Cisco ASR 9000 Series Aggregation Services Router Troubleshooting Feature Module

Revised: March 2009

This feature module describes techniques to troubleshoot the Cisco ASR 9000 Series Aggregation Services Router.

Contents

Release 3.7.2 Final Draft?Cisco Confidential

Access Control List (ACL)

ACLs are used for packet filtering and selecting traffic types to be analyzed, forwarded, or influenced in some way. Access Control Entries (ACEs) are individual permit or deny statement within an ACL. Each ACE includes an action element ("permit" or "deny") and a filter element based upon criteria such as source address, destination address, protocol, protocol-specific parameters, and so on. This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show access-lists ipv4** *access-list-name [hardware {ingress | egress} {sequence-number | location node-id} | summary [access-list-name] | access-list-name [sequence-number] | maximum [detail] [usage {pfilter location node-id}]*
2. **show pfilter-ea ha** *{info | chkpt ... {all | objid_num } location ...*
3. **show pfilter-ea fea** *{es-acl|ipv4-acl} acl_name location ...*
4. **show pfilter-ea fea** *summary location ...*
5. **show pfilter-ea fea** *acl-hash location ...*
6. **show pfilter-ea** *{es-acl|ipv-acl} trace {critical|intermittent} {all|error|event} location ...*
7. **show pfilter-ea fea** *trace location ...*

8. **debug pfilter-ea trace**
9. **debug pfilter-ea errors**
10. **debug pfilter-ea info**
11. **debug pfilter-ea all**
12. **debug feature_ea_dll** *{all | trace | error | info}*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show access-lists ipv4** *access-list-name [hardware {ingress | egress} {sequence number | location node-id} | summary [access-list-name] | access-list-name [sequence-number] | maximum [detail] [usage {pfilter location node-id}]* **Example:** `RP/0/RSP0/CPU0:router# show access-lists ipv4 dtho 10 ipv4 access-list dtho 10 permit ipv4 any any`	View all IPv4 ACL contents. Filter results using the following parameters and keywords: • **access-list-name**-—IPv4 ACL name. • **hardware**—Ingress specifies an inbound interface, egress specifies an outbound interface. • **sequence-number**—ACL number, 1 to 2147483646. • **location** *node-id—Rack/slot/module* notation of ACL. • **summary**—Summary of all current IPv4 ACLs. • **maximum**—Maximum configurable IPv4 ACLs and ACEs. • **detail**—Out-of-resource (OOR) details, OOR limits the number of ACLs and ACEs configured. • **usage**—View the usage of the ACL on a given LC. • **pfilter**—Packet filtering for the LC.
Step 2	**show pfilter-ea ha** *{info | chkpt ... {all | objid_num } location ...* **Example:** `RP/0/RSP0/CPU0:router# show pfilter-ea ha info location 0/1/CPU0`	View checkpoint record information and the pfilter-ea state.
Step 3	**show pfilter-ea fea** *{es-acl|ipv4-acl} acl_name location ...* **Example:** `RP/0/RSP0/CPU0:router# show pfilter-ea fea ipv4-acl dtho location 0/1/CPU0`	View TCAM/stats entries used by the ACL.
Step 4	**show pfilter-ea fea** *summary location ...* **Example:** `RP/0/RSP0/CPU0:router# show pfilter-ea fea summary location 0/1/CPU0`	View total TCAM/stats/hash entries used by pfilter-ea feature.
Step 5	**show pfilter-ea fea** *acl-hash location ...*	View ACL hash structure information.
Step 6	**show pfilter-ea** *{es-acl|ipv-acl} trace {critical|intermittent} {all|error|event} location ...*	View critical errors from the non-wrapping buffer. or information messages from a wrapping buffer.

	Command or Action	Purpose
Step 7	**show pfilter-ea fea** *trace location ...*	Logs critical error messages.
Step 8	**debug pfilter-ea** *trace*	View trace of function calls in pfilter-ea process.
Step 9	**debug pfilter-ea** *errors*	View errors observed in pfilter-ea process.
Step 10	**debug pfilter-ea** *info*	View debug messages about pfilter-ea process.
Step 11	**debug pfilter-ea** *all*	Turns on all debug messages.
Step 12	**debug feature_ea_dll** *{all | trace | error | info}*	View error messages at various levels.

ACL Messages Not Appearing

Step 1 View ACL's ACEs.

```
RP/0/RSP0/CPU0:router# show access-list ipv4
```

Step 2 View ACL's TCAM entries.

```
RP/0/RSP0/CPU0:router# show access-list ipv4 hardware {ingress | egress} detail ...
```

Step 3 View ACL's TCAM entries.

Step 4 Configure the logs in the ACL.

```
RP/0/RSP0/CPU0:router# ipv4 access-list log-update threshold
```

Step 5 Ensure that the **log** keyword is in the ACE and wait five minutes (the default time to return results).

```
RP/0/RSP0/CPU0:router# permit ipv4 any any log
```

Workaround

Step 1 If the **log** flag is not set in hardware display:

a. Remove the ACL.

b. Reapply the ACL.

Step 2 The packet that matches the ACE with the **log** keyword is changed in the hardware and sent to the LC. The packet bound for the LC is policed at rate 1000 pps. You cannot modify the rate.

Step 3 To get the log sooner, change the log-update threshold.

Fragmented Packets Being Accepted

Step 1 View ACL's ACEs.

```
RP/0/RSP0/CPU0:router# show access-list ipv4
```

Release 3.7.2 Final Draft?Cisco Confidential

Step 2 View IPv4 counters e.g., fragment etc...

```
RP/0/RSP0/CPU0:router# show ipv4 traffic
```

Step 3 View ACL's TCAM entries.

Step 4 `RP/0/RSP0/CPU0:router# show access-list ipv4 hardware {ingress/egress} location`

Step 5 Ensure that the **fragment** keyword is in the ACE.

```
RP/0/RSP0/CPU0:router# deny ipv4 any any fragments
```

Step 6 Check the fragment packet count received by the device.

