Understand and Configure NAT64

Contents

Introduction

This document describes how to understand and configure Network Address Translation (NAT).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- IPv6
- NAT

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure

that you understand the potential impact of any command.

Background Information

NAT64 is a mechanism for IPv4-to-IPv6 transition and IPv4-IPv6 coexistence. Together with DNS64, the primary purpose of NAT64 is to allow an IPv6-only client to initiate communications to an IPv4-only server. NAT64 can also be used for IPv4-only clients initiating communications with IPv6-only servers using static or manual bindings. Both scenarios are explained in this document.

Why There is a Need for NAT64

- Almost all modern IP devices are IPv6-capable, but still many older devices are IPv4-only. We need a way to connect these devices across an IPv6 network.
- Some older applications that incorporate IPv4 addresses into the upper layers can be expected to still be around for a while and must be adapted to IPv6.
- As IPv4 addresses become unavailable, IPv6 addresses are assigned to new devices; however, the majority of reachable content on the Internet is still IPv4. These new devices must reach that content.
- After few years, the opposite can apply: The majority of content can be IPv6, but the few remaining IPv4-only devices must still reach it.
- IPv4-only devices must speak to IPv6-only devices with minimal or no user awareness.

How to Make Communication Between IPv4 and IPv6 Possible

Since IPv6 is not backward compatible with IPv4, you are left with the necessity of transition mechanisms, which fall into one of three classes:

- Dual-stacked interfaces: The simplest solution to IPv4 and IPv6 co-existence (not interoperability) is to make interfaces bilingual, so they can speak IPv4 to IPv4 devices and IPv6 to IPv6 devices. Which version they use depends either on the version of packets they receive from a device or the type of address DNS gives them when they query for a device address. Dual stack was the intended means of transitioning from IPv4 to IPv6, but the assumption was that the transition would be complete before IPv4 was depleted. That has not happened, so dual stacking becomes more complex: How do you give every interface both an IPv4 address and an IPv6 address when not enough IPv4 addresses are available to go around?

- Tunnels: Tunnels are also about co-existence, not interoperability. They allow devices or sites of one version to communicate across a network segment—including the Internet—of the other version. So two IPv4 devices or sites can exchange IPv4 packets across an IPv6 network, or two IPv6 devices or sites can exchange IPv6 packets across an IPv4 network.

- Translators: Translators create interoperability between an IPv4 device and an IPv6 device by changing the header of a packet of one version to the header of the other version.

#Like other transition methods, translation is not a long-term strategy and the ultimate goal can be native IPv6. However translation offers two major advantages over tunneling:

- Translation provides a means for gradual and seamless migration to IPv6.
- Content providers can provide services transparently to IPv6 Internet users.

Types of NAT64 Translation

Stateless NAT64

In stateless NAT64, state is not preserved which means for every IPv6 user a dedicated IPv4 address is required. As we are in IPv4 depletion phase, it is very difficult to adopt this mode of NAT64. The only advantage of using stateless NAT64 when you have few numbers of IPv6 addresses (NAT46).

