Integrated Routing and Bridging (IRB) Support for the Cisco MGX-RPM-XF-512

The Integrated Routing and Bridging (IRB) Support for the Cisco MGX-RPM-XF-512 feature enables the Cisco MGX8850 and Cisco MGX8950 platforms that have installed Route Processor Modules (RPMs) to connect to a bridged network.

The feature allows IP routing between routed interfaces and bridge groups or between bridge groups. Local or unroutable traffic can be bridged among the bridged interfaces in the same bridge group, and routable traffic can be routed to other routed interfaces or bridge groups.

History for the IRB Support for the Cisco MGX-RPM-XF-512 Feature

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IRB Support for the Cisco MGX-RPM-XF-512

- Layer 2 bridging should be handled by a Route Processor (RP).
- Spanning Tree Protocol (STP) should be configured and operating correctly.

Restrictions for IRB Support for the Cisco MGX-RPM-XF-512

- IEEE STP is the only supported algorithm. DEC and IBM STP algorithms are not supported.
- IP is the only supported Layer 3 routing protocol. IPX and AppleTalk are not supported.
- IRB is not supported on Cisco AGS+ and Cisco 7000 series platforms.
- X.25 and ISDN bridged interfaces are not supported.
- IRB and concurrent routing and bridging cannot operate at the same time.
- IRB is not supported on source-route bridging (SRB).
- Hot Standby Router Protocol (HSRP) is not supported.

Information About IRB Support for the Cisco MGX-RPM-XF-512

To configure the IRB Support for the Cisco MGX-RPM-XF-512 feature, you should understand the following concepts:

- Bridging, page 2
- Bridge-Group Virtual Interface, page 3
- Integrated Routing and Bridging, page 3
- VRRP, page 4
- Design of the IRB Support for the Cisco MGX-RPM-XF-512 Feature, page 4

Bridging

Each bridge has a table that contains source MAC addresses and ports all of the workstations on every interface. The bridge learns the addresses and ports each time it sees a frame from a device. When the bridge receives a frame, it checks the table to determine on which port the destination MAC address exists. The bridge either filters the traffic (if the source and destination are on the same interface) or sends the frame out to the appropriate interface.

The following are bridge functions:

- Flooding—A bridge learns only unicast source addresses. The bridge floods all interfaces if it receives an unknown unicast frame (one that has no address in the bridge table). The frame is sent out to all forwarding interfaces except for the source. A legacy bridge can flood two other frame types: broadcast and multicast.
- Filtering and filtering database—When an interface has a source and destination address or the destination address is a special one for processing done by the upper-layer applications, filtering of the packets occurs. When filtering a packet, the packet is dropped or sent to upper-layer applications.

  The filtering database supports inquiries that decide whether received frames from a given reception port and with given values of destination MAC-address parameter are to be forwarded through a transmission port.
- Forwarding—A bridge forwards a frame when the destination address is a known unicast address (it has an entry in the bridging table) and there are no rules that block sending the packet to the destination interface.
- Time stamping—When a bridge learns a source address, it time-stamps the entry and starts a timer. Every time the bridge sees a frame from that source, the bridge updates the time stamp. If the timer expires, the bridge removes the entry from the table. The timer is configurable and has a default value of 300 seconds. Its valid range is from 10 to 1000000 seconds.
- Spanning Tree Protocol (STP)—STP is a loop-prevention protocol. It allows bridges to communicate with each other to discover loops in the network. The protocol specifies an algorithm that bridges use to create a loop-free logical topology. For more information, refer to *Understanding and Configuring Spanning Tree Protocol on Catalyst Switches.*

Bridge-Group Virtual Interface

Bridging is performed in the data-link layer (Layer 2) and routing is performed in the network layer (Layer 3) and each has a different protocol configuration. Bridging was not supported using a single interface until the addition of the Bridge-Group Virtual Interface (BVI).

The BVI is a virtual interface that represents a bridge group and enables both bridging and routing for the group that it represents. The bridge group makes the decision of whether or not a protocol is to be routed or bridged. All traffic in and out of the network flow through the BVI. Traffic coming from a bridged segment that is to be routed to the IP uses the BVI as the destination address, and traffic routed to the BVI from the IP is forwarded to the bridge group.

The BVI has its own MAC address that it borrows from the MAC address of one of its bridged interfaces to guarantee a globally unique address. The MAC address of the BVI is initialized when the BVI is created and dynamically updated when the membership of the bridge group changes. For redundancy, the BVIs are configured for Virtual Router Redundancy Protocol (VRRP) to prevent a single point of failure.

