command must specify the VRF. In the right column, the VRF context is specified once and the prompt changes to reflect that VRF; there is no need to specify the VRF in each command.

Table 156: EXEC Commands Routing Context

EXEC Commands CLI without Routing Context	EXEC Routing Context
—	`Router# routing-context vrf red` `Router%red#`
`Router# show ip route vrf red` [Routing table output for VRF red]	`Router%red# show ip route` [Routing table output for VRF red]
`Router# ping vrf red 10.1.1.1` [Ping result using VRF red]	`Router%red# ping 10.1.1.1` [Ping result using VRF red]
`Router# telnet 10.1.1.1 /vrf red` [Telnet to 10.1.1.1 in VRF red]	`Router%red# telnet 10.1.1.1` [Telnet to 10.1.1.1 in VRF red]
`Router# traceroute vrf red 10.1.1.1` [Traceroute output in VRF red]	`Router%red# traceroute 10.1.1.1` [Traceroute output in VRF red]

Output of traceroute Command Indicates VRF Name and VRF Tag

Output of the **traceroute** command is enhanced to make troubleshooting easier by displaying the incoming VRF name/tag and the outgoing VRF name/tag, as shown in the following example:

```
Router# traceroute vrf red 10.0.10.12
Type escape sequence to abort.
Tracing the route to 10.0.10.12
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.13.15 (red/13,red/13) 0 msec
    10.1.16.16 (red/13,red/13) 0 msec
    10.1.13.15 (red/13,red/13) 1 msec
  2 10.1.8.13 (red/13,red/13) 0 msec
    10.1.7.13 (red/13,red/13) 0 msec
    10.1.8.13 (red/13,red/13) 0 msec
  3 10.1.2.11 (red/13,blue/10) 1 msec 0 msec 0 msec
  4  *  *  *
```

Debug Output Filtering Per VRF

Using EVN, you can filter debug output per VRF by using the **debug condition vrf** command. The following is sample output from the **debug condition vrf** command:

```
Router# debug condition vrf red

Condition 1 set
CEF filter table debugging is on
CEF filter table debugging is on
R1#
```

```
*Aug 19 23:06:38.178: vrfmgr(0) Debug: Condition 1, vrf red triggered, count 1
R1#
```

CISCO-VRF-MIB

EVN provides a CISCO-VRF-MIB for VRF discovery and management.

How to Manage and Troubleshoot EVN

Setting the Routing Context for EXEC Mode to a Specific VRF

To reduce the repeated entering of virtual routing and forwarding (VRF) names when you are issuing EXEC commands on a router, set the routing context of the EXEC commands once, and then proceed with entering them in any order. Perform this task to set the routing context for EXEC mode to a specific VRF, issue EXEC commands, and then restore the system to the global EXEC context.

SUMMARY STEPS

1. **enable**
2. **routing-context vrf** *vrf-name*
3. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **static download**]
4. **ping** [*protocol* [**tag**] {*host-name* | *system-address*}]
5. **telnet** *host* [*port*]
6. **traceroute** [**vrf** *vrf-name* | **topology** *topology-name*] [*protocol*] *destination*
7. **routing-context vrf global**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**routing-context vrf** *vrf-name* **Example:** `Router# routing-context vrf red`	Enters the routing context for EXEC mode to a specified VRF.
Step 3	**show ip route** [*ip-address* [*mask*] [**longer-prefixes**]	*protocol* [*process-id*]	**static download**] **Example:** `Router%red# show ip route`	(Optional) Displays the current state of the routing table. • The system prompt changes to reflect the target VRF. • This example shows the **show ip route** command issued within the context of vNET red. The routing table for vNET red would be displayed.

	Command or Action	Purpose
Step 4	**ping** [*protocol* [**tag**] {*host-name* | *system-address*}] **Example:** `Router%red# ping 10.1.1.1`	(Optional) Sends an echo request packet to an address. • This example shows the **ping** command issued within the context of vNET red. Ping results using vNET red would be displayed.
Step 5	**telnet** *host* [*port*] **Example:** `Router%red# telnet 10.1.1.1`	(Optional) Logs in to a host that supports Telnet.
Step 6	**traceroute** [**vrf** *vrf-name* | **topology** *topology-name*] [*protocol*] *destination* **Example:** `Router%red# traceroute 10.1.1.1`	(Optional) Displays the route that packets will take to the destination.

Step 7	**routing-context vrf global**
	Example:
	`Router%red# routing-context vrf global`
	Example:
	`Router>`
		(Optional) Restores the system to the global EXEC context.
		• The prompt returns to the user EXEC prompt.

Enabling Debug Output for VRFs

SUMMARY STEPS

1. **enable**
2. **debug vrf** {**create** | **delete** | **error** | **ha** | **initialization** | **interface** | **ipv4** | **ipv6** | **issu** | **lock** | **lookup** | **mpls** | **selection**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug vrf** {**create**	**delete**	**error**	**ha**	**initialization**	**interface**	**ipv4**	**ipv6**	**issu**	**lock**	**lookup**	**mpls**	**selection**} **Example:**	Displays VRF debugging information.

	Command or Action	Purpose
	`Router# debug vrf ipv4`	

Setting SNMP v2c Context for Virtual Networks

Perform this task to map an SNMP v2c context to a VRF. The following SNMP v2c configurations will then be done by the system automatically:

- Context creation (instead of the **snmp-server context** command), using the same name as the *context-name* entered in the **snmp context** command.
- Group creation (instead of the **snmp-server group** command), using the same name as the *community-name* entered in the **snmp context** command.
- Community creation (instead of the **snmp-server community** command), using the same name as the *community-name* entered in the **snmp context** command. The default permission is **ro** (read-only).
- Community context mapping (instead of the **snmp mib community-map** command).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv4**
5. **snmp context** *context-name* [**community** *community-name* [**rw** | **ro**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf1`	Defines a virtual routing and forwarding instance (VRF) and enters VRF configuration mode.
Step 4	**address-family ipv4** **Example:** `Device(config-vrf)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.

	Command or Action	Purpose
Step 5	**snmp context** *context-name* [**community** *community-name* [**rw** | **ro**]] **Example:** `Router(config-vrf)# snmp context xxx community yyy`	Sets the SNMP v2c context for the VRF. • The default is read-only (**ro**).

Setting SNMP v3 Context for Virtual Networks

Perform this task to map an SNMP v3 context to a virtual routing and forwarding (VRF). The following SNMP v3 configurations will then be done by the system automatically:

- Context creation (instead of the **snmp-server context** command), using the same name as the *context-name* entered in the **snmp context** command.
- Group creation (instead of the **snmp-server group** command). The group name will be generated by appending "_acnf" to the *context-name* entered in the **snmp context** command.
- User creation (instead of the **snmp-server user** command). The user will be created using the details configured in the **snmp context** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv4**
5. **snmp context** *context-name* [**user** *username* [**credential** | [**encrypted**] [**auth** {**md5** *password* | **sha** *password*}] [**access** {*access-list-number* | *access-list-name* | **ipv6** *access-list-name*}]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf1`	Defines a VRF and enters VRF configuration mode.