Stateful NAT64

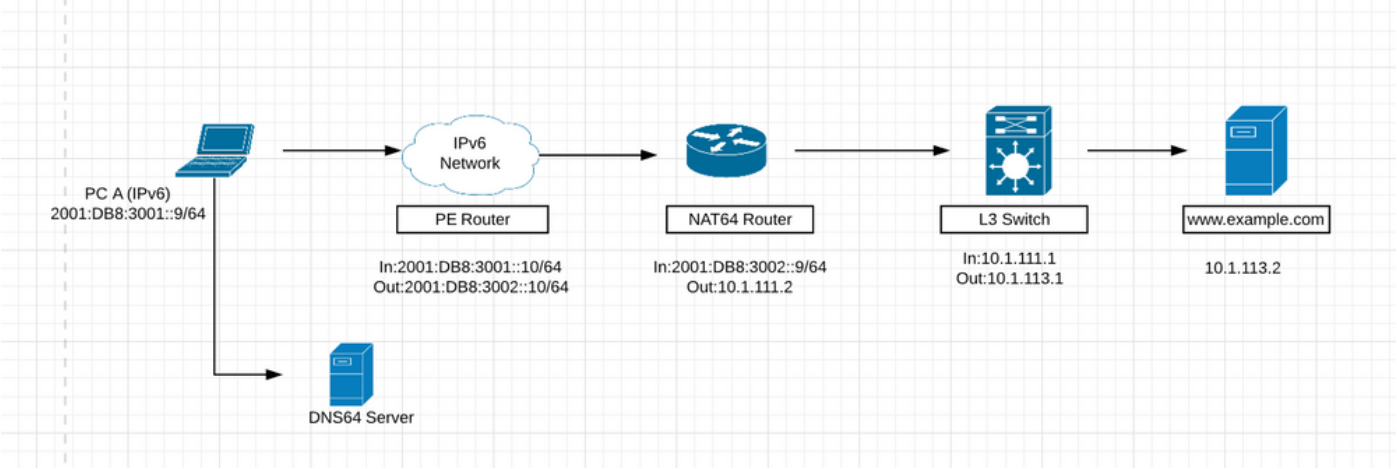
In stateful NAT64, states are maintained. A single IP Address is used for all the private users with different port numbers. In the previous diagram, a single IPv4 address is used with different port numbers for all the users of IPv6 which are in that LAN to access a public IPv4 server.

Here are more details about the difference between Stateful and Stateless NAT64 translation:

Stateless NAT64	**Stateful NAT64**
1:1 translation	1:N translation
No conservation of IPv4 address	Conserves IPv4 address
Assures end-to-end address transparency and scalability	Uses address overloading, hence lacks in end-to-end address transparency
No state or bindings created on the translation	State or bindings are created on every unique translation
Requires IPv4-translatable IPv6 addresses assignment (mandatory requirement)	No requirement on the nature of IPv6 address assignment
Requires either manual or DHCPv6 based address assignment for IPv6 hosts	Free to choose any mode of IPv6 address assignment viz. Manual, DHCPv6, SLAAC

- In this document, it is demonstrated stateful NAT64 with LAB exercise where IPv6 host wants to communicate to IPv4 server. Also, it is demonstrated stateless NAT64 where IPv4 hosts wants to reach out to IPv6 server. This scenario is also called NAT46.

Scenario 1: How to Communicate to IPv4 Server (Located in IPv4 Network) from Host in IPv6 Network

