# Configuring Bridge Domain Interfaces

The Cisco ASR 1000 Series Aggregation Services Routers support the bridge domain interface (BDI) feature for packaging Layer 2 Ethernet segments into Layer 3 IP address.



## Restrictions for Bridge Domain Interfaces

The following are the restrictions pertaining to bridge domain interfaces:

- Only 4096 bridge domain interfaces are supported per system.
- For a bridge domain interface, the maximum transmission unit (MTU) size can be configured between 1500 and 9216 bytes.
- Bridge domain interfaces support only the following features:
  - IPv4 Multicast
  - QoS marking and policing. Shaping and queuing are not supported
  - IPv4 and IPv6 VRF
  - IPv4 and IPv6 unicast forwarding
  - Dynamic routing such as BGP, OSPF, EIGRP, RIP, IS-IS, and STATIC
  - Hot Standby Router Protocol (HSRP) from IOS XE 3.8.0 onwards.
  - Virtual Router Redundancy Protocol (VRRP) from IOS XE 3.8.0 onwards.
  - Flexible NetFlow

**Note** Flexible NetFlow is supported from Cisco IOS XE 17.7.1a and later releases.

- Bridge domain interfaces do not support the following features:
  - PPP over Ethernet (PPPoE)
  - Bidirectional Forwarding Detection (BFD) protocol
  - QoS
  - Network-Based Application Recognition (NBAR) or Advanced Video Coding (AVC)

**Note** NAT is supported from XE16.2.1 and later releases.

# Information About Bridge Domain Interface

Bridge domain interface is a logical interface that allows bidirectional flow of traffic between a Layer 2 bridged network and a Layer 3 routed network traffic. Bridge domain interfaces are identified by the same index as the bridge domain. Each bridge domain represents a Layer 2 broadcast domain. Only one bridge domain interface can be associated with a bridge domain.

Bridge domain interface supports the following features:

- IP termination
- Layer 3 VPN termination
- Address Resolution Protocol (ARP), G-ARP, and P-ARP handling
- MAC address assignment

Prior to configuring a bridge domain interface, you must understand the following concepts:

- Ethernet Virtual Circuit Overview
- Bridge Domain Interface Encapsulation
- Assigning a MAC Address
- Support for IP Protocols
- Support for IP Forwarding
- Packet Forwarding
- Bridge Domain Interface Statistics

## Ethernet Virtual Circuit Overview

An Ethernet Virtual Circuit (EVC) is an end-to-end representation of a single instance of a Layer 2 service that is offered by a provider. It embodies the different parameters on which the service is being offered. In the Cisco EVC Framework, the bridge domains are made up of one or more Layer 2 interfaces known as service instances. A service instance is the instantiation of an EVC on a given port on a given router. Service instance is associated with a bridge domain based on the configuration.

An incoming frame can be classified as service instance based on the following criteria:

- Single 802.1Q VLAN tag, priority-tagged, or 802.1ad VLAN tag
- Both QinQ (inner and outer) VLAN tags, or both 802.1ad S-VLAN and C-VLAN tags
- Outer 802.1p CoS bits, inner 802.1p CoS bits, or both
- Payload Ethernet type (five choices are supported: IPv4, IPv6, PPPoE-all, PPoE-discovery, and PPPoE-session)

Service instance also supports alternative mapping criteria:

- Untagged—Mapping to all the frames lacking a 802.1Q or 802.1ad header
- Default—Mapping to all the frames

For more information on the EVC architecture, see the section *Configuring Ethernet Virtual Connections on the Cisco ASR 1000 Router* in the Carrier Ethernet Configuration Guide .

# Bridge Domain Interface Encapsulation

Security Group classification includes both Source and Destination Group, which is specified by source SGT and DGT. SGT Based PBR feature provides the PBR route-map match clause for SGT/DGT based packet classification. SGT Based PBR feature supports configuration of unlimited number of tags, but it is recommended to configure the tags based on memory available in the platform.

An EVC provides the ability to employ different encapsulations on each Ethernet flow point (EFP) present in a bridge domain. A BDI egress point may not be aware of the encapsulation of an egress packet because the packet may have egressed from one or more EFPs with different encapsulations.

In a bridge domain, if all the EFPs have different encapsulations, the BDI must be untagged (using the no 802.1Q tag). Encapsulate all the traffic in the bridge domain (popped or pushed) at the EFPs. Configure rewrite at each EFP to enable encapsulation of the traffic on the bridge domain.