Step 7 View TCAM entries.

Workaround

Fragmented packets are not matched against the deny ACE without the **fragment** keyword. If there is not an entry with the "fragment" flag do the following procedure.

Step 1 Remove the ACL from all interfaces.

Step 2 Add the explicit **fragment** keyword in the ACE to deny the fragment packet.

Step 3 Reapply the ACL to all interfaces.

Egress Counters Broken

Step 1 View known routes.

```
RP/0/RSP0/CPU0:router# show route ipv4
```

Step 2 View ARP table entries. Look for the next hop.

```
RP/0/RSP0/CPU0:router# show arp
```

Step 3 View TCAM entries for the ACL.

```
RP/0/RSP0/CPU0:router# show access-list ipv4 hardware
```

Step 4 View UIDB properties, like ACL and QOS. Look for ACL-enable and ACL-id.

```
RP/0/RSP0/CPU0:router# show uidb data location
```

Workaround

Step 1 If the route is missing or the ARP is incomplete, use the `no shut` command to recover.

Step 2 If the UIDB table or TCAM entry is incorrect, remove the ACL from all of the interfaces and reapply it.

ACL Interface Bind Rejected

Step 1 View errors encountered when the configuration was applied.

```
RP/0/RSP0/CPU0:router# show configuration failed
```

Step 2 View resources used by pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea summary
```

Step 3 View TCAM utilization for all features.

```
RP/0/RSP0/CPU0:router# show prm server tcam summary all all all
```

Workaround

If the error is related to TCAM space, remove ACEs from the ACL. There is a limit of 64 TCAM entries per ACL.

Single ACE Using Many TCAMs

Step 1 View ACL's ACEs.

```
RP/0/RSP0/CPU0:router# show access-list ipv4
```

Step 2 View resources used by pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea summary
```

Step 3 View TCAM utilization for all features.

```
RP/0/RSP0/CPU0:router# show prm server tcam summary all all all
```

Step 4 Check the number of ranges in the ACE.

ACL Using Varying TCAM Space

Step 1 View resources used by pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea summary
```

Step 2 View Pre-Internal Forwarding Information Base (Pre-IFIB) hardware statistic entries.

```
RP/0/RSP0/CPU0:router# show lpts pifib brief
```

Step 3 View TCAM utilization for all features, TCAM space is shared between the QoS/ACL and IFIB.

```
RP/0/RSP0/CPU0:router# show prm server tcam summary all all all
```

L3 Interface Ethernet Services Not Working

Ethernet services are not supported on L3 interfaces.

ACL Logs Not Working for Ethernet Services

Ethernet services logging is not supported.

Ethernet Services ACL Bind on Interface Rejected

Step 1 View any errors encountered when the configuration was applied.

```
RP/0/RSP0/CPU0:router# show configuration failed
```

Step 2 View ACL's ACEs.

```
RP/0/RSP0/CPU0:router# show access-list ipv4
```

Step 3 View trace log for pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea trace
```

Step 4 View the trace log for pfilter_ea resource manager dll.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea trace
```

Step 5 View resources used by pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea summary
```

Step 6 View TCAM utilization for all features.

```
RP/0/RSP0/CPU0:router# show prm server tcam summary all all all
```

Workaround

Step 1 If a field in the ACL is not supported, remove it from the ACE.

Step 2 If the TCAM is out of space, reduce the ACEs in the ACL.

Step 3 Reduce the ranges in the ACL.

Changing ACL Exhausts TCAM

Step 1 View ACEs configured for the AC.

```
RP/0/RSP0/CPU0:router# show access-list {ethernet-service/ipv4}
```

Step 2 View resources used by pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea summary
```

Step 3 View TCAM utilization for all features.

```
RP/0/RSP0/CPU0:router# show prm server tcam summary all all all
```

Workaround

Remove the old ACL before applying the new one.

ACLI on Bundle Interface Using TCAM Space in Network Processor (NP)

Step 1 View resources used by pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea summary
```

Step 2 View TCAM utilization for all features. If a bundle member belongs to that LC, the TCAMs of all the NPs on the same LC are programmed.

```
RP/0/RSP0/CPU0:router# show prm server tcam summary all all all
```

Cannot Delete ACL

Step 1 View any errors encountered when the configuration was applied.

```
RP/0/RSP0/CPU0:router# show configuration error
```

Step 2 View interfaces using the ACL.

```
RP/0/RSP0/CPU0:router# show access-list {ethernet-services/ipv4} usage pfilter
```

Step 3 View pfilter_ea information about the ACL.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea acl-hash
```

Step 4 View ACE information for ipv4.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea ipv4-acl
```

Step 5 View ACE information for Ethernet services.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea es-acl
```

Step 6 View ipv4 trace information.

```
RP/0/RSP0/CPU0:router# show access-list ipv4 trace
```

Step 7 View the Ethernet services trace.

```
RP/0/RSP0/CPU0:router# show access-list ethernet-services trace
```

Step 8 If the ACL is not attached to an interface, restart the process.

```
RP/0/RSP0/CPU0:router# process restart pfilter_ea
```

Bundle Member not Appearing

View resources used by pfilter_ea.