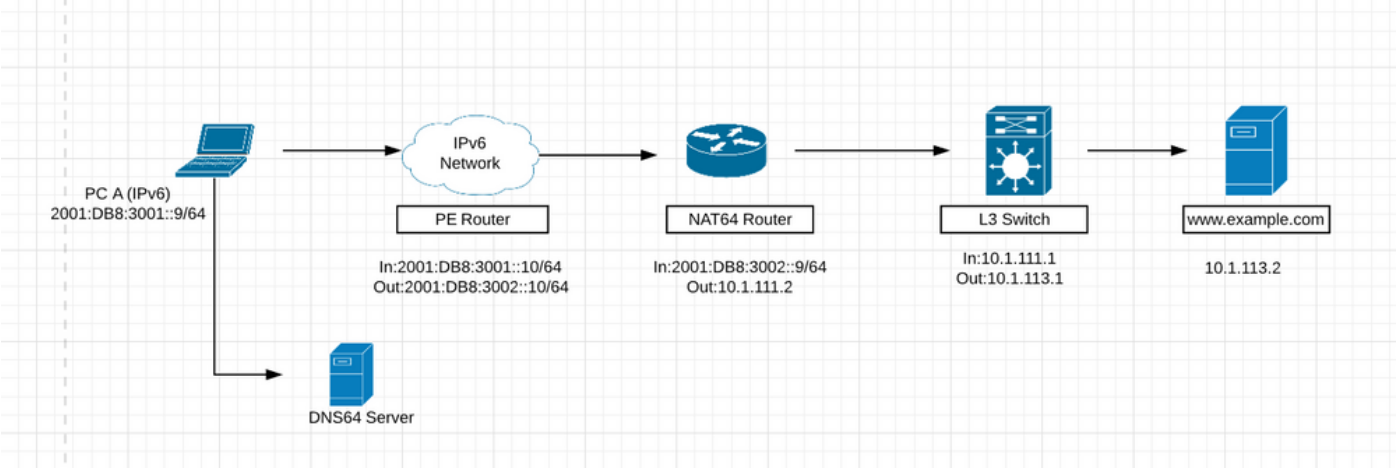


- In the previous picture, Host located in IPv6 network wants to reach to web server (www.example.com) with ip 10.1.113.2 located in ipv4 network.
- If you can directly ping the ipv4 address (10.1.113.2) from host in ipv6 network, the device does not understand this ipv4 address as it understands only ipv6 addresses. So the packet can get dropped on the host.
- Similarly, if you ping ipv6 address from ipv4 network, the device cannot understand that ip and it can throw an error as it is by default configured for ipv4 network only.
- Also, an ipv4 packet cannot get routed through an ipv6 only network and vice-versa. Hence, there is the need for translation so that you can translate the packets on edge devices to ipv4 or ipv6 depending upon requirement.

There are three main components to NAT64.

- NAT64 prefix: Any /32, /40, /48, /56, /64, or /96 prefix used with a converted IPv4 address to transmit the packet over the IPv6-only network. The NAT64 prefix can be a network-specific prefix (NSP) or a well-known prefix (WKP). An NSP is assigned by an organization and is usually a subnet from the organization's IPv6 prefix. The WKP for NAT64 is 64:ff9b::/96. If an NSP is not specified or configured, NAT64 can use the WKP to prepend the converted IPv4 address. The NAT64 prefix is also referred to as Pref64::/n.
- DNS64 server: The DNS64 server functions as a normal DNS server for IPv6 AAAA records but can also attempt to locate an IPv4 A record when a AAAA record is not available. If an A record is located, DNS64 converts the IPv4 A record into an IPv6 AAAA record using the NAT64 prefix. This gives the impression to the IPv6-only host that it can communicate with a server using IPv6.
- NAT64 router: The NAT64 router advertises the NAT64 prefix into the IPv6-only network along with performing the translation between the IPv6-only and IPv4-only networks.

Packet Flow in Case of Stateful NAT64



1. Suppose, in the previous picture, host present in IPv6 network wants to communicate to web server www.example.com (10.1.113.2) which is IPv4 only server.

2. To make this communication possible, you must have DNS64 server installed in IPv6 network which can understand and resolve DNS for ipv4 requests.

3. The DNS64 server functions as a normal DNS server for IPv6 AAAA records, but can also attempt to locate an IPv4 A record when a AAAA record is not available. If an A record is located, DNS64 converts the IPv4 A record into an IPv6 AAAA record using the NAT64 prefix. This gives the impression to the IPv6-only host that it can communicate with a server using IPv6.

4. Now DNS resolution request for www.example.com is sent to DNS64 server. It first looks up in its IPv6 AAAA record table but it does not find any IPv6 AAAA record because this website server belongs to Ipv4 address. After that, it looks in its IPv4 database and it finds IPv4 address matched to this website. Now DNS64 server can convert this IPv4 address into IPv6 address by converting this IPv4 address into hex and prepending NAT64 prefix to it. By doing so, this can give impression to IPv6 only host that it can communicate with web server using IPv6.

5. The packets gets routed in the IPv6 only network towards device doing NAT64 with the help of NAT64 prefix that was prepended to hex value of IPv4 address.

6. The NAT64 router advertises the NAT64 prefix into the IPv6-only network along with performing the translation between the IPv6-only and IPv4-only networks.

7. Once packet hits device doing NAT64 translation, the packets can be matched against ACL that you have configured for Nat64. If packets match this ACL, then packet can be translated using NAT64 further. If packet does not match configured ACL, then it can be routed using normal IPv6 routing towards its destination.

8. 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.

9. 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.