In a bridge domain, if all the EFPs have the same encapsulation, configure the encapsulations on the BDI using the encapsulation command. Enabling encapsulation at the BDI ensures effective pushing or popping of tags, thereby eliminating the need for configuring the rewrite command at the EFPs. For more information on configuring the encapsulations on the BDI, see the How to Configure a Bridge Domain Interface.

# Assigning a MAC Address

All the bridge domain interfaces on the Cisco ASR 1000 chassis share a common MAC address. The first bridge domain interface on a bridge domain is allocated a MAC address. Thereafter, the same MAC address is assigned to all the bridge domain interfaces that are created in that bridge domain.

**Note** You can configure a static MAC address on a bridge domain interface using the **mac-address** command.

# Support for IP Protocols

Bridge domain interfaces enable the Cisco ASR 1000 Series Aggregation Services Routers to act as a Layer 3 endpoint on the Layer 2 bridge domain for the following IP-related protocols:

- ARP
- DHCP
- HTTP
- ICMP
- NTP
- RARP
- SNMP
- TCP
- Telnet
- TFTP
- UDP

# Support for IP Forwarding

Bridge domain interface supports the following IP forwarding features:

- IPv4 input and output access control lists (ACL)
- IPv4 input and output QoS policies. The operations supported for the input and output service policies on a bridge domain interface are:
  - Classification
  - Marking
  - Policing
- IPv4 L3 VRFs

# Packet Forwarding

A bridge domain interface provides bridging and forwarding services between the Layer 2 and Layer 3 network infrastructure.

## Layer 2 to Layer 3

During a packet flow from a Layer 2 network to a Layer 3 network, if the destination MAC address of the incoming packet matches the bridge domain interface MAC address, or if the destination MAC address is a multicast address, the packet or a copy of the packet is forwarded to the bridge domain interface.

**Note** MAC address learning cannot not be performed on the bridge domain interface.

## Layer 3 to Layer 2

When a packet arrives at a Layer 3 physical interface of a router, a route lookup action is performed. If route lookup points to a bridge domain interface, then the bridge domain interface adds the layer 2 encapsulation and forwards the frame to the corresponding bridge domain. The byte counters are updated.

During a Layer 2 lookup on a bridge domain to which the bridge domain interface belongs, the bridge domain forwards the packets to the correct service instance based on the destination MAC address.

# Link States of a Bridge Domain and a Bridge Domain Interface

Bridge domain interface acts as a routable IOS interface on Layer 3 and as a port on a bridge domain. Both bridge domain interfaces and bridge domains operate with individual administrative states.

Shutting down a bridge domain interface stops the Layer 3 data service, but does not override or impact the state of the associated bridge domain.

Shutting down a bridge domain stops Layer 2 forwarding across all the associated members including service instances and bridge domain interfaces. The associated service instances influence the operational state of a bridge domain. Bridge domain interface cannot be operational unless one of the associated service instances is up.

**Note** Because a bridge domain interface is an internal interface, the operational state of bridge domain interface does not affect the bridge domain operational state.

## BDI Initial State

The initial administrative state of a BDI depends on how the BDI is created. When you create a BDI at boot time in the startup configuration, the default administrative state for the BDI is up. It will remain in this state unless the startup configuration includes the shutdown command. This behavior is consistent with all the other interfaces. When you create a BDI dynamically at command prompt, the default administrative state is down.

## BDI Link State

A BDI maintains a link state that comprises of three states: administratively down, operationally down, and up. The link state of a BDI is derived from two independent inputs: the BDI administrative state set by the corresponding users and the fault indication state from the lower levels of the interface states. It defines a BDI link state based on the state of the two inputs.

| **Fault Indication State** | **BDI Admin** | |
|---|---|---|
| {start emdash} {end emdash} | **Shutdown** | **No Shutdown** |
| **No faults asserted** | Admin-down | Up |
| **At least one fault asserted** | Admin-down | Operationally-Down |

# Bridge Domain Interface Statistics

For virtual interfaces, such as the bridge domain interface, protocol counters are periodically queried from the QFP.

When packets flow from a Layer 2 bridge domain network to a Layer 3 routing network through the bridge domain interface, the packets are treated as bridge domain interface input packets and bytes. When packets arrive at a Layer 3 interface and are forwarded through the bridge domain interface to a Layer 2 bridge domain, the packets are treated as output packets and bytes, and the counters are updated accordingly.