```
RP/0/RSP0/CPU0:router# show pfilter-ea fea summary
```

DF Bit Not Supported

The No Dont Fragment bit is not supported as match criteria in the current release.

Max ACL Limit Reached

The maximum number of ACL IDs per NP is 2048. Interfaces share TCAM entries for the acl_name and direction.

Unsupported Combinations in ACL

The current release supports the following combinations:

- VLAN OUT + L2 PROTO + MAC SA + MAC DA
- VLAN OUT + VLAN IN + MAC SA + MAC DA
- VLAN OUT + VLAN IN + L2 PROTO + MAC DA

No Statistics Counters

Statistics counters are not supported in the current release.

Release 3.7.2 Final Draft?Cisco Confidential

TCAMs Out of Resources

The "TCAMs Out of Resources" message means you have attempted to provision more than the available number of TCAM entries.

Bidirectional Forwarding Detection (BFD)

BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols.This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show bfd [ipv4 | all] [location** <*node*>**]**
2. **show bfd client [detail]**
3. **show bfd [ipv4 | all]** *session* **[detail][interface** <*ifname*> **[destination** <*address*>**]] [[***agent***] location** <*node*>**]**
4. **show bfd counters packet [interface** <*ifname*>**] location** <*node*>
5. **show bfd trace {adjacency | error | fsm | packet} [filter {destination <address> | interface** <*ifname*>**}] [location** <*node*>**]**
6. **show tech-support routing bfd** *{terminal [page] | file send-to* **[background] [compressed | uncompressed]}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show bfd** *[ipv4 | all] [location <node>]* **Example:** `RP/0/RSP0/CPU0:router# show bfd location 0/3/cpu0`	View general BFD information on the RSP like the number of sessions. Use the **location** keyword to display information for a specific LC. If not specified, information for all locations displays.
Step 2	**show bfd client** *[detail]* **Example:** `RP/0/RSP0/CPU0:router# show bfd client detail`	View BFD clients. Use the **detail** keyword to display more information.

	Command or Action	Purpose
Step 3	**show bfd** [ipv4 | all] session [detail][interface <ifname> [destination <address>]] [[agent] location <node>] **Example:** RP/0/RSP0/CPU0:router# show bfd session interface Gig2/1/0/0 detail	View BFD session information. Filter results using the following parameters and keywords: • **location**—BFD sessions hosted on this location. • **interface**—BFD sessions on the specified interface (no wild-card). • **destination** *address*—BFD sessions destined for that IP address. • **detail**—Detailed session information: statistics, number of state transitions. • **agent**—Cisco-support keyword used to obtain information directly from the LC bypassing the RSP.
Step 4	**show bfd counters packet** [interface <ifname>] location <node> **Example:** RP/0/RSP0/CPU0:router# show bfd counters packet interface POS 0/3/0/0 location 0/3/cpu0	View packet counters information. Filter results using the following parameters and keywords: • **location**—Packet counters for BFD sessions hosted on this location. • **interface**—Packet counters for BFD sessions on the specified interface (no wild-card).
Step 5	**show bfd trace** {adjacency | error | fsm | packet} [filter {destination <address> | interface <ifname>}] [location <node>] **Example:** RP/0/RSP0/CPU0:router# show bfd trace fsm location node-id	View tracing information from the RSP. Filter results using the following parameters and keywords: • **adjacency**—Traces generated when BFD receives an adjacency update from Adjacency Information Base (AIB) Finite State Machine (FSM) display. • **error**—Traces generated when an error is detected. • **fsm**—Traces generated when there is a state change in a session. • **packet**—Traces generated when there is a change in a tx or an rx packet. • **location**—Traces for BFD traces on the specified interface. **Note** To save the results, log the trace to a file.
Step 6	**show tech-support routing bfd** {terminal [page] | file send-to [background] [compressed | uncompressed]} **Example:** RP/0/RSP0/CPU0:router# show tech-support routing bfd	View BFD debugging information.

BFD Sessions in Down State

Step 1 Verify IP connectivity. Verify there is no IP packet loss.

```
RP/0/RSP0/CPU0:router# ping local-remote-address
```

Step 2 View routing protocol states. Verify the routing protocol is up.

```
RP/0/RSP0/CPU0:router# show router
```

Step 3 Ensure that the router and remote device are configured with the following:

a. Number of BFD sessions they can support

b. Timers to support the police rates

BFD Sessions Flap

Also see:

- BFD Sessions Down on Neighboring Router, page 13
- BFD Sessions Are Not Created on the LC, page 13

Step 1 Verify IP connectivity.

```
RP/0/RSP0/CPU0:router# ping local-IP-address
```

Step 2 View input and output counters.

```
RP/0/RSP0/CPU0:router# show interface
```

Step 3 View session detail information.

```
RP/0/RSP0/CPU0:router# show bfd session detail
```

Step 4 View session packet counters.

```
RP/0/RSP0/CPU0:router# show bfd counter
```

Step 5 View SPP counters.

```
RP/0/RSP0/CPU0:router# show spp node
```

Step 6 View UIDB BFD flag.

```
RP/0/RSP0/CPU0:router# show uidb data
```

Step 7 View BFD counters on NP.

```
RP/0/RSP0/CPU0:router# show controllers np counters
```

Step 8 View resource usage.

```
RP/0/RSP0/CPU0:router# monitor process
```

Step 9 View IP connectivity. Verify there is no IP packet loss.

```
RP/0/RSP0/CPU0:router# ping local remote address
```

Step 10 The BFD flap is a result of the application flap.

```
bfd_agent[104]: %BFD-6-SESSION_REMOVED : BFD session to neighbor v on interface Gi0/5/0/0
has been remove
```

Step 11 Verify the SPP is not losing packets.

```
RP/0/RSP0/CPU0:router# show spp node location
```

Step 12 Verify BFD is enabled in the UIDB.

```
RP/0/RSP0/CPU0:router# show uidb data location
```

Step 13 Check LC CPU and memory usage.

```
RP/0/RSP0/CPU0:router# monitor processes location
```

Step 14 Check the local interface counters.

```
RP/0/RSP0/CPU0:router# show interfaces type instance
```

Step 15 Check any qos policies applied to the interface.