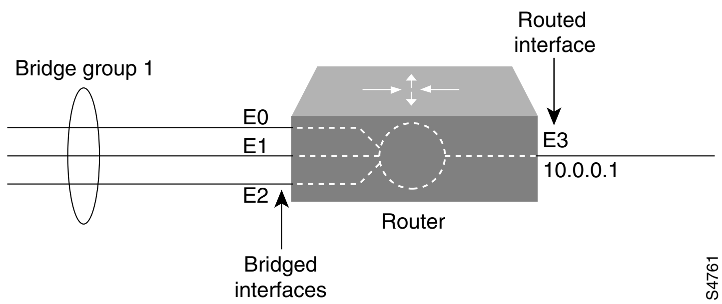
Integrated Routing and Bridging

While concurrent routing and bridging makes it possible to both route and bridge a specific protocol on separate interfaces within a router, the protocol is not switched between bridged and routed interfaces. Routed traffic is confined to the routed interfaces; bridged traffic is confined to bridged interfaces. A specified protocol may be either routed or bridged on a given interface, but not both.

Integrated routing and bridging (IRB) makes it possible to route a specific protocol between routed interfaces and bridge groups, or route a specific protocol between bridge groups. Local or unroutable traffic can be bridged among the bridged interfaces in the same bridge group, while routable traffic can be routed to other routed interfaces or bridge groups.

Figure 1 illustrates how integrated routing and bridging in a router interconnects a bridged network with a routed network.

Figure 1 Figure 8 Integrated Routing and Bridging Interconnecting a Bridged Network with a Routed Network



A specific protocol can be configured to route between routed interfaces and bridge groups or to route a between bridge groups. Specifically, local or unroutable traffic is bridged among the bridged interfaces in the same bridge group, while routable traffic is routed to other routed interfaces or bridge groups. Using integrated routing and bridging, you can do the following:

- Switch packets from a bridged interface to a routed interface
- Switch packets from a routed interface to a bridged interface
- Switch packets within the same bridge group

VRRP

The Virtual Router Redundancy Protocol (VRRP) is designed to eliminate the single point of failure inherent in the static default routed environment by automatically providing alternate router paths.

VRRP dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router controlling the IP address associated with a virtual router is called a master and forwards packets sent to these IP addresses. VRRP provides dynamic failover in the forwarding responsibility should the master become unavailable.

The virtual router associated with a given alternate path supported by VRRP uses the same IP address and MAC address as the routers for other paths. As a result, the host gateway information does not change, no matter what path is used. Backup of IP addresses is the primary function of VRRP.

Design of the IRB Support for the Cisco MGX-RPM-XF-512 Feature

The IRB Support for the Cisco MGX-RPM-XF-512 feature adds both bridging and routing functionality to the Cisco MGX-RPM-XF-512 that enables routing between interfaces and bridge groups or between bridge groups.

The Cisco RPM-XF-512 contains two customized application-specific integrated circuits (ASICs), called a Parallel Express Forwarding (PXF) microprocessor that punts the bridged packets to an route processor (RP). Packets are received from an IRB interface on its ATM PVCs. The encapsulation type and the packet header are checked. The IRB interface decides whether to route or bridge the packet.

If the IRB interface decides to bridge, the bridging is performed by an RP. Control packets on the BVI are punted to the RP. All other packets have the Layer 3 offset calculated and traverse the standard fast path.

How to Configure IRB Support for the Cisco MGX-RPM-XF-512

This section contains the following procedures:

- Enabling IRB, page 5 (required)
- Assigning a Bridge-Group Number to an Interface, page 6 (required)
- Assigning a Bridge-Group Protocol, page 7 (required)
- Configuring BVIs, page 8 (required)
- Configuring the Protocols for Routing or Bridging, page 10 (required)
- Verifying the IRB Configuration, page 11 (optional)
- Monitoring IRB and BVIs, page 15 (optional)

Enabling IRB

Perform this task to enable IRB.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge irb**
4. **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	**bridge irb** **Example:** `Router(config)# bridge irb`	Enables IRB for the routing of a protocol between routed interfaces and bridge groups or a protocol between bridge groups.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Assigning a Bridge-Group Number to an Interface

Perform this task to assign a bridge-group number to your interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bridge-group** *bridge-group*
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	**interface** *type number* **Example:** `Router(config)# interface ethernet1/0`	Enters interface configuration mode. The arguments are as follows: • *type*—Type of interface to be configured. Refer to the *Cisco IOS Interface and Hardware Component Command Reference*, Release 12.3T for more information on the available types. • *number*—Port, connector, or interface card number.
Step 4	**bridge-group** *bridge-group* **Example:** `Router(config-if)# bridge-group 15`	Assigns a bridge-group number to an interface. The *bridge-group* argument is a number from 1 to 255. There is no default.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.

Assigning a Bridge-Group Protocol

Perform this task to assign a bridge-group protocol.

Prerequisites

You should assign a bridge-group number and define STP before configuring the router for IRB.