	Command or Action	Purpose
Step 4	**address-family ipv4** **Example:** `Device(config-vrf)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 5	**snmp context** *context-name* [**user** *username* [**credential** | [**encrypted**] [**auth** {**md5** *password* | **sha** *password*}] [**access** {*access-list-number* | *access-list-name* | **ipv6** *access-list-name*}]]] **Example:** `Router(config-vrf)# snmp context green_ctx user green_comm encyrpted`	Sets the SNMP v3 context for the VRF.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Easy Virtual Network commands	Easy Virtual Network Command Reference
Overview of Easy Virtual Network	"Overview of Easy Virtual Network" module in the *Easy Virtual Network Configuration Guide*
Configuring Easy Virtual Network	"Configuring Easy Virtual Network" module in the *Easy Virtual Network Configuration Guide*
Easy Virtual Network shared services and route replication	"Easy Virtual Network Shared Services" module in the *Easy Virtual Network Configuration Guide*

MIBs

MIB	MIBs Link
Any MIB that gives VRF information will continue to work with Easy Virtual Network. VRF-independent MIBs report information on every VRF in a system: • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for EVN Management and Troubleshooting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 157: Feature Information for EVN Management and Troubleshooting

Feature Name	Releases	Feature Information
EVN Cisco EVN MIB		EVN Cisco EVN MIB simplifies SNMP configuration. The following command was modified: **snmp context**. .
EVN Traceroute		EVN Traceroute enhances output of the **traceroute** command to display the VRF name and tag. The following command was modified: **traceroute**. .
EVN VNET Trunk		Users can filter debug output per VRF by using the **debug condition vrf** command. The following commands were introduced: **debug condition vrf** , **debug vrf** .

CHAPTER 114

Configuring Easy Virtual Network Shared Services

This chapter describes how to use route replication and redistribution to share services in an Easy Virtual Network (EVN).

Prerequisites for Virtual IP Network Shared Services

- Read the “Overview of Easy Virtual Networks” module.
- Implement EVN based on the “Configuring Easy Virtual Networks” module.

Restrictions for Virtual IP Network Shared Services

- Route replication is supported for Static, Enhanced Interior Gateway Routing Protocol (EIGRP), and Open Shortest Path First (OSPF) routes. It is not possible to replicate routes to and from Border Gateway Protocol (BGP), but that is not an issue because the BGP import and export method of copying routes between Virtual Routing and Forwarding (VRF) is available in a virtual network.
- Inter-vrf redistribution for OSPFv2 (ipv4) is supported, but, OSPFv3 under ipv4 address-family is not supported

Information About Easy Virtual Network Shared Services

Shared Services in an Easy Virtual Network

There are some common services (such as database and application servers) that multiple virtual networks need to access. Sharing these services are beneficial because:

- They are usually not duplicated for each group.
- It is economical, efficient, and manageable.
- Policies can be centrally deployed.

To achieve route separation, you could replicate the service, either physically or virtually, one service for each virtual network. However, that solution might not be cost effective or feasible. For a router that supports EVN, the solution is to perform route replication and route redistribution.

Route replication allows shared services because routes are replicated between virtual networks and clients who reside in one virtual network can reach prefixes that exist in another virtual network.

A shared services approach works best for Dynamic Name Systems (DNS), Dynamic Host Configuration Protocol (DHCP), and corporate communications. It is not a solution for sharing access to an Internet gateway.

Easy Virtual Network Shared Services Easier than VRF-Lite

Sharing servers in VRF-Lite requires route distinguishers (RDs), route targets with importing and exporting, and configuring BGP.

In an EVN environment, shared services are achieved with route replication, which is a simple deployment. Route replication requires no BGP, no RD, no route targets, and no import or export.

In summary, the BGP import and export method of copying routes between VRFs works with both VRF-Lite and EVN. However, route replication is the simpler alternative to enable sharing of common services across multiple virtual networks.

Route Replication Process in Easy Virtual Network

With shared services, clients and servers are located in different virtual networks. To achieve connectivity between clients and servers, routes must be exchanged among virtual networks. Depending on whether VRF-Lite or EVN is implemented, route exchanges among VRFs are accomplished in one of the following ways:

- If VRF-Lite is implemented, route leaking is achieved via BGP by using the route import/export feature.
- If EVN is implemented, route replication is supported directly by the Routing Information Base (RIB); there is no dependency on BGP. After routes are replicated from a different virtual network, those routes are propagated across each virtual network through existing redistribution into the Interior Gateway Protocol (IGP).

In the following route replication scenario, a router has two VRFs named Services and User-A. OSPF is configured:

```
router ospf 99 vrf services
 network 126.1.0.0 0.0.255.255 area 0
!
router ospf 98 vrf user-a
 network 126.1.0.0 0.0.255.255 area 0
```

Furthermore, route replication is configured for VRF User-A:

```
vrf definition user-a
 !
 address-family ipv4
 route-replicate from vrf services unicast ospf 99
 exit-address-family
```

In the scenario, the following RIB for the VRF Services contains four routes, three of which are replicated to the RIB for VRF User-A. Route replication creates a link to the source RIB, as shown in the figure below.

RIB—VRF Services

Route	Type	Destination Interface	Next Hop
126.1.17.0/24	Connected	Gi0/1	
126.1.9.0/24	OSPF	Gi0/1	126.1.17.13
126.1.12.0/24	OSPF	Gi0/1	126.1.17.13
126.1.14.0/24	OSPF	Gi0/1	126.1.17.13

RIB—VRF User-A

Route	Type	Destination Interface	Next Hop
126.1.9.0/24	OSPF	Gi0/1	126.1.17.13
126.1.12.0/24	OSPF	Gi0/1	126.1.17.13
126.1.14.0/24	OSPF	Gi0/1	126.1.17.13

Configuring route replication allows mutual redistribution between virtual IP networks. In the case of shared services, you configure route replication within the VRF that needs access to shared services. Within each **route-replicate** command, you can optionally filter out routes with a route map to prevent a routing loop. That is, you do not want to redistribute routes back into the original routing protocol. You do not want a native route to show up as a replicated route.

Where to Implement Route Replication

We recommend implementing route replication on the router as close to the shared service as possible. Ideally, the router that is directly connected to the server subnet should be used, to eliminate the need to redistribute the host prefixes on the server VRF, and, thereby, avoid a potential routing loop.

Route Replication Behavior for Easy Virtual Network

This section describes the behavior of route replication for EVN, which differs from the behavior for Multi-Topology Routing. In an EVN environment:

- The **route-replicate** command is accepted only under the **address-family ipv4** command, which is configured under the **vrf definition** command.
- The **route-replicate** command replicates routes into the base topology within the specified address family.
- If **all** is specified as a source protocol, only one **route-replicate** command is allowed per VRF for a given destination topology.
- The **no route-replicate** command is allowed to exclude a source protocol.
- If **all** is specified as a source protocol, then connected routes are replicated (unlike in the Multi-Topology Routing version of the **route-replicate** command).
- A replicated route inherits the administrative distance and source protocol of the source route.

Route Preference Rules After Route Replication in Easy Virtual Network

If a route is replicated, the following rule determines route preference:

- If two routes are owned by the same protocol and have the same source VRF, and if one of the routes is NOT replicated, then the nonreplicated route is preferred.