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 these methods:

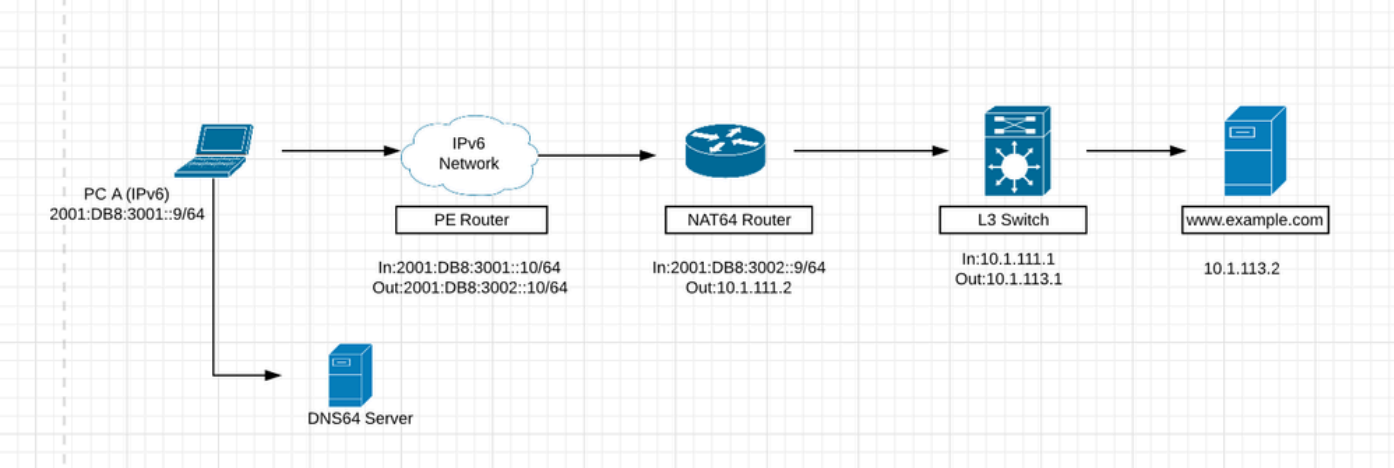
1. Extracting the destination IPv4 address by stripping the prefix from the IPv6 address. The source address is replaced by the allocated IPv4 address (and port).
2. The rest of the fields are translated from IPv6-to-IPv4 to form a valid IPv4 packet.

10. 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.

11.The return traffic and the subsequent traffic of the IPv6 packet flow can use this session database entry for translation.

- For NAT64 to work, there can be reachability to ipv6 address of interface which is in ipv6 network from the ipv6 and also the reachability can be from NAT64 router to ipv4 address of the server.

Guide to Configure NAT64



Step 1. Host A is an IPv6-only host that wants to communicate with the server www.example.com. This triggers a DNS query (AAAA: www.example.com) to the DNS64 server. The DNS64 is a key component to this process. A DNS64 server is both a DNS server for IPv6 and IPv4. It creates the illusion for the client that IPv4 servers can be reached using an IPv6 address.

Host A sends a DNS query (AAAA: www.example.com) to the DNS64 server. As far as host A is concerned, this is a normal DNS AAAA query for an IPv6 server.

Step 2. The DNS64 server receives the DNS AAAA query from host A. In an attempt to resolve the domain name, the DNS64 server sends a query to the DNS AAAA authoritative server for www.example.com.

Step 3. The IPv6 DNS AAAA authoritative server returns a response indicating that it does not have a AAAA resource record for www.example.com.

Step 4. On receiving an empty answer (name error) response to the AAAA query, this triggers the DNS64 server to send an A query (A: www.example.com) to the IPv4 DNS A authoritative server.

Step 5. The IPv4 DNS A authoritative server does have an A resource record for www.example.com and returns a response with the IPv4 address for the server (A: www.example.com 10.1.113.2).

Step 6. The DNS64 server receives the IPv4 address from the DNS A authoritative server and synthesizes a AAAA record by prefixing the address with its NAT64 prefix, 2800:1503:2000:1:1::/96, and converts the IPv4 address to hexadecimal, 0a01:7102.This address can be used by host A as the destination IPv6 address for reaching the www.example.com server.