Restrictions

IEEE protocol is supported only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **protocol ieee**
4. **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	**bridge** *bridge-group* **protocol ieee** **Example:** `Router(config)# bridge 10 protocol ieee`	Enables bridging that uses Spanning Tree Protocol (STP) in a bridge group. The argument and keyword are as follows: • *bridge-group*—Bridge-group number of a set of interfaces. The range is from 1 to 255. Frames are bridged only among interfaces in the same group. **Note** You will use this bridge-group number in subsequent **bridge** commands. • **ieee**—IEEE Ethernet STP. **Note** There are other available protocols (DEC, IBM, and VLAN-bridge), however IEEE Ethernet STP is the only protocol supported.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring BVIs

Perform this task to configure a Bridge-Group Virtual Interface (BVI). BVI is configured on a router and has all of the network layer attributes, such as a network address and the ability to perform filtering. The BVI performs like a normal interface but represents the entire corresponding bridge group to the interfaces within the router. The BVI is assigned the number of the bridge group that it represents.

Prerequisites

Although all bridged segments belonging to a bridge group can be represented as a single segment or network to a routing protocol, there are situations in which several individual networks coexist within the same bridged segment.

To make it possible for the routed domain to learn about the other networks behind the BVI, configure a secondary address on the BVI to add the corresponding network to the routing process.

Restrictions

Only one BVI is supported for each bridge group.

When you bridge and route a given protocol in the same bridge group, you must configure the network-layer attributes of the protocol on the BVI. Do not configure protocol attributes on the bridged interfaces. No bridging attributes can be configured on the BVI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **interface bvi** *bridge-group*
5. **ip address** *ip-address mask* [**secondary**]
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-if)# interface ethernet0/1`	Assigns and interface and enters interface configuration mode.
Step 4	**interface bvi** *bridge-group* **Example:** `Router(config-if)# interface bvi 245`	Specifies a BVI number for an interface. The *bridge-group* argument is from 1 to 255. There is no default.

	Command or Action	Purpose

Step 5	**ip address** *ip-address mask* [**secondary**]
	
	Example:
	`Router(config-if)# ip address 10.1.2.3 255.0.0.0`
	
	and
	
	`Router(config-if)# ip address 10.2.3.4 255.0.0.0 secondary`
		Specifies an IP address for an interface.
		
		The optional **secondary** keyword allows an unlimited number of secondary addresses to be assigned. Secondary addresses are treated like primary addresses, except the system never generates datagrams other than routing updates with secondary source addresses. IP broadcasts and Address Resolution Protocol (ARP) requests are handled properly, as are interface routes in the IP routing table.
		
		Note If any router on a network segment uses a secondary address, all other devices on that same segment must also use a secondary address from the same network or subnet. Inconsistent use of secondary addresses on a network segment can very quickly cause routing loops.

Step 6	**end** **Example:** `Router(config-if)# end`	Ends the configuration.

Configuring the Protocols for Routing or Bridging

Perform this task to configure protocols for routing or bridging.

When IRB is enabled, the default behavior in a bridge group is to bridge all packets. The bridge group can be explicitly configured to route a particular protocol, so that routed packets of the protocol are routed, while nonroutable packets of the protocol or packets for protocols for which the bridge group is not explicitly configured to route are bridged.

The bridge group can also be explicitly configured so that it does not bridge a particular protocol, and routable packets of the protocol are routed when the bridge is explicitly configured to route the protocol, and nonroutable packets are dropped because bridging is disabled for the protocol.

When you enable routing for a given protocol on the BVI, packets coming from a routed interface but destined for a host in a bridged domain are routed to the BVI. The BVI forwards the packets as bridged traffic to the corresponding bridged interface. All routable traffic received on a bridged interface is routed to other routed interfaces as if it is coming directly from the BVI.

Restrictions

IEEE protocol is supported only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **route** *protocol*
4. **bridge** *bridge-group* **bridge** *protocol*
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	**bridge** *bridge-group* **route** *protocol* **Example:** `Router(config)# bridge 10 route ieee`	Enables the routing of a specified protocol in a specified bridge group. The arguments and keywords are as follows: • *bridge-group*—Bridge-group number specified using the **bridge bridge** command. • *protocol*—IEEE and IP are the only supported protocols. **Note** You can use the **bridge bridge** command to configure the bridging of a specified protocol also.
Step 4	**bridge** *bridge-group* **bridge** *protocol* **Example:** `Router(config)# bridge 10 bridge ieee`	Enables a specified protocol that is to be bridged in a specified bridge group. The arguments and keywords are as follows: • *bridge-group*—Bridge-group number. • *protocol*—IEEE and IP are the only support protocols. **Note** You can use the **bridge route** command to configure the bridging of a specified protocol also.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits to global configuration mode.

Verifying the IRB Configuration

To verify the IRB configuration, perform the following steps.