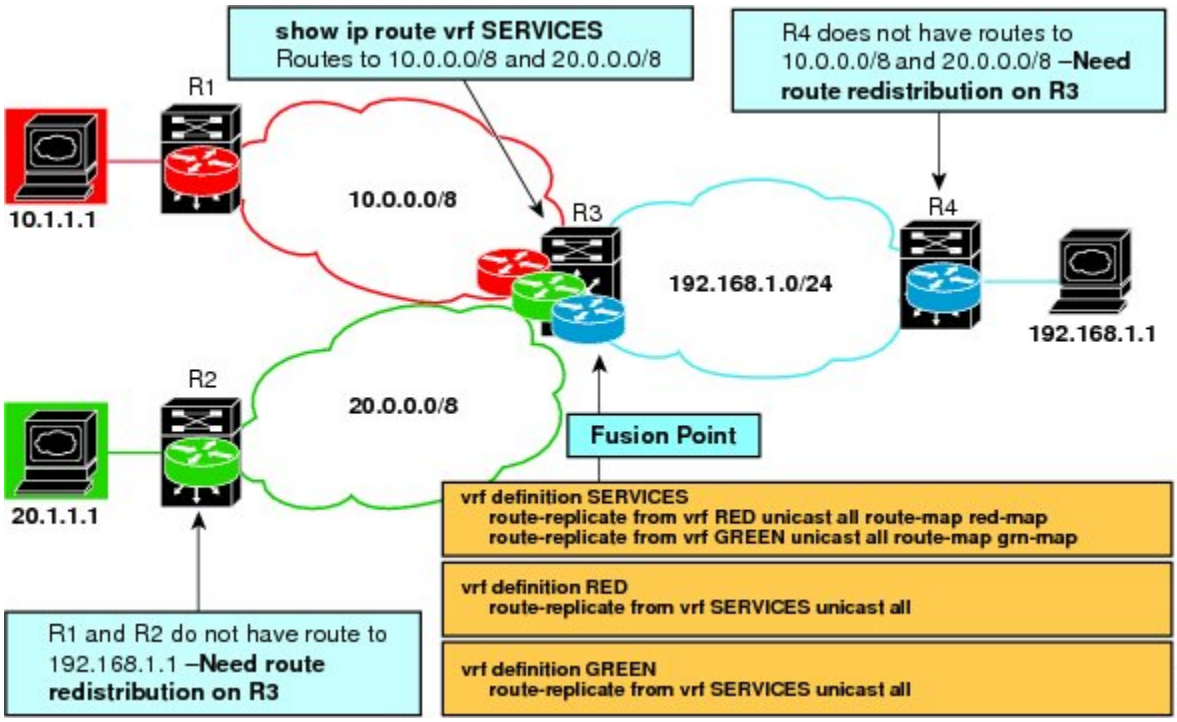
If the above rule does not apply, the following rules determine route preference, in this order:

1. Prefer the route with smaller administrative distance.
2. Prefer the route with smaller default administrative distance.
3. Prefer a non-replicated route over a replicated route.
4. Compare original vrf-names. Prefer the route with the lexicographically smaller vrf-name.
5. Compare original sub-address-families: Prefer unicast over multicast.
6. Prefer the oldest route.

How to Share Services Using Easy Virtual Network

Configuring Route Replication to Share Services in Easy Virtual Network

Perform this task to replicate routes from one VRF to another. The examples in the task table are based on the figure below.



In this particular task, routes from VRF SERVICES are replicated to both VRF RED and VRF GREEN, and VRF RED and VRF GREEN are not allowed to share routes between them. In order to allow bidirectional traffic, routes from VRF RED and VRF GREEN are also replicated to VRF SERVICES.

Note In a real EVN environment, there would also be route replication between VRF SERVICES and a third VRF, and maybe more VRFs. Such replication is left out of the following configuration task for the sake of brevity.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **vnet tag** *number*
5. **description** *string*
6. **address-family ipv4**
7. **exit**
8. **exit**
9. **vrf definition** *vrf-name*
10. **vnet tag** *number*
11. **description** *string*
12. **address-family ipv4**
13. **exit**
14. **exit**
15. **interface** *type number*

16. **vrf forwarding** *vrf-name*
17. **ip address** *ip-address mask*
18. **no shutdown**
19. **exit**
20. **router ospf** *process-id* **vrf** *vrf-name*
21. **network** *ip-address wildcard-mask* **area** *area-id*
22. **exit**
23. **router ospf** *process-id* [**vrf** *vrf-name*]
24. **network** *ip-address wildcard-mask* **area** *area-id*
25. **exit**
26. **vrf definition** *vrf-name*
27. **address-family ipv4**
28. **route-replicate from** [**vrf** *vrf-name*] {**multicast**| **unicast**} {**all**| *protocol-name*} [**route-map** *map-tag*]
29. **exit**
30. **exit**
31. **vrf definition** *vrf-name*
32. **address-family ipv4**
33. **route-replicate from** [**vrf** *vrf-name*] {**multicast**| **unicast**} {**all**| *protocol-name*} [**route-map** *map-tag*]
34. **end**
35. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition SERVICES`	Defines a VRF and enters VRF configuration mode.
Step 4	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 100`	Specifies the global, numeric tag for the VRF. • The same tag number must be configured for the same VRF on each edge and trunk interface.
Step 5	**description** *string* **Example:**	(Optional) Describes a VRF to help the network administrator looking at the configuration file.

	Command or Action	Purpose
	`Router(config-vrf)# description shared services`	
Step 6	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**exit** **Example:** `Router(config-vrf-af)# exit`	Exits to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.
Step 9	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition RED`	Defines a VRF and enters VRF configuration mode.
Step 10	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 200`	Specifies the global, numeric tag for the VRF. • The same tag number must be configured for the same VRF on each edge and trunk interface.
Step 11	**description** *string* **Example:** `Router(config-vrf)# description user of services`	(Optional) Describes a VRF to help the network administrator looking at the configuration file.
Step 12	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.
Step 13	**exit** **Example:** `Router(config-vrf-af)# exit`	Exits to VRF configuration mode.
Step 14	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.

	Command or Action	Purpose
Step 15	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface type and number and enters interface configuration mode.
Step 16	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding SERVICES`	Associates a VRF instance with an interface.
Step 17	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.3 255.255.255.0`	Sets a primary IP address for an interface.
Step 18	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Restarts an interface.
Step 19	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 20	**router ospf** *process-id* **vrf** *vrf-name* **Example:** `Router(config)# router ospf 99 vrf SERVICES`	Configures an OSPF routing process and enters router configuration mode. • This example uses OSPF; EIGRP is also available.
Step 21	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(config-router)# network 192.168.1.0 0.0.0.255 area 0`	Defines the interfaces on which OSPF runs and the area ID for those interfaces.
Step 22	**exit** **Example:** `Router(config-router)# exit`	Exits to global configuration mode.
Step 23	**router ospf** *process-id* [**vrf** *vrf-name*] **Example:** `Router(config)# router ospf 98 vrf RED`	Configures an OSPF routing process and enters router configuration mode.

	Command or Action	Purpose
Step 24	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(config-router)# network 192.168.1.0 0.0.0.255 area 0`	Defines the interfaces on which OSPF runs and the area ID for those interfaces.
Step 25	**exit** **Example:** `Router(config-router)# exit`	Exits to the global configuration mode.
Step 26	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition RED`	Defines a VRF and enters VRF configuration mode.
Step 27	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 28	**route-replicate from** [**vrf** *vrf-name*] {**multicast**| **unicast**} {**all**| *protocol-name*} [**route-map** *map-tag*] **Example:** `Router(config-vrf-af)# route replicate from vrf SERVICES unicast all`	Replicates routes into the base topology within the specified address family. • If the **all** keyword is specified as a source protocol, only one **route-replicate** command is allowed per VRF for a given destination topology. • Use the **connected** keyword as a source *protocol-name* in order to replicate only connected routes.
Step 29	**exit** **Example:** `Router(config-vrf-af)# exit`	Exits to VRF configuration mode.
Step 30	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.
Step 31	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition SERVICES`	Defines a VRF and enters VRF configuration mode.

	Command or Action	Purpose
Step 32	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 33	**route-replicate from** [**vrf** *vrf-name*] {**multicast**| **unicast**} {**all**| *protocol-name*} [**route-map** *map-tag*] **Example:** `Router(config-vrf-af)# route replicate from vrf RED unicast all`	Replicates routes into the base topology within the specified address family. • This is the reciprocal replication to Step 28 to allow bidirectional traffic.
Step 34	**end** **Example:** `Router(config-vrf-af)# end`	Exits configuration mode.
Step 35	**show ip route vrf** *vrf-name* **Example:** `Router# show ip route vrf RED`	(Optional) Displays routes, including those replicated, which are indicated by a plus sign (+).