A BDI maintains a standard set of Layer 3 packet counters as the case with all Cisco IOS interfaces. Use the show interface command to view the Layer 3 packet counters.

The convention of the counters is relative to the Layer 3 cloud. For example, input refers to the traffic entry to the Layer 3 cloud from the Layer 2 BD, while output refers to the traffic exit from the Layer 3 cloud to the Layer 2 BD.

Use the **show interfaces accounting** command to display the statistics for the BDI status. Use the **show interface** <*if-name*> command to display the overall count of the packets and bytes that are transmitted and received.

# Creating or Deleting a Bridge Domain Interface

When you define an interface or subinterface for a Cisco IOS router, you name it and specify how it is assigned an IP address. You can create a bridge domain interface before adding a bridge domain to the system. This new bridge domain interface will be activated after the associated bridge domain is configured.

**Note** When a bridge domain interface is created, a bridge domain is automatically created.

When you create the bridge domain interface and the bridge domain, the system maintains the required associations for mapping the bridge domain-bridge domain interface pair.

The mapping of bridge domain and bridge domain interface is maintained in the system. The bridge domain interface uses the index of the associated bridge domain to show the association.

# Bridge Domain Interface Scalability

The following table lists the bridge domain interface scalability numbers, based on the type of Cisco ASR 1000 Series Aggregation Services Router's Forwarding Processors.

***Table 1: Bridge Domain Interface Scalability Numbers Based on the Type of Cisco ASR 1000 Series Aggregation Services Router's Forwarding Processor***

| Description | ASR1000-ESP5,ASR 1001,ASR 1002-F (ESP2.5) | ASR1000-ESP10,ASR1000-ESP10-N,ASR1000-ESP20 | ASR1000-ESP40 |
|---|---|---|---|
| Maximum bridge domain interfaces per router | 4096 | 4096 | 4096 |

# Bridge-Domain Virtual IP Interface

The Virtual IP Interface (VIF) feature helps to associate multiple BDI interfaces with a BD instance. The BD-VIF interface inherits all the existing L3 features of IOS logical IP interface.

**Note** You must configure every BD-VIF interface with a unique MAC address and it should belong to a different VRF.

The Virtual IP Interface (VIF) feature has the following limitations:

- BD-VIF interface does not support IP multicast.
- Number of BD-VIF interfaces with automatically generated MAC address varies on the basis of platforms.
- BD-VIF Interface does not support MPLS.
- The maximum number of BD-VIF interfaces per bridge-domain and the total number of BD-VIF interface for per system vary based on the type of platforms.

The maximum number of BD-VIF supported on different platforms varies:

- ASR 1000 supports maximum 100 BD-VIF for a Bridge Domain
- CSR 1000v supports maximum 16 BD-VIF for a Bridge Domain
- ISR 4000 support maximum 16 BD-VIF for a Bridge Domain

From Cisco IOS XE 17.7.1a release, BD-VIF supports Flexible Netflow (FNF).

# How to Configure a Bridge Domain Interface

To configure a bridge domain interface, perform the following steps:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface BDI** *{interface number}*
4. **encapsulation** *encapsulation dot1q <first-tag> [second-dot1q <second-tag>]*
5. Do one of the following:
6. **match security-group destination tag** *sgt-number*
7. **mac address** *{mac-address}*
8. **no shut**
9. **shut**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode. Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface BDI** *{interface number}*<br>**Example:**<br>`Router(config-if)# interface BDI3` | Specifies a bridge domain interface. |
| **Step 4** | **encapsulation** *encapsulation dot1q <first-tag> [second-dot1q <second-tag>]*<br>**Example:**<br>`Router(config-if)# encapsulation dot1Q 1 second-dot1q 2` | Defines the encapsulation type.<br>The example shows how to define dot1q as the encapsulation type. |
| **Step 5** | Do one of the following:<br>**Example:**<br>**ip address** *ip-address mask*<br>**Example:**<br>**Example:**<br>**ipv6 address** *{X:X:X:X::X* **link-local** \| *X:X:X:X::X/prefix [***anycast** \| **eui-64***]* \| **autoconfig** *[***default***]}*<br>**Example:**<br>`Router(config-if)# ip address 10.2.2.1 255.255.255.0`<br>**Example:**<br>**Example:**<br>`Router(config-if)# ipv6 address AB01:CD1:123:C::/64 eui-64` | Specifies either the IPv4 or IPv6 address for the bridge domain interface. |
| **Step 6** | **match security-group destination tag** *sgt-number*<br>**Example:** | Configures the value for security-group destination security tag. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config-route-map)# match security-group destination tag 150` | |
| **Step 7** | **mac address** *{mac-address}*<br>**Example:**<br>`Router(config-if)# mac-address 1.1.3` | Specifies the MAC address for the bridge domain interface. |
| **Step 8** | **no shut**<br>**Example:**<br>`Router(config-if)# no shut` | Enables the bridge domain interface. |
| **Step 9** | **shut**<br>**Example:**<br>`Router(config-if)# shut` | Disables the bridge domain interface. |