```
RP/0/RSP0/CPU0:router# show policy-map interface
```

Step 16 Repeat steps on remote end.

BFD Sessions Flap Because of Local Echo Failure

BFD sessions flap may be locally triggered because the router detects echo failure.

Examine LC CUP utilization: **`monitor process location`**

Examine spp process on the LC CPU, this tells us about the delay encountered by BFD echo packets: **`show bfd trace performance reverse location`**

Rule out BFD echo packet loss: **`show bfd counters packet location`**

BFD Sessions Flap Because of SPP Process Restart

If BFD failure detection is configured to be within 1 second, the BFD session would flap if SPP process is restarted on the LC.

BFD Sessions Down on Neighboring Router

The neighbor router sends this message to indicate its BFD is going down:

```
LC/0/6/CPU0:Aug 8 16:42:56.821: bfd_agent[104]: %L2-BFD-6-SESSION_STATE_DOWN: BFD session
to neighbor 192.1.1.1 on interface Gi0/5/0/0 has gone down. Reason: Nbor signalled down
```

BFD Sessions Are Not Created on the LC

BFD sessions per LC is 1024. Configuring more than 1024 BFD sessions may result in random BFD sessions not being created.

Release 3.7.2 Final Draft?Cisco Confidential

Connectivity Fault Management (CFM)

CFM monitors, detects, and diagnoses remote network faults, end-to-end across the network. It does this using keepalives and MAC-based ping and traceroutes. This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **debug ethernet cfm platform**
2. **debug ethernet oam platform**
3. **debug BFD platform**
4. **show spp node**
5. **show spp client**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**debug ethernet cfm platform** **Example:** `RP/0/RSP0/CPU0:router# debug ethernet cfm platform`	View PD specific debugging information for CFM.
Step 2	**debug ethernet oam platform** **Example:** `RP/0/RSP0/CPU0:router# debug ethernet oam platform`	View PD specific debugging information for OAM.
Step 3	**debug BFD platform** **Example:** `RP/0/RSP0/CPU0:router# debug BFD platform`	View PD specific debugging information for BFD.

	Command or Action	Purpose
Step 4	**show spp node** **Example:** `RP/0/RSP0/CPU0:router# show spp node`	View SPP counters.
Step 5	**show spp client** **Example:** `RP/0/RSP0/CPU0:router# show spp client`	View for SPP drops.

No CCMs at MEP

Step 1 View configured MEPs and MIPs.

```
RP/0/RSP0/CPU0:router# show ethernet cfm local main
```

Step 2 View information about MEPs local node along with CCM statistics.

```
RP/0/RSP0/CPU0:router# show ethernet cfm location mep
RP/0/RSP0/CPU0:router# show ethernet cfm peer mep
```

Step 3 View remote MEPs shown by the specific LC CFM instance. If CCMs are not received, the peer will not display.

Step 4 Ensure that CFM is enabled for the interface in the NP. Verify the threshold matches the level given during CFM global domain config. Verify VLAN flags match the VLAN configuration for the interface.

```
RP/0/RSP0/CPU0:router# show uidb
```

Step 5 View CFM SID statistics seen by the SPP.

```
RP/0/RSP0/CPU0:router# show spp sid stats
```

Step 6 View CFM SID statistics seen by the NP.

```
RP/0/RSP0/CPU0:router# show control np stats
```

Step 7 View packets seen by the CFM PI. Enable all of the options. The output will show if packets are dropped, forwarded, or processed.

```
RP/0/RSP0/CPU0:router# debug ethernet cfm packet pack ccm
```

Step 8 View SPP drops.

```
RP/0/RSP0/CPU0:router# show spp client location 0/2/cpu0
```

Release 3.7.2 Final Draft?Cisco Confidential

High Level Packet to Low Level MEP Yields No Debug Messages

Step 1 View configured MEPs and MIPs.

```
RP/0/RSP0/CPU0:router# show ethernet cfm local main
```

Step 2 View information about MEPs local node along with CCM statistics.

```
RP/0/RSP0/CPU0:router# show ethernet cfm location mep
```

Step 3 View remote MEPs shown by the specific LC CFM instance. If CCMs are not received, the peer will not display.

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer mep
```

Step 4 Ensure that CFM is enabled for the interface in the NP. Verify the threshold matches the level given during CFM global domain config. Verify VLAN flags match the VLAN configuration for the interface.

```
RP/0/RSP0/CPU0:router# show uidb
```

Step 5 View CFM SID statistics seen by the SPP.

```
RP/0/RSP0/CPU0:router# show spp sid stats
```

Step 6 View CFM SID statistics seen by the NP.

```
RP/0/RSP0/CPU0:router# show control np stats
```

Step 7 View packets seen by the CFM PI. Enable all of the options. The output will show if packets are dropped, forwarded, or processed.

```
RP/0/RSP0/CPU0:router# debug ethernet cfm packet pack ccm
```

Step 8 View SPP drops.

```
RP/0/RSP0/CPU0:router# show spp client location 0/2/cpu0
```

Disabled CCM at MEP Affecting Remote/Peer MEPs

Step 1 View configured MEPs and MIPs.

```
RP/0/RSP0/CPU0:router# show ethernet cfm local main
```

Step 2 View information about MEPs local node along with CCM statistics.

```
RP/0/RSP0/CPU0:router# show ethernet cfm location mep
```

Step 3 View remote MEPs shown by the specific LC CFM instance. If CCMs are not received, the peer will not display.

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer mep
```

Step 4 Ensure that CFM is enabled for the interface in the NP. Verify the threshold matches the level given during CFM global domain config. Verify VLAN flags match the VLAN configuration for the interface.

```
RP/0/RSP0/CPU0:router# show uidb
```

Step 5 View CFM SID statistics seen by the SPP.

```
RP/0/RSP0/CPU0:router# show spp sid stats
```

Step 6 View CFM SID statistics seen by the NP.

```
RP/0/RSP0/CPU0:router# show control np stats
```

Step 7 View packets seen by the CFM PI. Enable all of the options. The output will show if packets are dropped, forwarded, or processed.

```
RP/0/RSP0/CPU0:router# debug ethernet cfm packet pack ccm
```

CFM ping and traceroute Result in “not found”

Step 1 View all CFM global configurations.

```
RP/0/RSP0/CPU0:router# show run ethernet cfm
```

Step 2 View local MEPs and their CCM statistics.