Step 7. The synthesized AAAA record is completely transparent to host A. To host A, it appears as if www.example.com is reachable over the IPv6 network and Internet. Host A now has the addressing information necessary to transmit IPv6 packets to www.example.com with these:

- IPv6 destination address: 2800:1503:2000:1:1::0a01:7102
- IPv6 source address: 2001:DB8:3001::9

Step 8. The NAT64 router receives the IPv6 packet sent by host A on its NAT64-enabled interface .It matches the incoming packets to configured ACL. If match is not found then the packet is forwarded untranslated using normal IPv6 routing. If match is found then the packet undergoes this translation:

- The IPv6 header is translated into an IPv4 header.
- The IPv6 destination address is translated into an IPv4 address by removing the IPv6 stateful NAT64 prefix 2800:1503:2000:1:1::/96 . The lower 32 bits of the IPv6 address, 0a01:7102, are represented as the dotted-decimal IPv4 address 10.1.113.2.
- The IPv6 source address is translated into an IPv4 address using the configured IPv4 address pool. Depending upon the NAT64 configuration, this can be either a 1:1 address translation or use IPv4 address overloading. This is similar to NAT for IPv4. In this scenario, host A's source IPv6 address is translated to the IPv4 address.
- Stateful NAT64 IP address translation states are created for both the source and destination addresses. These states are created the first time the translation is performed on the packet. This state is maintained for subsequent packets in the flow. The state ends when the traffic and the state maintenance timer expire.

```
HUB-BR-1#sh nat64 translations

Proto  Original IPv4          Translated IPv4
       Translated IPv6        Original IPv6
------------------------------------------------------------------------

icmp   10.1.113.2:2654        [2800:1503:2000:1:1:0:a01:7102]:2654
       50.50.50.50:2654       [2001:db8:3001::9]:2654

Total number of translations: 1
```

Step 9. After the NAT64 translation, the translated IPv4 packet is forwarded using the normal IPv4 route lookup process. In this scenario, the IPv4 destination address 10.1.113.2 is used to forward the packet.

Step 10. The www.example.com server at 10.1.113.2 replies, which is ultimately received by the NAT64 router.

Step 11. The NAT64 router receives the IPv4 packet from the www.example.com server on one of its NAT64-enabled interfaces. The router examines the IPv4 packet to determine whether a NAT64 translation state exists for the IPv4 destination address. If a translation state does not exist, the packet is discarded. If a translation state does exist for the IPv4 destination address, the NAT64 router performs these tasks:

- The IPv4 header is translated into an IPv6 header.
- The IPv4 source address is translated into an IPv6 source address using the existing NAT64 translation state. In this scenario, the source address is translated from an IPv4 address of 10.1.113.2 to the IPv6 address 2800:1503:2000:1:1::0a01:7102. The destination address is translated from an IPv4 address to 2001:DB8:3001::9.

Step 12. After the translation, the IPv6 packet is forwarded using the normal IPv6 route lookup process.

Configuration on NAT 46 Router

1. IPv6 facing interface:

```
HUB-BR-1#sh run int gig0/0/1
Building configuration...

Current configuration : 131 bytes
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
 nat64 enable
 cdp enable
 ipv6 address 2001:DB8:3002::9/64
end
```

2. IPv4 facing interface:

```
HUB-BR-1#sh run int gig0/0/0
Building configuration...

Current configuration : 119 bytes
!
interface GigabitEthernet0/0/0
 ip address 10.1.111.2 255.255.255.0
 negotiation auto
 nat64 enable
 cdp enable
end
```

3. Create ACL matching ipv6 traffic.

```
HUB-BR-1#sh ipv6 access-list nat64acl
IPv6 access list nat64acl
    permit ipv6 2001:DB8:3001::/64 any sequence 10
HUB-BR-1#
```

4. Enable NAT64 IPv6-to-IPv4 address mapping:

#nat64 prefix stateful 2800:1503:2000:1:1::/96 ----------> Server IP can get mapped to this ipv6 ip address. You can configure any ipv6 network address here but this ipv6 network address can be reachable from your ipv6 network. Also, DNS64 server must have mapping of this ipv6 network address to server ipv4 address.

5. #nat64 v4 pool pool1 10.50.50.50 ----------> Original ipv6 source address can get translated to ips of this pool while packet can be entering into ipv4 network.