SUMMARY STEPS

1. **show interfaces bvi** *bridge-group*
2. **show interfaces** *type number*
3. **show smf** *interface-name*
4. **show bridge verbose**

DETAILED STEPS

Step 1 **show interfaces bvi** *bridge-group*

Use the **show interfaces bvi** command to display the number of input and output packets on the interface and the MAC addresses of the BVI, for example:

```
Router# show interfaces bvi 100

BVI100 is up, line protocol is up
 Hardware is BVI, address is 0096.8e80.a510 (bia 004c.2613.1c10)
 Internet address is 10.1.1.1/24
 MTU 4470 bytes, BW 100000 Kbit, DLY 5000 usec, reliability 255/255, txload 1/255, rxload
   1/255
 Encapsulation ARPA, loopback not set
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input never, output never, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: fifo
 Output queue: 0/0 (size/max)
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
     268916 packets input, 17459591 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     4 packets output, 240 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Step 2 **show interfaces** *type number*

Use the **show interfaces** command to display a bridged interface and its protocol that can route to another routed interface (if the packet is routable), and protocols that the bridged interface bridges, for example:

```
Router# show interfaces FastEthernet1/1

Routed protocols on FastEthernet1/1:
 ip
Bridged protocols on FastEthernet1/1:
 clns       decnet     ip

Software MAC address filter on FastEthernet1/1
 Hash Len   Address           Matches    Act      Type
 0x00: 0   ffff.ffff.ffff     228978      RCV      Physical broadcast
 0x2A: 0   0900.2b01.0001     0           RCV      DEC spanning tree
 0x9E: 0   0096.8e80.a510     0           RCV      Interface MAC address
 0x9E: 1   0096.8e80.a510     0           RCV      Bridge-group Virtual Interface
 0xC0: 0   0100.0ccc.cccc     79428       RCV      CDP
 0xC2: 0   0180.c200.0000     677678      RCV      IEEE spanning tree
 0xC2: 1   0180.c200.0000     0           RCV      IBM spanning tree
 0xC2: 2   0100.0ccd.cdce     0           RCV      VLAN Bridge STP

Switch1
Not bridging this sub-interface.

Switch1.100
Routed protocols on Switch1.100:
 ip
Bridged protocols on Switch1.100:
 clns       decnet     ip
```

```
 Software MAC address filter on Switch1.100
 Hash Len    Address      Matches Act      Type
 0x00: 0 ffff.ffff.ffff         1 RCV Physical broadcast
 0x26: 0 004c.2613.1c00         0 RCV Interface MAC address
 0x2A: 0 0900.2b01.0001         0 RCV DEC spanning tree
 0x36: 0 004c.2613.1c10         0 RCV Interface MAC address
 0x9E: 0 0096.8e80.a510         0 RCV Bridge-group Virtual Interface
 0xC2: 0 0180.c200.0000         0 RCV IEEE spanning tree
 0xC2: 1 0180.c200.0000         0 RCV IBM spanning tree
 0xC2: 2 0100.0ccd.cdce         0 RCV VLAN Bridge STP

Switch1.200
Bridged protocols on Switch1.200:
 clns       decnet     ip

 Software MAC address filter on Switch1.200
 Hash Len    Address      Matches Act      Type
 0x00: 0 ffff.ffff.ffff         0 RCV Physical broadcast
 0x26: 0 004c.2613.1c00         0 RCV Interface MAC address
 0x2A: 0 0900.2b01.0001         0 RCV DEC spanning tree
 0x36: 0 004c.2613.1c10         0 RCV Interface MAC address
 0x36: 1 004c.2613.1c10         0 RCV Bridge-group Virtual Interface
 0xC2: 0 0180.c200.0000         0 RCV IEEE spanning tree
 0xC2: 1 0180.c200.0000         0 RCV IBM spanning tree
 0xC2: 2 0100.0ccd.cdce         0 RCV VLAN Bridge STP

Virtual-Access1

BVI100
 Routed protocols on BVI100:
 ip

BVI200
Routed protocols on BVI200:
 ip
```

Step 3 **show smf** *interface-name*

Use the **show smf** command to display the configured SMF on the interfaces, for example:

```
Router# show smf fastethernet1/0

Software MAC address filter on Switch1.100
 Hash Len    Address      Matches Act      Type
 0x00: 0 ffff.ffff.ffff         1 RCV Physical broadcast
 0x26: 0 004c.2613.1c00         0 RCV Interface MAC address
 0x2A: 0 0900.2b01.0001         0 RCV DEC spanning tree
 0x36: 0 004c.2613.1c10         0 RCV Interface MAC address
 0x9E: 0 0096.8e80.a510         0 RCV Bridge-group Virtual Interface
 0xC2: 0 0180.c200.0000         0 RCV IEEE spanning tree
 0xC2: 1 0180.c200.0000         0 RCV IBM spanning tree
 0xC2: 2 0100.0ccd.cdce         0 RCV VLAN Bridge STP
```

Step 4 **show bridge verbose**

Use the **show bridge verbose** command to display the extended bridge table including per bridge interface (flood port) receive (Rx) and transmit (Tx) counts, for example:

```
Router# show bridge verbose

Total of 300 station blocks, 261 free
Codes: P - permanent, S - self
```

```
BG Hash      Address      Action Interface      VC    Age   RX count   TX count