## Example

The following example shows the configuration of a bridge domain interface at IP address 10.2.2.1 255.255.255.0:

```
Router# configure terminal
Router(config)# interface BDI3
Router(config-if)# encapsulation dot1Q 1 second-dot1q 2
Router(config-if)# ip address 10.2.2.1 255.255.255.0
Router(config-if)# mac-address 1.1.3
Router(config-if)# no shut
Router(config-if)# exit
```

# Displaying and Verifying Bridge Domain Interface Configuration

### SUMMARY STEPS

1. **enable**
2. **show interfaces bdi**
3. **show platform software interface fp active name**
4. **show platform hardware qfp active interface if-name**
5. **debug platform hardware qfp feature**
6. **platform trace runtime process forwarding-manager module**
7. **platform trace boottime process forwarding-manager module interfaces**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode. Enter your password, if prompted. |
| **Step 2** | **show interfaces bdi**<br>**Example:**<br>`Router# show interfaces BDI3` | Displays the configuration summary of the corresponding BDI. |
| **Step 3** | **show platform software interface fp active name**<br>**Example:**<br>`Router# show platform software interface fp active name BDI4` | Displays the bridge domain interface configuration in a Forwarding Processor. |
| **Step 4** | **show platform hardware qfp active interface if-name**<br>**Example:**<br>`Router# show platform hardware qfp active interface if-name BDI4` | Displays the bridge domain interface configuration in a data path. |
| **Step 5** | **debug platform hardware qfp feature**<br>**Example:**<br>`Router# debug platform hardware qfp active feature l2bd client all` | The selected CPP L2BD Client debugging is on. |
| **Step 6** | **platform trace runtime process forwarding-manager module**<br>**Example:**<br>`Router(config)# platform trace runtime slot F0 bay 0 process forwarding-manager module interfaces level info` | Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Forwarding Manager process. |
| **Step 7** | **platform trace boottime process forwarding-manager module interfaces**<br>**Example:**<br>`Router(config)# platform trace boottime slot R0 bay 1 process forwarding-manager forwarding-manager level max` | Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Route Processor Forwarding Manager process during bootup. |

### What to do next

For additional information on the commands and the options available with each command, see the Cisco IOS Configuration Fundamentals Command Reference Guide.

# Configuring Bridge-Domain Virtual IP Interface

```
enable
configure terminal
[no] interface BD-VIF interface-number
  [ [no] vrf forwarding vrf-name]
  [ [no] mac address mac-address]
  [ [no] ip address ip-address mask]
  [ [no] ipv6 address {X:X:X:X::X link-local| X:X:X:X::X/prefix [anycast | eui-64] |
autoconfig [default]}]

exit
```

To delete BD-VIF interface, use the 'no' form of the command.

# Associating VIF Interface with a Bridge Domain

```
enable
configure terminal
bridge-domain bridge-domain number
[no] member BD-VIF interface-number
exit
```

To dissociate the VIF interface, use the 'no' form of the command.

# Verifying Bridge-Domain Virtual IP Interface

All existing show commands for interface and IP interface can be used for the BD-VIF interface.

show interface bd-vif *bd-vif-id*

show ip interface bd-vif *bd-vif-id*

show bd-vif interfaces in fman-fp

show pla sof inter fp ac brief | i BD_VIF

# Example Configuration Bridge-Domain Virtual IP Interface

```
Detail sample:

interface Port-channel1
mtu 9000
no ip address
 !Ethernet service endpoint one per neutron network
service instance 1756 ethernet
  description 4e8e5957-649f-477b-9e5b-f1f75b21c03c
  encapsulation dot1q 1756
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1756
!
```

```
interface BD-VIF5001
no shutdown
vrf forwarding vrf5001
ip address 10.0.0.1 255.255.255.0
interface BD-VIF5002
no shutdown
vrf forwarding vrf5002
ip address 10.0.0.2 255.255.255.0

bridge-domain 1756
member Port-channel1 service-instance 1756
member bd-vif5001
member bd-vif5002
```