```
RP/0/RSP0/CPU0:router# show ethernet cfm location main
```

Step 3 View CFM CCM received from Peer MEPs.

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer meps
```

Step 4 View CCM database entries.

```
RP/0/RSP0/CPU0:router# show ethernet cfm location ccm
```

Step 5 Verify CFM connectivity CFM.

```
RP/0/RSP0/CPU0:router# ping ethernet cfm
```

Step 6 Start an CFM trace.

```
RP/0/RSP0/CPU0:router# trace ethernet cfm
```

Dropped CFM PDUs

Step 1 Statistics of CFM PDUs per interface.

```
RP/0/RSP0/CPU0:router# show ethernet cfm interfaces statistics
```

Release 3.7.2 Final Draft?Cisco Confidential

Step 2 View MST status.

```
RP/0/RSP0/CPU0:router# show spanning-tree mst mstp
```

Step 3 View peer MEPs seen by every local MEP.

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer meps
```

Step 4 Check STP status on the interfaces with MEPs or MIPs. CFM PDUs originating at MEPs on a STP block port get forwarded, however PDUs forwarded on a MIP are subject to STP port state. This means if MIP is on a port which is STP blocked, then CFM PDUs will be dropped at the MIP.

Step 5 View STP state and CFM peer MEP status.

```
RP/0/RSP0/CPU0:router# show spanning-tree mst mstp
```

Step 6 Verify forwarded packets are being dropped at the MIP by enabling packet debug.

```
RP/0/RSP0/CPU0:router# debug ethernet cfm pack rec dropped int gig
```

Viewing CFM Statistics in a Multi-level Multi-domain Case

View per domain statistics of CFM PDUs. Look for statistics related to a particular domain.

```
show ethernet cfm location mep domain
```

CFM ping Showing Sequence Errors

Step 1 View CFM connectivity.

```
RP/0/RSP0/CPU0:router# ping ethernet cfm
```

Step 2 Ensure that the fabric receive count at the NP.