100 03/0   00c0.4300.2625 forward FastEthernet1/1 -      1          9          0
100 04/0   0050.538d.0400 forward FastEthernet1/1 -      4          1          0
100 09/0   00c0.4300.6b62 forward FastEthernet1/1 -      1          9          0
100 0E/0   00c0.4300.2628 forward FastEthernet1/1 -      1          9          0
100 15/0   0800.2077.7065 forward FastEthernet1/1 -      2          1          0
100 21/0   00c0.4300.3d1c forward FastEthernet1/1 -      1          9          0
100 27/0   00c0.4300.694e forward FastEthernet1/1 -      1          9          0
100 36/0   00c0.4300.93a5 forward FastEthernet1/1 -      1          9          0
100 3C/0   0005.5fbc.3408 forward FastEthernet1/1 -      1          1          0

BG Hash      Address      Action Interface      VC    Age   RX count   TX count

100 49/0   0000.8106.fbb2 forward FastEthernet1/1 -      0     160358          0
100 55/0   0800.20d1.3065 forward FastEthernet1/1 -      0          4          0
100 56/0   0003.3219.acfa forward FastEthernet1/1 -      0          1          0
100 57/0   00c0.4300.9ec9 forward FastEthernet1/1 -      1          9          0
100 58/0   0800.20a8.471f forward FastEthernet1/1 -      2          1          0
100 58/1   00c0.4300.6931 forward FastEthernet1/1 -      1          9          0
100 58/2   0004.5a44.dc84 forward FastEthernet1/1 -      3         12          0
100 74/0   0001.6443.6c18 forward FastEthernet1/1 -      0          1          0
100 7D/0   00c0.4300.dfa2 forward FastEthernet1/1 -      1          9          0
100 88/0   00c0.4301.058d forward FastEthernet1/1 -      1          9          0
100 8D/0   0001.96a4.dd50 forward FastEthernet1/1 -      4          1          0

BG Hash      Address      Action Interface      VC    Age   RX count   TX count

100 9A/0   0001.4226.5bc1 forward FastEthernet1/1 -      3          1          0
100 9D/0   00c0.4300.a13c forward FastEthernet1/1 -      1          9          0
100 9F/0   0007.856e.148b forward FastEthernet1/1 -      0          1          0
100 A9/0   0000.0c07.ac05 forward FastEthernet1/1 -      0     247809          0
100 AA/0   0001.6443.12b8 forward FastEthernet1/1 -      1          8          0
100 AC/0   0000.0c07.ac00 forward FastEthernet1/1 -      0     235943          0
100 AE/0   0001.6443.1fb1 forward FastEthernet1/1 -      1          9          0
100 B3/0   00c0.4300.e655 forward FastEthernet1/1 -      1          9          0
100 B3/1   0005.00d8.4ffc forward FastEthernet1/1 -      0     261036          0
100 BB/0   0004.defe.47fc forward FastEthernet1/1 -      0         22          0
100 BD/0   0030.71f8.219c forward FastEthernet1/1 -      1          9          0

BG Hash      Address      Action Interface      VC    Age   RX count   TX count

100 BF/0   0001.4226.59e6 forward FastEthernet1/1 -      0          1          0
100 DC/0   0030.71f8.09d5 forward FastEthernet1/1 -      1          9          0
100 E0/0   00e0.8f09.e000 forward FastEthernet1/1 -      0          1          0
100 E1/0   00e0.8f09.e001 forward FastEthernet1/1 -      1         20          0
100 E7/0   0004.6edb.57b0 forward FastEthernet1/1 -      0         53          0
100 EA/0   0001.0263.4ba1 forward FastEthernet1/1 -      0         10          0
100 EE/0   0030.71f8.05eb forward FastEthernet1/1 -      1          9          0
100 F8/0   00c0.4300.07ff forward FastEthernet1/1 -      1          9          0

Flood ports (BG 100)          RX count    TX count
FastEthernet1/1                 920790           0
Switch1.100                          0      920790

BG Hash      Address      Action Interface      VC    Age   RX count   TX count

Flood ports (BG 200)          RX count    TX count
Switch1.200                          0           0
```

Monitoring IRB and BVIs

To monitor IRB and BVIs, perform the following steps.

SUMMARY STEPS

1. **show pxf cpu tbridge**
2. **show pxf cpu statistics diversion**
3. **show pxf cpu subblock**

DETAILED STEPS

Step 1 **show pxf cpu tbridge**

Use the **show pxf cpu tbridge** command to display the SMF table for the BVI, for example:

```
Router# show pxf cpu tbridge

============Bridge-group Virtual Interface SMF table ====================

SMF Entry    Mac Address     SMF MATCH     BVI Flags
     1      0000.0000.0000           0      0x0
     2      0000.0000.0000           0      0x0
     3      0000.0000.0000           0      0x0
     4      0000.0000.0000           0      0x0
     5      0000.0000.0000           0      0x0
     6      0000.0000.0000           0      0x0
     7      0000.0000.0000           0      0x0
     8      0000.0000.0000           0      0x0
     9      0000.0000.0000           0      0x0