Example

The following is sample output from the **show ip route vrf** command based on the task in the preceding task table:

```
Router# show ip route vrf RED

Routing Table: RED
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
Gateway of last resort is not set
   192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C   +    192.168.1.0/24 is directly connected (SERVICES), GigabitEthernet0/0/0
L   +    192.168.1.3/32 is directly connected (SERVICES), GigabitEthernet0/0/0
Router#
```

What to Do Next

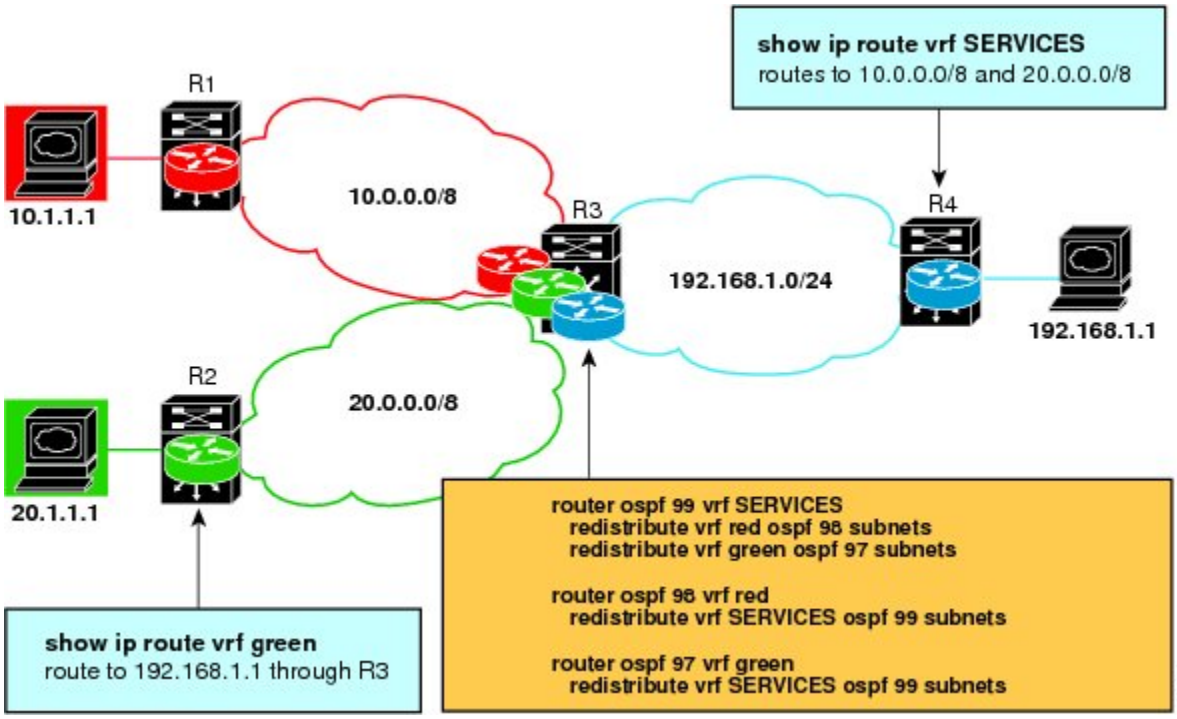
After you perform the “Configuring Route Replication to Share Services in Easy Virtual Network” task, you must configure VRF GREEN as per the figure above, noting that Router 3 has routes to 10.0.0.0/8 and 20.0.0.0/8 and Router 1 and Router 2 have a route to 192.168.1.0/24.

After the configuration is complete, Router 1 and Router 2 still do not have a route to the shared service residing on 192.168.1.1 and Router 4 does not have routes to 10.0.0.0/8 and 20.0.0.0/8. Such access requires the route redistribution performed in the next task, "Configuring Redistribution to Share Services in EVN".

Configuring Redistribution to Share Services in Easy Virtual Network

This task is based on the assumption that you also performed the task, Configuring Route Replication to Share Services in EVN.

The figure below shows the same networks we used in the figure above. In this task, we perform redistribution on Router 3 so that Router 1 and Router 2 have a route to the shared service residing on 192.168.1.1.



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* **vrf** *vrf-name*
4. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
5. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
6. **exit**
7. **router ospf** *process-id* **vrf** *vrf-name*
8. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
9. **exit**
10. **router ospf** *process-id* **vrf** *vrf-name*
11. **redistribute vrf** *vrf-name* **ospf** *process-id* **subnets**
12. **end**
13. **show ip route vrf** *vrf-name*

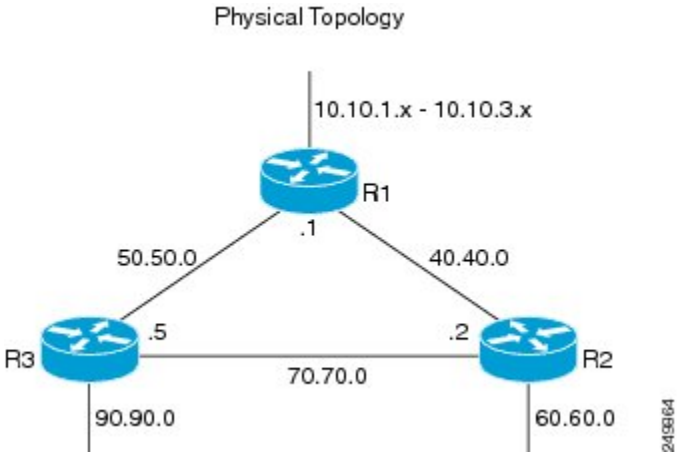
DETAILED STEPS

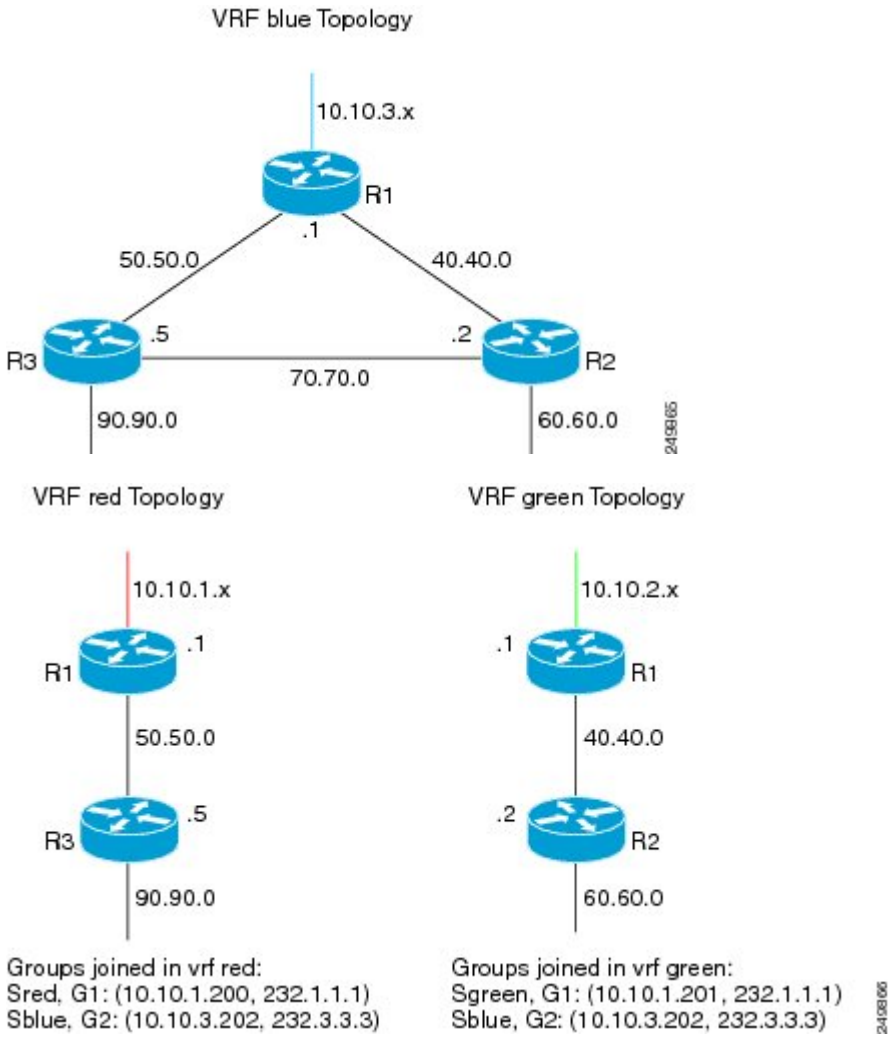
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **vrf** *vrf-name* **Example:** `Router(config)# router ospf 99 vrf SERVICES`	Configures an OSPF routing process and enters router configuration mode.
Step 4	**redistribute vrf** *vrf-name* **ospf** *process-id* **subnets** **Example:** `Router(config-router)# redistribute vrf RED ospf 98 subnets`	Redistributes routes from one routing domain into another routing domain.
Step 5	**redistribute vrf** *vrf-name* **ospf** *process-id* **subnets** **Example:** `Router(config-router)# redistribute vrf GREEN ospf 97 subnets`	Redistributes routes from one routing domain into another routing domain.
Step 6	**exit** **Example:** `Router(config-router)# exit`	Exits to global configuration mode.
Step 7	**router ospf** *process-id* **vrf** *vrf-name* **Example:** `Router(config)# router ospf 98 vrf RED`	Configures an OSPF routing process and enters router configuration mode.
Step 8	**redistribute vrf** *vrf-name* **ospf** *process-id* **subnets** **Example:** `Router(config-router)# redistribute vrf SERVICES ospf 99 subnets`	Redistributes routes from one routing domain into another routing domain.
Step 9	**exit** **Example:**	Exits to global configuration mode.