```
RP/0/RSP0/CPU0:router# show control np register
```

Dynamic Host Configuration Protocol (DHCP) Snooping

DHCP snooping is a DHCP security feature that provides security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding table. An untrusted message is a message that is received from outside the network or firewall and that can cause traffic attacks within your network. This section describes the following:

Operational Commands

The DHCP application runs on the RSP. It has several operational EXEC mode CLI commands that present the application's configuration state, DHCP Client state and DHCP Packet statistics.

- **show dhcp ipv4 snoop binding**—View the state of DHCP Clients in a table.
- **show dhcp ipv4 snoop binding mac-address** *macaddress*—View detailed state of DHCP Clients with the specified MAC Address.
- **show dhcp ipv4 snoop binding summary**—View the total number of DHCP Clients.
- **show dhcp ipv4 snoop profile**—View a list of DHCP Snoop profiles.
- **show dhcp ipv4 snoop profile name** *name*—View details of a specific DHCP snoop profile.
- **show dhcp ipv4 snoop statistics**—View aggregate DHCP Snoop packet Rx, Tx, And Drops for each bridge domain.
- **show dhcp ipv4 snoop statistics bridge-domain** *name*—View detailed DHCP Snoop packet Rx, Tx, and Drops for each message type in a bridge domain.

Trace Commands

The DHCP application has over 1200 Trace logs. The Trace logs record significant events that occur in the application. Trace logs that are associated with a specific DHCP Client will contain the Client's MAC Address.

- **show dhcp ipv4 trace {errors | events | packets | snoop {errors | events | internals}}**
- **show dhcp ipv4 trace errors**—View error traces.
- **show dhcp ipv4 trace events**—View event traces.
- **show dhcp ipv4 trace packets**—View packet processing traces.
- **show dhcp ipv4 trace snoop errors**—View error traces for DHCP Snoop feature.
- **show dhcp ipv4 trace snoop events**—View event traces for the DHCP Snoop feature.
- **show dhcp ipv4 trace snoop internal**—View internal debug traces for the DHCP Snoop feature.

Syslog Commands

The DHCP application has over 1600 syslog logs. These logs record events that occur in the application.

- **debug dhcp ipv4 {events | events | packets | snoop {errors | events | internal}}**

- **debug dhcp ipv4 errors**—View error logs.
- **debug dhcp ipv4 events**—View event logs.
- **debug dhcp ipv4 packets**—View packet processing logs.
- **debug dhcp ipv4 snoop errors**—View error logs for DHCP Snoop feature.
- **debug dhcp ipv4 snoop events**—View event logs for the DHCP Snoop feature.
- **debug dhcp ipv4 snoop internal**—View internal debug logs for the DHCP Snoop feature.

Technical Support Commands

The DHCP application has four tech-support commands that call groups of DHCP CLI commands. Use tech-support commands for information about the DHCP application for debugging.

- **show tech-support dhcp ipv4 snoop [bridge-domain** *bridgedomainname* | **profile-name** *profilename*] **file** *filename*
- **show tech-support dhcp ipv4 snoop file** *filename*
- **show tech-support dhcp ipv4 snoop bridge-domain-name** *bridgedomainname* **file** *filename*—View information for the specified bridge domain.
- **show tech-support dhcp ipv4 snoop profile-name** *profilename* **file** *filename*—View information for the specified profile.

Action Commands

Use the following CLI commands to clear DHCP Snoop binding states:

- **clear dhcp ipv4 snoop binding [bridge-domain** *bridgename* | **mac-address** *macaddress*]
- **clear dhcp ipv4 snoop binding**—Clears all DHCP Snoop Client bindings.
- **clear dhcp ipv4 snoop binding bridge-domain** *bridgedomainname*—Clears all DHCP Snoop Client bindings in the specified bridge domain.
- **clear dhcp ipv4 snoop binding mac-address** *macaddress*—Clears the DHCP Snoop Client bindings with the specified MAC address.

L2VPN Commands

DHCP Snoop is enabled on L2VPN ACs by attaching a DHCP Snoop profile to a bridge domain or AC. The DHCP Snoop trusted attribute is configured on a AC according to the value of the trusted attribute in the DHCP Snoop profile. L2VPN CLI commands are used to display the status of DHCP Snoop attributes on L2VPN bridge domains and ACs.

- **show l2vpn bridge-domain bd-name** *bridgename* **detail**—View the L2VPN DHCP Snoop configuration for the specified bridge domain.
- **show l2vpn forwarding interface** *interface* **detail location** *location*—View the L2VPN DHCP Snoop configuration for a specific interface.

Release 3.7.2 Final Draft?Cisco Confidential

UIDB Commands

The UIDB contains network processor configuration. When DHCP Snoop is enabled on a L2VPN AC, there is a DHCP Snoop trusted bit that is set in the ingress and egress entries in the UIDB for each AC. Also, there is a DHCP Snoop enabled bit that is set in the ingress entries in the UIDB for each AC.

- **show uidb data location** *location* **filename ingress**—View the Ingress UIDB DHCP Snoop configuration for the specified interface.
- **show uidb data location** *location* **filename egress**—View the Egress UIDB DHCP Snoop configuration on the specified interface.

L2SNOOP Commands

L2SNOOP receives and transmits DHCP Snoop packets between NETIO and the DHCP Snoop application on the RSP.

show l2snoop statistics pcb all—View the L2SNOOP DHCP Packet Rx/Tx statistics to/from the DHCP Snoop Application on the RSP.

NETIO Commands

NETIO receives and transmits DHCP Snoop packets between the fabric and L2SNOOP on the RSP.

- **show netio clients**—View the NETIO statistics for Rx/Tx DHCP Packets to/from L2SNOOP on the RSP.
- **show netio chains ifhandle 0x08000000**—View the NETIO statistics for Rx/Tx DHCP Packets to/from the fabric on the RSP.

NP Commands

Network processors receive DHCP Packets from the interface controllers and sent them to the RSP or forward them depending on the UIDB configuration. The network processors also receive the DHCP Packets from the fabric that were inject by the DHCP Snoop application on the RSP.

show controller NP counters all—View the NP statistics for DHCP packets that are punted to the fabric by the NP on the LC.

Interface Controller Commands

Interface controllers receive and transmit DHCP Snoop packets between the wire and the network processors.

show controllers *interface* **stats**—View the Interface Controller statistics that include DHCP packets that are transmit and receive from the wire.

Ethernet Operations, Administration, and Maintenance (EOAM) Manageability

This section contains the following:

Discovery Operational Status is Local/Remote Reject

Using the following commands, ensure that the require remote parameter is enabled on one port but not the other:

- **`show ethernet oam discovery`**—View the neighbor discovery status.
- **`show ethernet oam configuration`**—View the configuration.

Using the following command, ensure that the Require remote loopback support parameter is enabled on one port and remote loopback is disabled on the other:

`show ethernet oam configuration`—View the configuration.

Link Monitor Events Are Not Triggering

- **`show ethernet oam platform linkmon trace`**—View link monitoring traces.
- **`show ether-ctrl trace configuration`**—View Ethernet controller link monitoring settings, ensure that the window size and threshold are correct.
- **`show controllers`**—View interface error counters. Verify the errors are detected.

No CCMs are Received at MEP

- **`show ethernet cfm location main`**—Ensure that MEPs are properly configured.
- **`show ethernet cfm peer mep`**—If CCMs are not received then, the peer will not appear.
- **`show uidb`**—Ensure that CFM is enabled for that interface in the NP.
- **`show spp sid stats`**—Ensure that SPP SID stats to see CFM traffic is injected and punted.
- **`show control np stats`**—From the RSP, ensure that the NP hosts the MEP.
- **`show spp client`**—From the RSP, look for SPP drops.
- **`debug ethernet cfm packets`**—Enable debug for all packet types.

Release 3.7.2 Final Draft?Cisco Confidential

Higher Level Packets to a Lower Level MEP