!Entry for BVI 10

    10      0000.0c09.6504           0      0x1
    11      0000.0000.0000           0      0x0
    12      0000.0000.0000           0      0x0
    13      0000.0000.0000           0      0x0
    14      0000.0000.0000           0      0x0
    15      0000.0000.0000           0      0x0
    16      0000.0000.0000           0      0x0
    17      0000.0000.0000           0      0x0
    18      0000.0000.0000           0      0x0
    19      0000.0000.0000           0      0x0
    20      0000.0000.0000           0      0x0
.
.
.
```

Step 2 **show pxf cpu statistics diversion**

Use the **show pxf cpu statistics diversion** command to display the packets that are diverted to the RP, for example:

```
Router# show pxf cpu statistics diversion

Diversion Cause Stats:
 local      = 31
 dest       = 0
 option     = 0
```

```
 protocol   = 0
 encap      = 0
 oam f5     = 149
 oam f4     = 0
 atm ilmi   = 0
 camp       = 0
 puissantly = 0
 pickiest   = 0
 pudgiest   = 0
 flippant   = 0
 mtu        = 0
 arp        = 1
 rarp       = 0
 icmp       = 0
 divert     = 0
 no_group   = 0
 direct     = 0
 local_mem  = 0
 p2p_prune  = 0
 assert     = 0
 dat_prune  = 0
 join_spt   = 0
 null_out   = 0
 igmp       = 0
 register   = 0
 no_fast    = 0
 ipc_resp   = 0
 keepalive  = 0
 min_mtu    = 0
 icmp_frag  = 0
 icmp_bad   = 0
 mpls_ttl   = 0
 tfib       = 0
 multicast  = 0
 clns_isis  = 0
 ppp_cntrl  = 0
 tun_norte  = 0
 tun_nofrg  = 0
 ctcp_in    = 0
 vsi_sig    = 8
 mvpn_tfrg  = 0
 cdp        = 0

!New output.

 smf_msmtch = 0
 irb_stp    = 0
 brdg_ip    = 0
 no_rt_ip   = 0
 multi_mac  = 0
```

Step 3 **show pxf cpu subblock**

Use the **show pxf cpu subblock** command to display the encapsulation type of a bridged subinterface.

```
Router# show pxf cpu subblock sw1.100

Switch1.100 is up
 ICB = C001, LinkId = 3, interface PXF, enabled
 IOS encapsulation type 33 ATM

!BVI encapsulation.

 ICB: Index: 49155 Min mtu: 4 Max mtu: 4486 Encapsulation Type:8
```

```
 VCCI maptable location = 0x8340A800
 VCCImap entry: vcci: 0x5   u0 : 0x64   Max mtu : 4486
             Min mtu : 0x4   vc_type_flags: 0x20
 VCCI 0x5        seg channel id 0x1A5
    icmp ipaddress 4.4.4.1          timestamp 0
    feature_data: flags 0x0000 fib_index 0x0
 col_5_cicb.flags : 0x00
```

Configuration Examples for IRB Support for the Cisco MGX-RPM-XF-512

This section contains the following configuration examples:

Assigning a Bridge-Group Number to an Interface: Example

The following example shows how to assign a bridge-group number to an interface:

```
interface Ethernet1
 no ip address
 bridge-group 2
```

Assigning a Bridge-Group Protocol and Routing Protocol: Example

The following example shows how to assign a bridge-group protocol:

```
bridge irb
bridge 1 protocol ieee
 bridge 1 route ip
 bridge 1 route ipx
bridge 2 protocol ieee
 bridge 2 route ipx
 bridge 2 route ip
```

Enabling IRB Configuration: Example

The following example shows how to enable IRB:

```
bridge irb
bridge 1 protocol ieee
 bridge 1 route ip
 bridge 1 route ipx
```

```
bridge 2 protocol ieee
 bridge 2 route ipx
 bridge 2 route ip
```

Configuring BVI on an Interface

The following example shows how to configure BVI on an interface and a secondary IP address:

```
interface BVI1
 ip address 10.0.0.1 255.0.0.0 secondary
 ip address 10.0.0.1 255.0.0.0
 ipx network 15
```

Running IRB Configuration: Example

The following example shows both bridging and routing on bridge group 1:

```
Router# show running-config
!
interface Ethernet2
 ip address 10.0.0.1 255.0.0.0
 bridge-group 1
!
interface Ethernet3
 ip address 10.0.0.1 255.0.0.0
 bridge-group 1
!
interface BVI1
 no ip address
!
router igrp 123
 network 10.0.0.0
 network 10.0.0.0
 network 10.0.0.0
!
bridge irb
bridge 1 protocol ieee
 bridge 1 route ipx
 bridge 1 route ip
 no bridge 1 bridge ip
```

Additional References

The following sections provide references related to the IRB Support for the Cisco MGX-RPM-XF-512 feature.

Related Documents

Related Topic	Document Title
IP addressing and services configuration tasks	*Cisco IOS IP Configuration Guide*, Release 12.3
IP addressing and services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services*, Release 12.3T

Related Topic	Document Title
Cisco PXM45-based MGX8850 and MGX8950 switch configuration tasks	*MGX 8850 (PXM1E/PXM45), MGX 8950, and MGX 8830 Software Configuration Guide*, Release 4
Interface and hardware component commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Interface and Hardware Component Command Reference*, Release 12.3T
Spanning Tree Protocol overview and configuration tasks	*Understanding and Configuring Spanning Tree Protocol on Catalyst Switches*

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
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
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new and modified commands only.

New Commands

- **debug pxf tbridge**
- **show pxf cpu tbridge**

Modified Commands

- **show pxf cpu statistics**
- **show pxf cpu subblock**

debug pxf tbridge

To enable debugging of parallel express forwarding (PXF) transparent bridging, use the **debug pxf tbridge** command in privileged EXEC mode. To disable the debugging, use the **no** form of this command.

debug pxf tbridge

no debug pxf tbridge

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.

Examples

The following example shows that BVI100 has been removed from the Software Mac-address Filter (SMF) table:

```
Router# debug pxf tbridge

*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Deleting BVI entry 100 from
SMF table.
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: BVI 100 ICM programming
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Successfully removed SMF
entry for bvi 100
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Deleting BVI entry 100 from
SMF table.
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: BVI 100 ICM programming
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Successfully removed SMF
entry for bvi 100
*Feb  8 18:39:05.178: %SYS-5-CONFIG_I: Configured from console by vty0
(CROI_MASTER_000A004B)
*Feb  8 18:39:06.710: %LINK-5-CHANGED: Interface BVI100, changed state to administratively
down
*Feb  8 18:39:07.710:%LINEPROTO-5-UPDOWN: Line protocol on Interface BVI100, changed state
to down
```

The following example shows that BVI is configured and that the SMF entry has been updated:

```
Router# debug pxf tbridge

*Feb  8 18:39:16.398:
Note: A random mac address of 0000.0ceb.c0f8 has been chosen for BVI in bridge group  100
since there is no mac address associated with the selected interface.
*Feb  8 18:39:16.398: Ensure that this address is unique.
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: SMF update for Switch1.1: BVI 100 Mac
Address 0000.0ceb.c0f8
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: BVI 100 ICM programming
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: Successfully updated SMF entry for bvi 100
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: SMF update for Switch1.1:
BVI 100 Mac Address 0000.0ceb.c0f8
```

```
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: BVI 100 ICM programming
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: Successfully updated SMF entry for bvi 100
*Feb  8 18:39:16.886: %SYS-5-CONFIG_I: Configured from console by vty0
(CROI_MASTER_000A004B)
*Feb  8 18:39:18.394: %LINK-3-UPDOWN: Interface BVI100, changed state to up
*Feb  8 18:39:19.394: %LINEPROTO-5-UPDOWN: Line protocol on Interface BVI100, changed
state to up
```

Related Commands

Command	Description
show pxf cpu statistics	Displays PXF CPU statistics for a configured router.
show pxf cpu subblock	Displays PXF CPU subblocks for a bridged subinterface.
show pxf cpu tbridge	Displays PXF CPU statistics in a CPU for transparent bridging.

show pxf cpu statistics

To display parallel express forwarding (PXF) central processing unit (CPU) statistics for a configured router, use the **show pxf cpu statistics** command in privilege EXEC mode.

show pxf cpu statistics [**crtp** | **diversion** | **drop** | **ip** | **mlp** | **qos** | **spd**]

Syntax DescriptionA

crtp	(Optional) IP header compression statistics.
diversion	(Optional) Packets that need to be bridged, as well as control packets such as Spanning Tree Protocol (STP) and Virtual Router Redundancy Protocol (VRRP), that are not processed by PXF and are diverted to a route processor (RP).
drop	(Optional) Packets that are dropped by the PXF.
ip	(Optional) IP statistics.
mlp	(Optional) Multilink PPP (MLP) statistics.
qos	(Optional) Quality of Service (QoS) statistics.
spd	(Optional) Multicast Selective Packet Discard (SPD) statistics.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2	This command was introduced.
12.3(14)T	This command was enhanced to include counters for Integrated Routing and Bridging (IRB) functionality.

Examples

The following is sample output from the **show pxf cpu statistics** command for diversion statistics:

```
Router# show pxf cpu statistics diversion

Diversion Cause Stats:
 local     = 31
 dest      = 0
 option    = 0
 protocol  = 0
 encap     = 0
 oam f5    = 149
 oam f4    = 0
 atm ilmi  = 0
 comp      = 0
 ip_sanity = 0
 ip_bcast  = 0
 ip_dest   = 0
 fib_punt  = 0
 mtu       = 0
 arp       = 1
 rarp      = 0
 icmp      = 0
 divert    = 0
```