	Command or Action	Purpose
	`Router(config-router)# exit`	
Step 10	**router ospf** *process-id* **vrf** *vrf-name* **Example:** `Router(config)# router ospf 97 vrf GREEN`	Configures an OSPF routing process and enters router configuration mode.
Step 11	**redistribute vrf** *vrf-name* **ospf** *process-id* **subnets** **Example:** `Router(config-router)# redistribute vrf SERVICES ospf 99 subnets`	Redistributes routes from one routing domain into another routing domain.
Step 12	**end** **Example:** `Router(config-router)# end`	Exits configuration mode.
Step 13	**show ip route vrf** *vrf-name* **Example:** `Router# show ip route vrf RED`	(Optional) Displays routes, including those replicated, which are indicated by a plus sign (+).

Configuration Example for Easy Virtual Network Shared Services

Example: Easy Virtual Network Route Replication and Route Redistribution in a Multicast Environment





In the figures above there are three multicast streams:

- Sred, G1: (10.10.1.200, 232.1.1.1)--Source and receivers in VRF red
- Sgreen, G1: (10.10.2.201, 232.1.1.1)--Source and receivers in VRF green
- Sblue, G2: (10.10.3.202, 232.3.3.3)--Source in blue and receivers in VRFs red and green.

The server-prefix in VRF blue (10.10.3.0/24) is replicated and distributed into VRFs red and green on R3 and R2.

Multicast group 232.3.3.3 with its source in VRF blue has receivers in both VRF red and VRF green. The stream is transmitted over the shared VRF (blue), and then replicated into VRF red on R3 and into VRF green on R2.

R1 Configuration

```
vrf definition blue
 vnet tag 4
 !
```

```
 address-family ipv4
 exit-address-family
!
vrf definition green
 vnet tag 3
 !
 address-family ipv4
 exit-address-family
!
vrf definition red
 vnet tag 2
 !
 address-family ipv4
 exit-address-family
!
vrf list vnet-list1
 member blue
 member red
!
vrf list vnet-list2
 member blue
 member green
!
vrf list vnet-list3
 member blue
!
ip multicast-routing distributed
ip multicast-routing vrf red distributed
ip multicast-routing vrf green distributed
ip multicast-routing vrf blue distributed
!
interface FastEthernet0/0/2
 vnet trunk list vnet-list1                                [vnet trunk for red and blue]
 ip address 50.50.0.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
 !
interface GigabitEthernet0/1/1
 vnet trunk list vnet-list2                                   [vnet trunk for green and blue]

 ip address 40.40.0.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
interface GigabitEthernet0/1/3
 ip address 10.10.0.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
interface GigabitEthernet0/1/3.2
 vrf forwarding red
 encapsulation dot1Q 2
 ip address 10.10.1.1 255.255.255.0
 ip pim sparse-dense-mode
!
interface GigabitEthernet0/1/3.3
 vrf forwarding green
 encapsulation dot1Q 3
 ip address 10.10.2.1 255.255.255.0
 ip pim sparse-dense-mode
!
```

```
interface GigabitEthernet0/1/3.4
 vrf forwarding blue
 encapsulation dot1Q 4
 ip address 10.10.3.1 255.255.255.0
 ip pim sparse-dense-mode
!
router ospf 201 vrf red
 nsf
 redistribute connected subnets
 network 10.10.1.0 0.0.0.255 area 0
 network 50.50.0.0 0.0.0.255 area 0
!
router ospf 202 vrf green
 nsf
 network 10.10.2.0 0.0.0.255 area 0
 network 40.40.0.0 0.0.0.255 area 0
!
router ospf 203 vrf blue
 router-id 11.11.11.11
 nsf
 network 10.10.3.0 0.0.0.255 area 0
 network 40.40.0.0 0.0.0.255 area 0
 network 50.50.0.0 0.0.0.255 area 0
!
router ospf 200
 nsf
 redistribute connected subnets
 network 10.10.0.0 0.0.0.255 area 0
 network 40.40.0.0 0.0.0.255 area 0
 network 50.50.0.0 0.0.0.255 area 0
!
ip pim ssm default
ip pim vrf red ssm default
ip pim vrf green ssm default
ip pim vrf blue ssm default
!
```

R2 Configuration

```
vrf definition blue
 vnet tag 4
 !
 address-family ipv4
 exit-address-family
!
vrf definition green
 vnet tag 3
 !
 address-family ipv4
  route-replicate from vrf blue unicast all route-map blue-map
  [replicate routes from blue to green]
 exit-address-family
!
vrf definition red
 vnet tag 2
 !
 address-family ipv4
 exit-address-family
!
vrf list vnet-list1
 member blue
 member green
!
```

```
vrf list vnet-list2
member blue
!
ip multicast-routing distributed
ip multicast-routing vrf red distributed
ip multicast-routing vrf  green distributed
ip multicast-routing vrf blue distributed
!
interface FastEthernet0/0/6
 vnet trunk list vnet-list2                    [vnet trunk for blue]
 ip address 70.70.0.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
interface GigabitEthernet0/1/2
 vnet trunk list vnet-list1    [vnet trunk for green and blue]
 ip address 40.40.0.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
interface GigabitEthernet0/1/4
 vnet trunk list vnet-list1                                [vnet trunk for green and blue]

 ip address 60.60.0.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
router ospf 202 vrf green
 redistribute connected subnets
 redistribute vrf blue ospf 203 subnets route-map blue-map  [redistribute routes replicated
 from blue in red]
 network 40.40.0.0 0.0.0.255 area 0
 network 60.60.0.0 0.0.0.255 area 0
!
router ospf 203 vrf blue
 router-id 22.22.22.22
 network 40.40.0.0 0.0.0.255 area 0
 network 60.60.0.0 0.0.0.255 area 0
 network 70.70.0.0 0.0.0.255 area 0
!
router ospf 200
 redistribute connected subnets
 network 40.40.0.0 0.0.0.255 area 0
 network 60.60.0.0 0.0.0.255 area 0
 network 70.70.0.0 0.0.0.255 area 0
!
ip pim ssm default
ip pim vrf red ssm default
ip pim vrf green ssm default
ip pim vrf blue ssm default
!
ip prefix-list server-prefix seq 5 permit 10.10.3.0/24
!
route-map blue-map permit 10
 match ip address prefix-list server-prefix
!
```