# Configuring Flexible NetFlow over a Bridge Domain Virtual IP Interface

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip | ipv6**}**flow monitor** *monitor-name* [**sampler** *sampler-name*] {**input | output**}
5. **exit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode. Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *type number*<br>**Example:**<br>`Device (config)# interface BD-VIF 100` | Specifies an interface and enters interface configuration mode. Enter the BD-VIF number. |
| **Step 4** | {**ip | ipv6**}**flow monitor** *monitor-name* [**sampler** *sampler-name*] {**input | output**}<br>**Example:**<br>`Device(config-if)# ip flow monitor FLOW-MONITOR-1 input` | Enables a Flexible NetFlow flow monitor for IP traffic that the router is receiving or transmitting on the interface. |
| **Step 5** | **exit**<br>**Example:**<br>`Device(config-if)# exit` | Exits interface configuration mode and returns to privileged EXEC mode. |

# Examples: Flexible NetFlow over a Bridge Domain Virtual IP Interface

The following is a sample output for the **show platform hardware qfp active interface if-name** command showing the QFP information and flow direction for flow monitors. The table below provides the key to the CLI output.

| Configuration | Output |
|---|---|
| ip flow monitor <monitor-name> input | IPV4_INPUT_FNF_FIRST<br>IPV4_INPUT_FNF_FINAL |
| ip flow monitor <monitor-name> output | IPV4_BDI_OUTPUT_FNF_FINAL |
| ipv6 flow monitor <monitor-name> input | IPV6_INPUT_FNF_FIRST<br>IPV6_INPUT_FNF_FINAL |
| ipv6 flow monitor <monitor-name> output | IPV6_BDI_OUTPUT_FNF_FINAL |

```
Device# show run interface bd-vif2
Building configuration...

Current configuration: 227 bytes
!
interface BD-VIF2
vrf forwarding vrf1
ip flow monitor test1 input
ip flow monitor test1 output
ip address 10.11.11.11 255.255.255.0
ipv6 flow monitor test2 input
ipv6 flow monitor test2 output
ipv6 address 2001:DB8::1/32
end

Device# show platform hardware qfp active interface if-name BD-VIF 2
General interface information
  Interface Name: BD-VIF2
  Interface state: VALID
  Platform interface handle: 20
  QFP interface handle: 17
  Rx uidb: 262138
  Tx uidb: 262127
  Channel: 0
Interface Relationships

BGPPA/QPPB interface configuration information
  Ingress: BGPPA/QPPB not configured. flags: 0000
  Egress: BGPPA not configured. flags: 0000

ipv4_input enabled.
ipv4_output enabled.
ipv6_input enabled.
ipv6_output enabled.
layer2_input enabled.
layer2_output enabled.
ess_ac_input enabled.

Features Bound to Interface:
2 GIC FIA state
66 PUNT INJECT DB
70 cpp_l2bd_svr
```

```
43 icmp_svr
45 ipfrag_svr
46 ipreass_svr
47 ipv6reass_svr
44 icmp6_svr
58 stile
Protocol 0 - ipv4_input
FIA handle - CP:0x55a7f59df038 DP:0x3fff1000
  IPV4_INPUT_DST_LOOKUP_ISSUE (M)
  IPV4_INPUT_ARL_SANITY (M)
  IPV4_INPUT_SRC_LOOKUP_ISSUE
  IPV4_INPUT_DST_LOOKUP_CONSUME (M)
  IPV4_INPUT_SRC_LOOKUP_CONSUME
  IPV4_INPUT_FOR_US_MARTIAN (M)
  IPV4_INPUT_STILE_LEGACY
  IPV4_INPUT_FNF_FIRST
  IPV4_INPUT_LOOKUP_PROCESS (M)
  IPV4_INPUT_FNF_FINAL
  IPV4_INPUT_IPOPTIONS_PROCESS (M)
  IPV4_INPUT_GOTO_OUTPUT_FEATURE (M)
Protocol 1 - ipv4_output
FIA handle - CP:0x55a7f59df0d8 DP:0x3ffeff00
  IPV4_VFR_REFRAG (M)
  IPV4_OUTPUT_SRC_LOOKUP_ISSUE
  IPV4_OUTPUT_L2_REWRITE (M)
  IPV4_OUTPUT_SRC_LOOKUP_CONSUME
  IPV4_OUTPUT_STILE_LEGACY
  IPV4_OUTPUT_FRAG (M)
  IPV4_BDI_OUTPUT_FNF_FINAL.
  BDI_VLAN_TAG_ATTACH_AND_LAYER2_LOOKUP_GOTO
  LAYER2_BRIDGE
  BDI_OUTPUT_GOTO_OUTPUT_FEATURE
  IPV4_OUTPUT_DROP_POLICY (M)
  DEF_IF_DROP_FIA (M)
Protocol 6 - ipv6_input
FIA handle - CP:0x55a7f59dee58 DP:0x3fff4300
  IPV6_INPUT_SANITY_CHECK (M)
  IPV6_INPUT_DST_LOOKUP_ISSUE (M)
  IPV6_INPUT_SRC_LOOKUP_ISSUE
  IPV6_INPUT_ARL (M)
  IPV6_INPUT_DST_LOOKUP_CONT (M)
  IPV6_INPUT_SRC_LOOKUP_CONT
  IPV6_INPUT_DST_LOOKUP_CONSUME (M)
  IPV6_INPUT_SRC_LOOKUP_CONSUME
  IPV6_INPUT_STILE_LEGACY
  IPV6_INPUT_FNF_FIRST
  IPV6_INPUT_FOR_US (M)
  IPV6_INPUT_LOOKUP_PROCESS (M)
  IPV6_INPUT_FNF_FINAL
  IPV6_INPUT_LINK_LOCAL_CHECK (M)
  IPV6_INPUT_GOTO_OUTPUT_FEATURE (M)
Protocol 7 - ipv6_output
FIA handle - CP:0x55a7f59dee08 DP:0x3fff4b80
  IPV6_VFR_REFRAG (M)
  IPV6_OUTPUT_SRC_LOOKUP_ISSUE
  IPV6_OUTPUT_SRC_LOOKUP_CONT
  IPV6_OUTPUT_SRC_LOOKUP_CONSUME
  IPV6_OUTPUT_L2_REWRITE (M)
  IPV6_OUTPUT_STILE_LEGACY
  IPV6_OUTPUT_FRAG (M)
  IPV6_BDI_OUTPUT_FNF_FINAL
  BDI_VLAN_TAG_ATTACH_AND_LAYER2_LOOKUP_GOTO
  LAYER2_BRIDGE
  BDI_OUTPUT_GOTO_OUTPUT_FEATURE
```

```
  IPV6_OUTPUT_DROP_POLICY (M)
  DEF_IF_DROP_FIA (M)
```