```
 no_group  = 0
 direct    = 0
 local_mem = 0
 p2p_prune = 0
 assert    = 0
 dat_prune = 0
 join_spt  = 0
 null_out  = 0
 igmp      = 0
 register  = 0
 no_fast   = 0
 ipc_resp  = 0
 keepalive = 0
 min_mtu   = 0
 icmp_frag = 0
 icmp_bad  = 0
 mpls_ttl  = 0
 tfib      = 0
 multicast = 0
 clns_isis = 0
 ppp_cntrl = 0
 tun_norte = 0
 tun_nofrg = 0
 ctcp_in   = 0
 vsi_sig   = 8
 mvpn_tfrg = 0
 cdp       = 0

!IRB counters

 smf_msmtch= 0
 irb_stp   = 0
 brdg_ip   = 0
 no_rt_ip  = 0
 multi_mac = 0
```

Related Commands

Command	Description
debug pxf tbridge	Displays debugging output of the PXF transparent bridging.
show pxf cpu subblock	Displays PXF CPU for a bridged subinterface.
show pxf cpu tbridge	Displays PXF CPU statistics for transparent bridging.

show pxf cpu subblock

To display parallel express forwarding (PXF) central processing unit (CPU) statistics for a bridged subinterface (encapsulation type), use the **show pxf cpu subblock** command in privileged EXEC mode.

show pxf cpu subblock *interface-name*

Syntax Description

interface-name	Name of the interface.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2	This command was introduced.
12.3(14)T	This command was enhanced to display more information for all subblocks.

Examples

The following is sample output from the **show pxf cpu subblock** command, which shows the bridge-group virtual interface software MAC-address filtering (SMF) table:

```
Router# show pxf cpu subblock switch1.100

Switch1.100 is up

 ICB = C001, LinkId = 3, interface PXF, enabled
 IOS encapsulation type 33 ATM

!BVI encapsulation denoted by the type.

 ICB: Index: 49155 Min mtu: 4 Max mtu: 4486 Encapsulation Type:8
 VCCI maptable location = 0x8340A800
 VCCImap entry: vcci: 0x5   u0 : 0x64   Max mtu : 4486
             Min mtu : 0x4   vc_type_flags: 0x20
 VCCI 0x5         seg channel id 0x1A5
    icmp ipaddress 4.4.4.1          timestamp 0
    feature_data: flags 0x0000 fib_index 0x0
 col_5_cicb.flags : 0x00
```

Related Commands

Command	Description
debug pxf tbridge	Displays debugging output of the PXF transparent bridging.
show pxf cpu statistics	Displays PXF CPU statistics for a configured router.
show pxf cpu tbridge	Displays PXF CPU statistics for transparent bridging.

show pxf cpu tbridge

To display parallel express forwarding (PXF) central processing unit (CPU) statistics for transparent bridging, use the **show pxf cpu tbridge** command in privileged EXEC mode.

show pxf cpu tbridge

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.

Examples

The following is sample output from the **show pxf cpu tbridge** command, which shows the bridge-group virtual interface software MAC-address filtering (SMF) table:

```
Router# show pxf cpu tbridge

Bridge-group Virtual Interface SMF table ========================================

SMF Entry    Mac Address     SMF MATCH     BVI Flags
     1      0000.0000.0000           0      0x0
     2      0000.0000.0000           0      0x0
     3      0000.0000.0000           0      0x0
     4      0000.0000.0000           0      0x0
     5      0000.0000.0000           0      0x0
     6      0000.0000.0000           0      0x0
     7      0000.0000.0000           0      0x0
     8      0000.0000.0000           0      0x0
     9      0000.0000.0000           0      0x0

!Entry for BVI 10.

    10      0000.0c09.6504           0      0x1

!Bridged packets.

    11      0000.0000.0000           0      0x0001
    12      0000.0000.0000           0      0x0
    13      0000.0000.0000           0      0x0
    14      0000.0000.0000           0      0x0
    15      0000.0000.0000           0      0x0
    16      0000.0000.0000           0      0x0
    17      0000.0000.0000           0      0x0

!Routed packets.

    18      0000.0000.0000           0      0x0100
    19      0000.0000.0000           0      0x0
    20      0000.0000.0000           0      0x0
.
.
```

Related Commands

Command	Description
debug pxf tbridge	Displays debugging output of the PXF transparent bridging.
show pxf cpu statistics	Displays PXF CPU statistics for a configured router.
show pxf cpu subblock	Displays PXF CPU statistics for a bridged subinterface.

Glossary

ARP—Address Resolution Protocol.

BPDU—Bridge Protocol Data Unit (spanning tree frame types).

BVI—Bridge-Group Virtual Interface.

filtering—Does not replay packets received by a bridge port to other ports on that bridge, in order to prevent the duplication of packets.

IRB—Integrated Routing and Bridging.

SMF—Software MAC-address Filter.

STP—Spanning Tree Protocol.

VLAN—Virtual LAN.

VRRP—Virtual Router Redundancy Protocol (RFC 2338).

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.