R3 Configuration

```
vrf definition blue
```

```
 vnet tag 4
 !
 address-family ipv4
 exit-address-family
!
vrf definition green
 vnet tag 3
 !
 address-family ipv4
 exit-address-family
!
vrf definition red
 vnet tag 2
 !
 address-family ipv4
  route-replicate from vrf blue unicast all route-map blue-map   [replicate routes from
blue  to red]
 exit-address-family
!
vrf list vnet-list1
 member blue
 member red
!
vrf list vnet-list2
 member blue
!
ip multicast-routing distributed
ip multicast-routing vrf red distributed
ip multicast-routing vrf green distributed
ip multicast-routing vrf blue distributed
!
interface GigabitEthernet0/2/0
 vnet trunk list vnet-list1                          [vnet trunk for red and blue]
 ip address 90.90.0.5 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
interface GigabitEthernet1/2/0
 vnet trunk list vnet-list1                        [vnet trunk for red and blue]
 ip address 50.50.0.5 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
interface FastEthernet2/0/0
 vnet trunk list vnet-list2                          [vnet trunk for blue]
 ip address 70.70.0.5 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
!
router ospf 201 vrf red
 redistribute connected subnets
 redistribute vrf blue ospf 203 subnets route-map blue-map      [redistribute routes
replicated from blue in red]
 network 50.50.0.0 0.0.0.255 area 0
 network 90.90.0.0 0.0.0.255 area 0
!
router ospf 203 vrf blue
 router-id 55.55.55.55
 network 50.50.0.0 0.0.0.255 area 0
 network 70.70.0.0 0.0.0.255 area 0
 network 90.90.0.0 0.0.0.255 area 0
```

```
!
router ospf 200
 redistribute connected subnets
 network 50.50.0.0 0.0.0.255 area 0
 network 70.70.0.0 0.0.0.255 area 0
 network 90.90.0.0 0.0.0.255 area 0
!
ip pim ssm default
ip pim vrf red ssm default
ip pim vrf green ssm default
ip pim vrf blue ssm default
!
ip prefix-list server-prefix seq 5 permit 10.10.3.0/24
!
route-map blue-map permit 10
 match ip address prefix-list server-prefix
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Easy Virtual Network commands	Easy Virtual Network Command Reference
Overview of Easy Virtual Network	"Overview of Easy Virtual Network" module in the *Easy Virtual Network Configuration Guide*
Configuring Easy Virtual Network	"Configuring Easy Virtual Network" module in the *Easy Virtual Network Configuration Guide*
Easy Virtual Network management and troubleshooting	"Easy Virtual Network Management and Troubleshooting" module in the *Easy Virtual Network Configuration Guide*

MIBs

MIB	MIBs Link
Any MIB that gives VRF information will continue to work with Easy Virtual Network. VRF-independent MIBs report information on every VRF in a system: • CISCO-MVPN-MIB • MPLS-VPN-MIB • CISCO-VRF-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Easy Virtual Network Shared Services

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 158: Feature Information for Easy Virtual Network Shared Services

Feature Name	Releases	Feature Information
EVN Route Replication	Cisco IOS XE Release 3.2S 15.0(1)SY 15.1(1)SG Cisco IOS XE Release 3.3SG 15.3(2)T	This module describes how to use route replication and redistribution to share services in an EVN environment. This feature modifies the following command: **redistribute** (IP) This feature introduces the following command: **route-replicate** (VRF address family)

PART XI

Addressing Fragmentation and Reassembly

CHAPTER 115

Virtual Fragmentation Reassembly

Virtual fragmentation reassembly (VFR) is automatically enabled by some features (such as NAT, Cisco IOS XE Firewall, IPSec) to get Layer 4 or Layer 7 information. VFR enables the Cisco IOS XE Firewall to create appropriate dynamic access control lists (ACLs) to protect the network from various fragmentation attacks.

Most non-initial fragments do not have the Layer 4 header because it usually travels with the initial fragments (except in the case of micro-fragmentation and tiny fragments). Due to this, some features (such as NAT, Cisco IOS XE Firewall, IPSec) are unable to gather port information from the packet. These features may need to inspect the Layer 7 payload, for which the fragments need to be reassembled, and then refragmented later.

Note From Cisco IOS XE Release 17.7.1, when you are running a Cisco IOS-XE router as an SSL VPN gateway, an extra SSL VPN overhead is added due to the TLS encapsulation. To prevent IP fragmentation and reassembly of packets between SSL VPN client and server, you must adjust the TCP-MSS value optimally. Otherwise, packet drop due to the IPFragErr error could occur in the SSL VPN gateway. This guideline is applicable for the Cisco 4400 Series ISR platform.

Restrictions for Virtual Fragmentation Reassembly

Performance Impact

VFR causes a performance impact on the basis of functions such as packet copying, fragment validation, and fragment reorder. This performance impact varies depending on the number of concurrent IP datagrams that are being reassembled.

VFR Configuration

The reassembly process requires all fragments within an IP datagram. If fragments within an IP datagram are sent to different devices due to load balancing (per packet load balancing or include ports on Cisco Catalyst 6500 Series Switches or Cisco Nexus devices), VFR may fail and fragments may be dropped.

Information About Virtual Fragmentation Reassembly

VFR Detection of Fragment Attacks

VFR is responsible for detecting and preventing the following types of fragment attacks:

- Tiny fragment attack—In this type of attack, the attacker makes the fragment size small enough to force Layer 4 (TCP and UDP) header fields into the second fragment. Thus, the ACL rules that have been configured for those fields do not match.
- VFR drops all tiny fragments, and an alert message such as "VFR-3-TINY_FRAGMENTS" is logged to the syslog server.
- Overlapping fragment attack—In this type of attack, the attacker can overwrite the fragment offset in the noninitial IP fragment packets. When the firewall reassembles the IP fragments, it might create wrong IP packets, causing the memory to overflow or the system to reload.
- VFR drops all fragments within a fragment chain if an overlap fragment is detected.
- Buffer overflow attack—In this type of denial-of-service (DoS) attack, the attacker can continuously send a large number of incomplete IP fragments, causing the firewall to consume time and memory while trying to reassemble the fake packets.

To avoid buffer overflow and control memory use, configure a maximum threshold for the number of IP datagrams that are being reassembled and the number of fragments per datagram. You can use the **ip virtual-reassembly** command or the **ip virtual-reassembly-out** command to specify these parameters.

When the maximum number of datagrams that can be reassembled at any given time is reached, all subsequent fragments are dropped, and the global statistics item "ReassDrop" is incremented by one.

When the maximum number of fragments per datagram is reached, subsequent fragments are dropped, and the global statistics item "ReassTooManyFrags" is incremented by one.

In addition to the maximum threshold values being configured, each IP datagram is associated with a managed timer. If the IP datagram does not receive all of the fragments within the specified time, the timer expires and the IP datagram and all of its fragments are dropped.

VFR Enablement

VFR is designed to work with any feature that requires fragment reassembly (such as Cisco IOS XE Firewall, NAT, and IPSec). By default, NAT, Cisco IOS XE Firewall, Crypto-based IPSec, NAT64, and onePK enable and disable VFR internally; that is, when these features are enabled on an interface, VFR is automatically enabled on that interface.

If more than one feature attempts to automatically enable VFR on an interface, VFR maintains a reference count to keep track of the number of features that have enabled VFR. When the reference count is reduced to zero, VFR is automatically disabled.

If NAT is enabled on an interface (such as GigabitEthernet 0/0/0), VFR (input/output) is enabled on this interface.