6. #nat64 v6v4 list nat64acl pool pool1 overload ---------->This can translate ipv6 addresses matching nat64acl to ipv4 address from the pool

7. Hex value of 10.1.113.2 is 0a01:7102 .Once this configuration is done, ping 2800:1503:2000:1:1::0a01:7102 address from PC A.

#ping 2800:1503:2000:1:1::0a01:7102

Verify NAT64 Details

#show nat64 translation

```
HUB-BR-1#sh nat64 translations

Proto  Original IPv4         Translated IPv4
       Translated IPv6       Original IPv6
------------------------------------------------------------------------------

icmp   10.1.113.2:7749       [2800:1503:2000:1:1:0:a01:7102]:7749
       50.50.50.50:7749      [2001:db8:3001::9]:7749

Total number of translations: 1
```

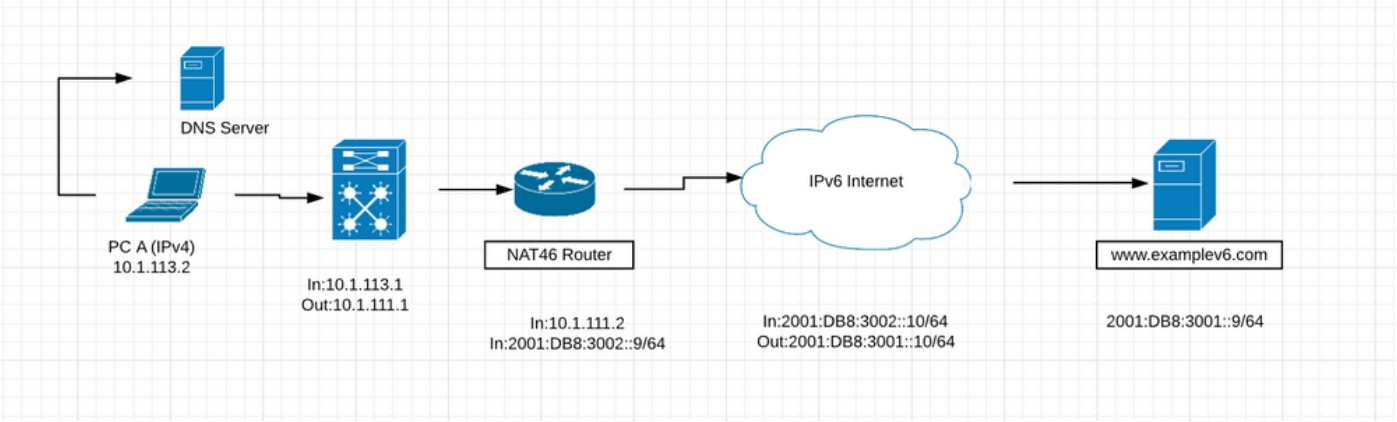
#show nat64 statistics

```
HUB-BR-1#sh nat64 statistics
NAT64 Statistics

Total active translations: 1 (0 static, 1 dynamic; 1 extended)
Sessions found: 33
Sessions created: 4
Expired translations: 4
Global Stats:
   Packets translated (IPv4 -> IPv6)
      Stateless: 0
      Stateful: 18
      MAP-T: 0
   Packets translated (IPv6 -> IPv4)
      Stateless: 0
      Stateful: 20
      MAP-T: 0

Interface Statistics
   GigabitEthernet0/0/0 (IPv4 configured, IPv6 not configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 0
         Stateful: 15
         MAP-T: 0
      Packets translated (IPv6 -> IPv4)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets dropped: 5
   GigabitEthernet0/0/1 (IPv4 not configured, IPv6 configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets translated (IPv6 -> IPv4)
         Stateless: 0
         Stateful: 20
         MAP-T: 0
      Packets dropped: 0
Dynamic Mapping Statistics
   v6v4
      access-list nat64acl pool pool1 refcount 1
         pool pool1:
            start 50.50.50.50 end 50.50.50.50
            total addresses 1, allocated 1 (100%)
            address exhaustion packet count 0
Limit Statistics
```

Scenario 2: Traffic Initiated from IPv4-only Clients to IPv6-only Servers



- The previous figure shows a scenario where clients in an IPv4-only network communicate with an IPv6-only server using NAT64. The goal is to provide access to IPv6 services transparent to the IPv4 clients. In this scenario, the DNS64 server is not required. Static mapping between the IPv6 and IPv4 address is configured on the NAT64 router.