The following is a sample out of the **show flow monitor** [[**name**] [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]] command showing the cache output in record format.

```
Device# show flow monitor name FLOW-MONITOR-1 cache format record

Cache type:  Normal
Cache size:  1000
Current entries:  4
High Watermark:  4
Flows added:  101
Flows aged:  97
- Active timeout (1800 secs) 3
- Inactive timeout  (15 secs) 94
- Event aged  0
- Watermark aged  0
- Emergency aged
IPV4 DESTINATION ADDRESS:
198.51.100.1 0
ipv4 source address: 10.10.11.1
trns source port: 25
trns destination port: 25
counter bytes: 72840
counter packets: 1821
IPV4 DESTINATION ADDRESS: 198.51.100.2
ipv4 source address: 10.10.10.2
trns source port: 20
trns destination port: 20
counter bytes: 3913860
counter packets: 7326
IPV4 DESTINATION ADDRESS: 198.51.100.200
ipv4 source address: 192.168.67.6
trns source port: 0
trns destination port: 3073
counter bytes: 51072
counter packets: 1824


Device# show flow monitor name FLOW-MONITOR-2 cache format record

Cache type:  Normal
Cache size: 1000
Current entries: 2
High Watermark: 3
Flows added: 95
Flows aged: 93
- Active timeout (1800 secs) 0
- Inactive timeout (15 secs) 93
- Event aged 0
- Watermark aged 0
- Emergency aged 0
IPV6 DESTINATION ADDRESS: 2001:DB8:0:ABCD::1
ipv6 source address: 2001:DB8:0:ABCD::2
trns source port: 33572
trns destination port: 23
counter bytes: 19140
counter packets: 349
IPV6 DESTINATION ADDRESS: FF02::9
ipv6 source address: 2001:DB8::A8AA:BBFF:FEBB

trns source port: 521
trns destination port: 521
```

```
counter bytes: 92
counter packets: 1
```