```
Device(config-if)# do show ip virtual-reassembly features
GigabitEthernet0/0/0:
  Virtual Fragment Reassembly (VFR) Current Status is ENABLED [in]
  Features to use if VFR is Enabled:NAT
GigabitEthernet0/0/0:
  Virtual Fragment Reassembly (VFR) Current Status is ENABLED [out]
  Features to use if VFR is Enabled:NAT
```

If Cisco IOS XE Firewall is enabled on an interface (such as GigabitEthernet 0/0/0), VFR (out) is enabled on this interface.

```
Device(config-if)# do show ip virtual-reassembly features
GigabitEthernet0/0/0:
  Virtual Fragment Reassembly (VFR) Current Status is ENABLED [out]
  Features to use if VFR is Enabled:FW
```

If IPSec is enabled on an interface (such as GigabitEthernet 0/0/0), VFR (out) is enabled on this interface.

```
Device(config-if)# do show ip virtual-reassembly features
GigabitEthernet0/0/0:
  Virtual Fragment Reassembly (VFR) Current Status is ENABLED [out]
  Features to use if VFR is Enabled:IPSec
```

Note If VFR is enabled by features such as NAT and Cisco IOS XE Firewall, the **ip virtual-reassembly [-out]** command is not displayed in the output of the **show running-config** command.

VFR can be manually enabled or disabled using the [**no**] **ip virtual-reassembly [-out]** command.

If VFR is manually enabled, regardless of whether it is enabled by features such as NAT and Cisco IOS XE Firewall, the **ip virtual-reassembly [-out]** command is displayed in the output of the **show running-config** command.

VFR Disablement

You can disable virtual fragmentation reassembly (VFR) using the following methods:

- If VFR is manually enabled, it can be manually disabled using the **no ip virtual-reassembly [-out]** command. This command is not displayed in the output of the **show running-config** command.
- If VFR is enabled by a feature (such as NAT or Cisco IOS Firewall), it can be manually disabled or it can be disabled by disabling the feature. If it is manually disabled, the **no ip virtual-reassembly [-out]** command is displayed in the output of the **show running-config** command.
- If VFR is both manually enabled and enabled by features, it can be manually disabled using the **no ip virtual-reassembly [-out]** command. This command is displayed in the output of the **show running-config** command.

Note If VFR is not enabled, the **no ip virtual-reassembly [-out]** command is not displayed in the output of the **show running-config** command.

To enable VFR after it is disabled, that is, when the **no ip virtual-reassembly [-out]** command is displayed in the output of the **show running-config** command, manually enable VFR using the **ip virtual-reassembly [-out]** command or disable related features and then enable the features again.

In a crypto map-based IPSec deployment scenario (such as GETVPN), VFR is enabled by default in devices which are configured with IPSec. Fragments of the same packet may be sent to different devices (which are IPSec-enabled) by upper devices due to the packet load balance algorithm (per packet load balance or per destination on some Nexus devices). VFR may drop the fragments if it does not receive all fragment of the same IP packet. The recommended workaround of this issue is to change the load balance algorithm to ensure all fragments of the same packet go to the same path. If Layer 4 information (ports) is not a filter criterion in IPSec policy, another workaround is to manually disable VFR using **no ip virtual-reassembly [-out]** on interfaces where IPSec is configured.

VFR on Outbound Interfaces

In Cisco IOS XE Release 3.2S and later releases, you can use the **ip virtual-reassembly-out** command to manually enable or disable VFR on outbound interface traffic.

How to Configure Virtual Fragmentation Reassembly

Configuring VFR

Perform this task to enable VFR on an interface to specify maximum threshold values to combat buffer overflow and control memory usage, and to verify any VFR configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **ip virtual-reassembly** [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**drop-fragments**]
5. **end**
6. **show ip virtual-reassembly** [*interface type*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface GigabitEthernet0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip virtual-reassembly** [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**drop-fragments**] **Example:** `Device(config-if)# ip virtual-reassembly max-reassemblies 64 max-fragments 16 timeout 5`	Enables VFR on the interface and specifies the maximum threshold values.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show ip virtual-reassembly** [*interface type*] **Example:** `Device# show ip virtual-reassembly GigabitEthernet0/0/1`	Displays the configuration and statistical information of the VFR. • If an interface is not specified, VFR information is shown for all configured interfaces.

Enabling VFR Manually on Outbound Interface Traffic

Perform this task to enable VFR manually on outbound interface traffic. You can use this procedure to reenable VFR on outbound interface traffic if it is disabled, for example, by the **no ip virtual-reassembly** command.

Note If VFR is enabled on both inbound and outbound interface traffic, you can use the **no ip virtual-reassembly [-out]** command to disable it on only the outbound interface traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip virtual-reassembly** [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**drop-fragments**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip virtual-reassembly** [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**drop-fragments**] **Example:** `Device(config-if)# ip virtual-reassembly max-reassemblies 64 max-fragments 16 timeout 5`	Enables VFR on the interface and specifies the maximum threshold values.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode.

Troubleshooting Tips

To display debugging messages related to the VFR subsystem, use the **debug ip virtual-reassembly** command.

Configuration Examples for Virtual Fragmentation Reassembly

Example: Configuring VFR on Outbound Interface Traffic

The following example shows how to manually enable VFR on outbound traffic on interfaces GigabitEthernet0/0/1, GigabitEthernet0/0/0.773, and Serial 3/0:

```
interface Loopback 0
 ip address 10.0.1.1 255.255.255.255
 !
interface GigabitEthernet0/0/1
 description LAN1
 ip address 10.4.0.2 255.255.255.0
```

```
 ip virtual-reassembly-out
 !
interface GigabitEthernet0/0/0.773
 encapsulation dot1Q 773
 description LAN2
 ip address 10.15.0.2 255.255.255.0
 ip virtual-reassembly-out
 !
interface Serial 3/0
 description Internet
 ip unnumbered Loopback0
 encapsulation ppp
 ip virtual-reassembly-out
 serial restart-delay 0
```

Additional References for Virtual Fragmentation Reassembly

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Dynamic IDS	Cisco IOS Intrusion Prevention System
CBAC	"Configuring Context-Based Access Control" chapter

RFCs

RFCs	Title
RFC 791	*Internet Protocol*
RFC 1858	*Security Considerations for IP Fragment Filtering*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Virtual Fragmentation Reassembly

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 159: Feature Information for Virtual Fragmentation Reassembly

Feature Name	Releases	Feature Information
Virtual Fragmenatation Reassembly	Cisco IOS XE Release 3.2S	VFR enables the Cisco IOS Firewall to create the appropriate dynamic ACLs to protect the network from various fragmentation attacks. In Cisco IOS Release XE 3.2S, functionality to manually configure VFR for outbound or inbound interface traffic was added. The following commands were introduced or modified: **ip virtual-reassembly-out, show ip virtual-reassembly**.

CHAPTER 116

IPv6 Virtual Fragmentation Reassembly

Information About IPv6 Virtual Fragmentation Reassembly

IPv6 Virtual Fragmentation Reassembly

Fragmentation is a process of breaking down an IP datagram into smaller packets to be transmitted over different types of network media. Non-initial fragments of a fragmented IPv6 packet is used to pass through IPsec and NAT64 without any examination due to the lack of the L4 header, which usually is only available on the initial fragment. The IPv6 Virtual Fragmentation Reassembly (VFR) feature provides the ability to collect the fragments and provide L4 info for all fragments for IPsec and NAT64 features.

How to Implement IPv6 Virtual Fragmentation Reassembly

Configuring IPv6 Virtual Fragmentation Reassembly

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. ipv6 virtual-reassembly [**in** | **out**] [**max-reassemblies** *maxreassemblies*] [**max-fragments** *max-fragments*] [**timeout** *seconds*] [**drop-fragments**
5. **exit**
6. show ipv6 virtual-reassembly interface *interface-type*
7. show ipv6 virtual-reassembly features interface *interface-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 3/1/1`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 virtual-reassembly [**in** | **out**] [**max-reassemblies** *maxreassemblies*] [**max-fragments** *max-fragments*] [**timeout** *seconds*] [**drop-fragments** **Example:** `Router(config-if)# ipv6 virtual-reassembly max-reassemblies 32 max-fragments 4 timeout 7`	Enables VFR on an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and places the router in global configuration mode. • Enter this command twice to reach privileged EXEC mode.
Step 6	show ipv6 virtual-reassembly interface *interface-type* **Example:** `Router# show ipv6 virtual-reassembly interface e1/1/1`	Displays VRF configuration and statistical information on a specific interface.
Step 7	show ipv6 virtual-reassembly features interface *interface-type* **Example:** `Router# show ipv6 virtual-reassembly features`	Displays VFR information on all interfaces or on a specified interface.

Configuration Example for IPv6 Virtual Fragmentation Reassembly

Example: Configuring IPv6 Virtual Fragmentation Reassembly