Higher level CFM packets that exceed the configured threshold will be forwarded by the NP. The results of the show interface count should match the incoming packets.

- **show int**—View interface statistics.
- **show control np counters**—View aggregate statistics at NP.

Disabled CCM at MEP

CFM Continuity Check Messages are unidirectional. If CCM is disabled at a MEP, the peer MEPs will not receive CCM and they will time out the entry for the source MEP.

show ethernet cfm peer meps

Receiving “not found” Error

- **run ethernet cfm**—View all CFM global configuration.
- **show ethernet cfm location main**—View local MEPs and their CCM statistics.
- **show ethernet cfm peer meps**—View CFM CCM received from Peer MEPs.
- **show ethernet cfm location ccm**—View CCM database entries.
- **ping ethernet cfm**—Performs CFM ping.
- **ping trace ethernet cfm**—Performs CFM trace.

Dropped Packets

- **show ethernet cfm interfaces statistics**—View per interface drop statistics of CFM PDUs.
- **show spanning-tree mst mstp**—View MST status.
- **show ethernet cfm peer meps**—View Peer MEPs seen by every local MEP.

Internet Protocol (IP)

This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show cef location** *node-id*
2. **show cef ipv4** *{prefix/mask}* **location** *node-id*
3. **show bgp summary**
4. **show bgp [{ipv4 | all} {unicast | multicast | all}] dampened-paths**
5. **show bgp [{ipv4 | all} {unicast | multicast | all}] flap-statistics [regexp** *regular-expression* | *filter-list access-list* | *cidr-only* | *{ip-address [{mask | /prefix-length} [longer-prefixes]]* **[detail]***}]*
6. **show arp [vrf** *vrf-name] [ip-address* **[location** *node-id]* | *hardware-address* **[location** *node-id]* | **traffic [location** *node-id* | *interface-instance]*
7. **show ip route**
8. **show interfaces interface_num accounting**
9. **show cef ipv4** *{prefix/mask}* **hardware {ingress | egress} location** *node-id*
10. **show cef platform trace common**
11. **show cef platform resource {all | np0 | np1 | np2 | np3} location** *node-id*
12. **debug cef platform common all location** *node-id*
13. **debug cef platform common error location** *node-id*
14. **debug cef platform common events location** *node-id*
15. **debug cef platform common info location** *node-id*
16. **debug cef platform common ipfrr location** *node-id*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`show cef location node-id`
	
	Example:
	`RP/0/RSP0/CPU0:router# show cef location 0/2/CPU0`
		View all IPv4 routes of Cisco's Express Forwarding (CEF) on a Line Card (LC).
		
		Note Use this when there are only a few routes.

Step 2	`show cef ipv4 {prefix/mask} location node-id`
	
	Example:
	`RP/0/RSP0/CPU0:router# show cef ipv4 192.1.1.1/32 location 0/2/CPU0`
		View a prefix's route on an LC.

	Command or Action	Purpose
Step 3	**show bgp summary**	View Border Gateway Protocol (BGP) neighbors without an inbound and outbound policy for each active address family. **Note** Use this when there are many routes.
Step 4	**show bgp** *[{ipv4 | all} {unicast | multicast | all}] dampened-paths* **Example:** `RP/0/RSP0/CPU0:router# show bgp dampened-paths`	View which routes have dampening enabled.
Step 5	**show bgp** *[{ipv4 | all} {unicast | multicast | all}] flap-statistics [regexp regular-expression | filter-list access-list | cidr-only | {ip-address [{mask | /prefix-length} [longer-prefixes]] [detail]}]* **Example:** `RP/0/RSP0/CPU0:router# show bgp flap-statistics`	View BGP flap statistics. **Note** Use this for routes that have had dampening enabled. If you do not specify arguments or keywords, all routes for the address family display. If you enter an IP address without mask or prefix length the longest matching prefix displays.
Step 6	**show arp** *[vrf vrf-name] [ip-address [location node-id] | hardware-address [location node-id] | traffic [location node-id | interface-instance]* **Example:** `RP/0/RSP0/CPU0:router# show arp`	View Address Resolution Protocol (ARP) records. For bundle and VLAN-on-Bundle interfaces enter **location** *node-id*. This tells the system which cache entries to display.
Step 7	**show ip route** **Example:** `RP/0/RSP0/CPU0:router# show ip route`	View route entries on the route switch processor (RSP).

Step 8	**show interfaces interface_num accounting**
	Example:
	`RP/0/RSP0/CPU0:router# show interfaces gigabitEthernet 0/1/0/2 accounting`
	`GigabitEthernet0/1/0/2`
		View packet accounting on an interface per protocol.

Step 9	**show cef ipv4** *{prefix/mask} hardware {ingress | egress} location node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 38.1.1.2/32 hardware egress location node-id`	View IPv4 prefix/route in the hardware of an LC. This information helps determine if the destination's IP or prefix action is COMPLETE, PUNT or DROP.
Step 10	**show cef platform trace common** **Example:** `RP/0/RSP0/CPU0:router# show cef platform trace common all errors location node-id`	View common Dynamic Link Library (DLL) code traces.

	Command or Action	Purpose
Step 11	**show cef platform resource location** **Example:** `RP/0/RSP0/CPU0:router# show cef platform resource np0 location node-id`	View the maximum resources of each hardware data structure and if Out Of Resource (OOR) limits have been reached.
Step 12	**debug cef platform common all location node-id**	Activates all debugs in common DLL code.
Step 13	**debug cef platform common error location node-id**	View errors in common DLL code.
Step 14	**debug cef platform common events location node-id**	View major events in common DLL code.
Step 15	**debug cef platform common info location node-id**	View information messages.
Step 16	**debug cef platform common ipfrr location node-id**	View IP fast reroute (IPFRR) messages.

Traffic Loss

Verify packet loss using the **show interface accounting** command or examining Rx packets on the destination.

- **show controllers NP counter {np0 | np1 | np2 | np3} location node-id—View the various counters in the respective NPs.**
- **show cef {ipv4}** *(destination-ip)/(mask)* **hardware egress detail location** *node-id*—View the hardware data structures involved with the prefix *(destination-ip)/(mask)*.
- **show arp location** *node-id*—View the ARP information on the particular LC or RSP.
- **show cef trace error all reverse location** *node-id*—View any PI code ltrace errors recorded.
- **show cef platform trace {ipv4} error all reverse location** *node-id*—View any platform code ltrace errors in protocol ipv4.

Below are the counters in ucode indicating a successful packet transfer from ingress to egress:

- PARSE_ENET_RECEIVE_CNT —View the packets entering on the input interface.
- MODIFY_FABRIC_TRANSMIT_CNT —View the packets transmitted to the egress LC.
- PARSE_FABRIC_RECEIVE_CNT—Display the packets received from the fabric on the egress LC.
- MODIFY_ENET_TRANSMIT_CNT—Display the packets sent outside the egress interface.

For a no-drop traffic case, all of these counters should match. If there is any discrepancy, the investigation should accordingly proceed. For example, if you see fewer instances of PARSE_FABRIC_RECEIVE_CNT than MODIFY_FABRIC_TRANSMIT_CNT, packet loss is happening inside the fabric. More troubleshooting needs to be done on why fabric is dropping packets.

Packets Dropped in Hardware

show controller NP counters {NP0 | NP1 | NP2 | NP3} location *node-id*—View NP counters.

If IPV4_PLU_PUNT_EXCD is incrementing, there is not a COMPLETED adjacency for the next hop. This means ucode keeps punting these packets. If this exceeds the rate limit, packets are dropped.

An IPV4_TTL_ERROR means the Cisco ASR 9000 Series Router is receiving packets with a ttl of either 1 or 0 and ucode drops them.

Packets Switched in Software

Step 1 Verify the hardware chains for the destination IP address are pointing to either of the following:

- COMPLETE adjacency—A valid outgoing path exists.
- PUNT adjacency—The hardware does not know how to send the packet out, it just punts the packet to be switched in software. If the transmit adjacency is PUNT, this could be because ARP is not resolved yet.

Step 2 Show if an ARP entry exists for the destination IP.