The following is a sample out of the **show flow interface** command showing the flow status for an interface.

```
Device# show flow interface BD-VIF2001

Interface GigabitEthernet0/0/0
FNF: monitor: FLOW-MONITOR-1
direction: Input
traffic(ip): on
FNF: monitor: FLOW-MONITOR-2
direction:    Input traffic(ipv6): on

Device# show flow interface BD-VIF2002

Interface GigabitEthernet1/0/0
FNF: monitor: FLOW-MONITOR-1
direction: Output
traffic(ip): on
FNF: monitor: FLOW-MONITOR-2
direction:    Input traffic(ipv6): on
```

The following is a sample output of the **show platform hardware qfp active interface if-name** | **in FNF** command showing the QFP information and flow direction for flow monitors in Flexible NetFlow configuration. The table below provides the key to the CLI output.

| Configuration | Output |
|---|---|
| ip flow monitor <monitor-name> input | IPV4_INPUT_FNF_FIRST<br>IPV4_INPUT_FNF_FINAL |
| ip flow monitor <monitor-name> output | IPV4_BDI_OUTPUT_FNF_FINAL |
| ipv6 flow monitor <monitor-name> input | IPV6_INPUT_FNF_FIRST<br>IPV6_INPUT_FNF_FINAL |
| ipv6 flow monitor <monitor-name> output | IPV6_BDI_OUTPUT_FNF_FINAL |

```
Device# show run interface bd-vif2
Building configuration...

Current configuration : 227 bytes
!
interface BD-VIF2
vrf forwarding vrf1
ip flow monitor test1 input
ip flow monitor test1 output
ip address 10.11.11.11 255.255.255.0
ipv6 flow monitor test2 input
ipv6 flow monitor test2 output
ipv6 address 2001::8/64
end
 Device# show platform hardware qfp active interface if-name BD-VIF 2  | in FNF
  IPV4_INPUT_FNF_FIRST
  IPV4_INPUT_FNF_FINAL
  IPV4_BDI_OUTPUT_FNF_FINAL.
  IPV6_INPUT_FNF_FIRST
  IPV6_INPUT_FNF_FINAL
  IPV6_BDI_OUTPUT_FNF_FINAL
```

The **clear flow monitor name** *monitor-name* [**cache** [**force-export**] | **force-export** | **statistics**] command clears a Flexible NetFlow flow monitor, flow monitor cache, or flow monitor statistics, and can be used to force the export of the data in the flow monitor cache.

For more details on configuring Flexible NetFlow, see the Flexible NetFlow Configuration Guide, Cisco IOS XE 17.

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Configuring Ethernet Virtual Connections on the Cisco ASR 1000 Series Aggregation Services Routers | Carrier Ethernet Configuration Guide |
| EVC Quality of Service | http://www.cisco.com/en/US/docs/ios/ios_xe/qos/configuration/guide/qos_evc_xe.html |

### 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:<br>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. | https://www.cisco.com/c/en_in/support/index.html |

# Feature Information for Configuring Bridge Domain Interfaces

The following table lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

**Note** The table below 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.

***Table 2: Feature Information for Configuring Bridge Domain Interfaces***

| Feature Name | Releases | Feature Information |
|---|---|---|
| Configuring Bridge Domain Interface | Cisco IOS XE 3.2.0S | This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. |
| Configuring Bridge Domain Interface | Cisco IOS XE 3.7.0S | This feature was updated on the Cisco ASR 1000 Series Aggregation Services Routers. The following section was updated for this feature:<br>Information About Bridge Domain Interface, on page 2 |
| Bridge-Domain Virtual IP Interface | Cisco IOS XE Gibraltar 16.12 | The Bridge-Domain Virtual IP Interface (VIF) now connects multiple Bridge Domain Interfaces (BDI) with a single BD instance so that each IP subnet within an L2 network can be associated with a single VRF. |
| Flexible NetFlow (FNF) on Bridge-Domain Virtual IP Interface (BD-VIF) | Cisco IOS XE Cupertino 17.7.1a | This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced:<br>**{ip \| ipv6} flow monitor** *monitor-name* **[sampler** *sampler-name***] {input \| output}** |