- This scenario is unlikely for the foreseeable future. Most servers that are enabled for IPv6 can also be IPv4 capable. It is more likely that IPv6 servers can be running dual-stack for quite some time. IPv6-only servers can eventually become more common, but not anytime soon.

Guide to Configure NAT46

Step 1. The first step is to configure IPv6-to-IPv4 static mapping on NAT46 router to provide access to the IPv6 server 2001:DB8:3001::9/64 from the IPv4 address 10.1.113.2. Also, the IPv4 address 10.50.50.50 needs to be registered as a DNS resource record for www.examplev6.com on the DNS server. The static NAT64 mapping is created using this command:

NAT64-Router(config)# nat64 v6v4 static 2001:DB8:3001::9 10.50.50.50

Step 2. PC A is an IPv4-only host that wants to communicate with the server www.examplev6.com . This triggers a DNS query (A: www.examplev6.com) to its IPv4 DNS authoritative server.

Step 3. The DNS server responds with an A resource record for www.examplev6.com, 10.50.50.50.

Step 4. Host A now has the addressing information necessary to transmit IPv4 packets to www.examplev6.com with

- IPv4 destination address: 10.50.50.50
- IPv4 source address: 10.1.113.2

Step 5. The NAT64 router receives the IPv4 packet on its NAT64-enabled interface and performs these tasks:

- The IPv4 header is translated into an IPv6 header.
- The IPv4 destination address is translated into an IPv6 address using the existing NAT64 translation state created by the static configuration in Step 1. The destination IPv4 address of 10.50.50.50 is translated to the IPv6 destination address 2001:DB8:3001::9.
- The IPv4 source address is translated into an IPv6 address by adding the stateful NAT64 prefix 2800:1503:2000:1:1::/96 to the IPv4 address. This results in an IPv6 source address of 2800:1503:2000:1:1::0a01:7102. (0a01:7102 is the hexadecimal equivalent of 10.1.113.2.)

Step 6. After the translation, the IPv6 packet is routed using the normal IPv6 routing process. The packet is ultimately routed to the www.examplev6.com server at 2001:DB8:3001::9 .

Step 7. The server www.examplev6.com replies with a packet destined for host A.

Step 8. The NAT64 router receives the IPv6 packet sent by the IPv6 server on its NAT64-enabled interface and performs these tasks:

- The IPv6 header is translated into an IPv4 header.
- The IPv6 source address is translated to 10.50.50.50 using stateful translation table.
- The IPv6 destination address is translated into an IPv4 address by removing the IPv6 stateful NAT64 prefix 2800:1503:2000:1:1::/96. The lower 32 bits of the IPv6 address, 0a01:7102, are represented as the dotted-decimal IPv4 address 10.1.113.2.

Step 9. After the translation, the NAT64 router forwards the packet to 10.1.113.2 using the normal IPv4 routing process.

- Similar to the previous scenario, transparent communication is established between the IPv4-only client and the IPv6-only server using stateful NAT64. The configurations are similar except for the static mapping command discussed in Step 1.

Configuration on NAT 46 Router

1. IPv4 facing interface:

```
HUB-BR-1#sh run int gig0/0/0
Building configuration...

Current configuration : 137 bytes
!
interface GigabitEthernet0/0/0
 ip address 10.1.111.2 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 nat64 enable
 cdp enable
end
```

2. IPv6 facing interface:

```
HUB-BR-1#sh run int gig0/0/1
Building configuration...

Current configuration : 131 bytes
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
 nat64 enable
 cdp enable
 ipv6 address 2001:DB8:3002::9/64
end
```

3. Other configs needed on the router to translate traffic successfully from IPv4 to IPv6:

```
nat64 prefix stateful 2800:1503:2000:1:1::/96
nat64 v6v4 static 2001:DB8:3001::9 50.50.50.50
```

After the configuration is successful, ping 10.50.50.50 from IPv4 host.