```
Router# show ipv6 virtual-reassembly interface gigabitethernet1/1/1
GigabitEthernet1/1/1:
IPv6 Virtual Fragment Reassembly (VFR) is ENABLED(in)
Concurrent reassemblies (max-reassemblies): 64
Fragments per reassembly (max-fragments): 16
Reassembly timeout (timeout): 3 seconds
Drop fragments: OFF
Current reassembly count: 0
Current fragment count: 0
Total reassembly count: 6950
Total reassembly timeout count: 9
GigabitEthernet1/1/1:
IPv6 Virtual Fragment Reassembly (VFR) is ENABLED(out)
Concurrent reassemblies (max-reassemblies): 64
Fragments per reassembly (max-fragments): 16
Reassembly timeout (timeout): 3 seconds
Drop fragments: OFF
Current reassembly count: 0
Current fragment count: 0
Total reassembly count: 0
Total reassembly timeout count: 0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Master Commands List, All Releases*
IPv6 commands	*IPv6 Command Reference*
Cisco IOS IPv6 features	*IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Virtual Fragmentation Reassembly

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 160: Feature Information for IPv6 Virtual Fragmentation Reassembly

Feature Name	Releases	Feature Information
IPv6 Virtual Fragmentation Reassembly	Cisco IOS XE Release 3.4S	The IPv6 VFR feature provides the ability to collect the fragments and provide L4 info for all fragments for IPsec and NAT64 features.

CHAPTER 117

GRE Fragment and Reassembly Performance Tuning

The GRE Fragment and Reassembly Performance Tuning feature enables you to customize reassembly resources. Reassembly resources are equally allocated to each interface to prevent fragment-related attack. However, in some generic routing encapsulation (GRE) tunnel deployments, fragments are reassembled in specific interfaces. This feature also allows you to adjust the reassembly timer to free up incomplete fragment sessions quickly and reserve the reassembly resources for high priority packets.

Restrictions for GRE Fragment and Reassembly

- The IPv4 or IPv6 protocol must be enabled on an interface.
- This feature supports manually created tunnel interfaces or physical interfaces (virtual template is not officially supported).

Information About GRE Fragment and Reassembly

Fragmentation and Reassembly

In Cisco software, packets may be dropped due to nonavailability of reassembly resources of an interface when fragments arrive concurrently on an interface, though, other interfaces have the resources to reassemble fragments. In some cases, some interfaces need additional resources, such as generic routing encapsulation (GRE) tunnel deployment, and resources are freed only when fragments are reassembled. Therefore, if all fragments are not received, the reassembly resources are not freed.

The GRE Fragment and Reassembly Performance Tuning feature improves reassembly performance by reassembling high priority fragments first so that these fragments are not dropped when low priority fragments occupy the reassembly resources.

Out of Order Packet Processing

Sometimes, a big packet may be received before a small packet, but forwarded after a small packet. Consider a scenario, in which a big packet followed by a small packet (packet size smaller than the egress interface MTU). The big packet may be fragmented and reassembled. Fragmentation and reassembly of the big packet requires an additional processor cycle. Devices that run on Cisco IOS XE software follow multithread processing. That is, small packet require shorter processing time and, hence, may be forwarded before the fragmented big packet. This process results in packet sequence changes on the receiver's end (big packets received before small packets, but may be forwarded out after small packets).

How to Use GRE Fragment and Reassembly

Configuring GRE Fragment and Reassembly (GFR)

Perform this task to do the following:

- Enable generic routing encapsulation (GRE) Fragment and Reassembly (GFR) on an interface
- Specify maximum threshold values to combat buffer overflow and control memory usage
- Verify GFR configurations

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Enter one of the following:
   - **ip reassembly** [**max-reassemblies** *number*] [**timeout** *milliseconds*] [**percentage** *percent* {**dscp** *dscp-value* | **precedence** *precedence-value*}]
   - **ipv6 reassembly** [**max-reassemblies** *number*] [**timeout** *milliseconds*] [**percentage** *percent* {**dscp** *dscp-value* | **precedence** *precedence-value*}]
5. **end**
6. Enter one of the following:
   - **show ip reassembly interface** *type number*
   - **show ipv6 reassembly interface** *type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.

Step 4	Enter one of the following:
	• **ip reassembly** [**max-reassemblies** *number*] [**timeout** *milliseconds*] [**percentage** *percent* {**dscp** *dscp-value* | **precedence** *precedence-value*}]
	• **ipv6 reassembly** [**max-reassemblies** *number*] [**timeout** *milliseconds*] [**percentage** *percent* {**dscp** *dscp-value* | **precedence** *precedence-value*}]
	Example:
	`Device(config-if)# ip reassembly max-reassemblies 1024 timeout 1000 percentage 50 precedence critical routine`
	Example:
	`Device(config-if)# ipv6 reassembly max-reassemblies 1024 timeout 1000 percentage 50 precedence critical routine`
		Enables GFR on an IPv4 or IPv6 interface, as appropriate.

Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Step 6	Enter one of the following:
	• **show ip reassembly interface** *type number*
	• **show ipv6 reassembly interface** *type number*
	Example:
	`Device# show ip reassembly GigabitEthernet 0/0/0`
	Example:
	`Device# show ipv6 reassembly GigabitEthernet 0/0/0`
		Displays statistical information of the GFR configured about the interface.

Configuration Examples for GRE Fragment and Reassembly

Example: Configuring GFR

The following example shows how to configure GFR on a Gigabit Ethernet interface and specify the maximum reassembly and timeout settings:

```
interface GigabitEthernet 0/0/0
 ip address 10.10.10.1 255.255.255.0
 ipv6 address 2001:DB8:1::1
 ip reassembly max-reassemblies 1024 timeout 1 percentage 50 dscp ef
 ipv6 reassembly max-reassemblies 1024 timeout 1 percentage 50 dscp ef
 ip virtual-reassembly max-reassemblies 1024 timeout 1 percentage 10 dscp af41
 ipv6 reassembly out max-reassemblies 1024 timeout 1 percentage 50 precedence cs1
```

Additional References for GRE Fragment and Reassembly

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Virtual Fragmentation and Reassembly	*Virtual Fragmentation and Reassembly*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GRE Fragment and Reassembly

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 161: Feature Information for GRE Fragment and Reassembly

Feature Name	Releases	Feature Information
GRE Fragment and Reassembly Performance Tuning	Cisco IOS XE Release 3.8S	The GRE Fragment and Reassembly Performance Tuning feature enables you to customize reassembly resources. Reassembly resources are equally allocated to each interface to prevent fragment-related attack. However, in some generic routing encapsulation (GRE) tunnel deployments, fragments are reassembled in specific interfaces. This feature also allows you to adjust the reassembly timer to free up incomplete fragment sessions quickly and reserve the reassembly resources for high priority packets. The following commands were introduced or modified: **ip reassembly**, **show ip reassembly**.