#ping 10.50.50.50

Verifying NAT46

#show nat64 translations

```
HUB-BR-1#sh nat64 translations

Proto  Original IPv4         Translated IPv4
       Translated IPv6       Original IPv6
------------------------------------------------------------------------------

illegal ---                   ---
       50.50.50.50           2001:db8:3001::9
icmp   10.1.113.2:11         [2800:1503:2000:1:1:0:a01:7102]:11
       50.50.50.50:11        [2001:db8:3001::9]:11

Total number of translations: 2
```

#show nat46 statistics

```
HUB-BR-1#sh nat64 statistics
NAT64 Statistics

Total active translations: 2 (1 static, 1 dynamic; 1 extended)
Sessions found: 9967
Sessions created: 14
Expired translations: 14
Global Stats:
   Packets translated (IPv4 -> IPv6)
      Stateless: 0
      Stateful: 4990
      MAP-T: 0
   Packets translated (IPv6 -> IPv4)
      Stateless: 0
      Stateful: 4992
      MAP-T: 0

Interface Statistics
   GigabitEthernet0/0/0 (IPv4 configured, IPv6 not configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 0
         Stateful: 1947
         MAP-T: 0
      Packets translated (IPv6 -> IPv4)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets dropped: 58
   GigabitEthernet0/0/1 (IPv4 not configured, IPv6 configured):
      Packets translated (IPv4 -> IPv6)
         Stateless: 0
         Stateful: 0
         MAP-T: 0
      Packets translated (IPv6 -> IPv4)
         Stateless: 0
         Stateful: 1947
         MAP-T: 0
      Packets dropped: 0
Dynamic Mapping Statistics
   v6v4
Limit Statistics
```

Translation Scenarios and Their Applicability

Scenarios for IPv6/IPv4 Translation	**Applicability**	**Example**
Scenario 1: An IPv6 network to the IPv4 Internet	• IPv6-only network wanting to transparently access both IPv6 and existing IPv4 content. • Initiated from IPv6 hosts and network.	• ISPs rolling out new services and networks for IPv6-only smartphones (third-generation [3G], Long-Term Evolution [LTE], and so on) handsets. • Enterprises deploying IPv6-only network.
Scenario 2: The IPv4 Internet to an IPv6	• Servers in IPv6-only network wanting to transparently serve both IPv4 and IPv6	Upcoming or existing content providers rolling out services in IPv6-only

network	users. • Initiated from IPv4 hosts and network.	environment.
Scenario 3: The IPv6 Internet to an IPv4 network	• Servers in existing IPv4-only network wanting to serve IPV6 Internet users. • Initiated from IPv6 hosts and network.	Existing content providers migrating to IPv6 and thus wanting to offer services to IPv6 Internet users as part of coexistence strategy.
Scenario 4: An IPv4 network to the IPv6 Internet	Not a viable case in the near future; this scenario can probably occur only some time after the early stage of the IPv6/IPv4 transition.	None
Scenario 5: An IPv6 network to an IPv4 network	Both an IPv4 network and an IPv6 network are within the same organization.	Similar to scenario 1, catering to Intranet instead of Internet.
Scenario 6: An IPv4 network to an IPv6 network	Both an IPv4 network and an IPv6 network are within the same organization.	Similar to scenario 2, catering to intranet instead of Internet.
Scenario 7: The IPv6 Internet to the IPv4 Internet	Would suffer from poor throughput.	None
Scenario 8: The IPv4 Internet to the IPv6 Internet	No viable translation technique to handle unlimited IPv6 address translation.	None

Important Troubleshooting Commands in Case There are Issues During NAT64 Implementation

#show platform hardware qfp active statistics drop (to see if there are any NAT64 drops)

#show running-config | include nat64 (to see if everything is configures on Cisco IOS®)

#show platform hardware qfp active feature nat64 datapath statistics (to check the reason for drop counter)

#show platform hardware qfp active feature nat64 datapath pool (to check the pool is configured properly)

#show platform hardware qfp active feature nat64 datapath map (to check and see pool to mapping config is done properly)

#show platform software object-manager F0 pending-ack-update (to check if there are any pending objects)