```
RP/0/RSP0/CPU0:router# show arp location node-id
```

a. If an ARP entry does not exist or is incomplete, add a static ARP entry. Ensure that the transmit adjacency points to 'COMPLETE'.

```
RP/0/RSP0/CPU0:router# show cef {ipv4} 192.1.1.1/32 hardware egress detail location
node-id
```

b. If so, then it means the issue is that of ARP entry not getting updated. The triage should now focus on why the ARP entry is not getting added. (this includes steps like **show arp**, **show arp idb**, **show adjacency gig node-id detail location node-id**, **show arp trace …** etc).

c. However, if the transmit adjacency still points to 'PUNT', it means ARP is adding the entry in its database but some how fib_mgr fails to mark the adjacency as 'COMPLETE'.

d. This could be a fib_mgr, ARP or AIB problem. Delete and reconfigure the static ARP entry with AIB and CEF debugs on. The debugs show if ARP is adding the entry inside the AIB and if the AIB is informing fib_mgr.

Step 3 Packets could be dropped in the fabric. To verify this, view the fabric counters.

Workaround

Step 1 Use the **shut** and **no shut** command on the outgoing interface.

Step 2 Add a static ARP entry for the destination IP.

Traceroute Fails

Use traceroute to verify the connectivity to a destination. When traceroute fails to a destination, use the following commands:

- **show cef {ipv4}** *(destination_ip)/(mask)* **hardware egress detail location** *node-id*—View the hardware data structures involved with the prefix
- **show interface** *(outgoing_interface)* **accounting**—View input and/or output packets from the outgoing_interface.
- **show controller NP counter {np0 | np1 | np2 | np3} location** *node-id*—View counters inside the respective {np0 | np1 | np2 | np3}.

Release 3.7.2 Final Draft?Cisco Confidential

Step 1 Check if the destination ip address has the proper transmit adjacency. See the 'Tx Adjacency' state (it should be 'COMPLETE').

```
RP/0/RSP0/CPU0:router# show cef {ipv4} <prefix> hardware egress detail location node-id
```

Step 2 If the transmit adjacency is not complete, there is an issue. If it is pointing to 'PUNT', that means probably the mac-address corresponding to the destination IP has not been learnt. Try adding a 'static arp' entry and see if transmit adjacency moves to 'COMPLETE'. If the destination IP is advertised by a routing protocol such as OSPF, then the transmit adjacency should never show as 'PUNT'.

If the transmit adjacency is shown as 'DROP', that means there is a static route to the destination IP explicitly pointing the route to a DROP.

If the transmit adjacency is shown as 'COMPLETE', it means there is no problem in the hardware chains that are setup. You should see the counters.

Step 3 Look for drops in the hardware (NP).

```
RP/0/RSP0/CPU0:router# show controller NP counter {np0 | np1| np2| np3} location node-id
```

Step 4 See if output packets are equal the traceroute packets sent.

```
RP/0/RSP0/CPU0:router# show interface outgoing_interface accounting
```

Workaround

Step 1 Use the **shut** and **no shut** command on the outgoing interface.

Step 2 Add a static ARP entry for the destination IP.

Adding Routes Fails

During OOR (Out Of Resource) the router does not accept more routes until existing routes are deleted.

- **show cef platform resource {all| np0 | np1| np2| np3} location** *node-id*—View usage details of various hardware data structures.
- **show cef resource location** *node-id*—View PI state of various data structures, ideally the state should be GREEN. If it is either YELLOW or RED, it means we entered a condition called OOR (Out Of Resource).
- **show cef platform trace {ipv4} error reverse location** *node-id*—View platform ltrace errors for protocols ipv4.
- **show cef platform trace common error reverse location** *node-id*—View platform ltrace common errors for all protocols.

Step 1 Determine if the router is in OOR state by executing this command, if the state is either YELLOW or RED, it means OOR.

```
RP/0/RSP0/CPU0:router# show cef platform resource location node-id
```

Step 2 Determine which resource exactly is OOR, compare ‘max entries’ and ‘used entries’ too see which of the data structures is using the entries close to the max limit.

```
RP/0/RSP0/CPU0:router# show cef resource location node-id
```

Step 3 After determining which data structure is out of resource, you can verify if it is expected or unexpected. Usually, for each LEAF (either IPv4), it requires 4 entries of NR_LDI structure. So if you find the NR_LDI structure going OOR, see if you have appropriate number of IP leaves to take this NR_LDI number to such a limit.

Step 4 If **show cef resource location node-id** shows the state in ‘GREEN’, then it is not an OOR condition. The reason for not being able to add further routes is some thing else. You may have to enable the following debugs to observe what is happening:

a. `RP/0/RSP0/CPU0:router# debug cef platform common error location node-id`

b. `RP/0/RSP0/CPU0:router# debug cef platform {ipv4} error location node-id`

c. `RP/0/RSP0/CPU0:router# debug cef error location node-id`

d. `RP/0/RSP0/CPU0:router# debug cef {ipv4} error location node-id`

e. If you observe any tracebacks, decode the tracebacks by using SBT tool.

Workaround

If it is OOR condition and expected, delete some existing routes.

Continuous Tracebacks

- **show cef platform trace common all all reverse location** *node-id*—View all platform ltrace common messages.
- **show cef platform trace {ipv4} all all reverse location** *node-id*—View all platform ltrace protocol messages for ipv4.
- **show controller NP drvlog location** *node-id*—View the NP driver logs.

Step 1 When the tracebacks appear continuously on the console (typically after every 15 seconds), programming of the entry inside the hardware is not successful. This causes the software to try repeatedly after every 15 seconds.

Step 2 Ensure that prm_server (PRM is a layer just above hardware) is up: **show controller NP summary**

- If prm_server is down, rectify that problem.
- If prm is down, use **show controllers np drvlog location node-id** to find out of there were NP initialization errors. If there are, it is likely an NP problem.

Workaround

Step 1 Restart prm_server process.

Step 2 Reboot LC.

fib_mgr Does Not Start

Fib_mgr depends on underlying hardware. If the underlying process or hardware does not come up, it is likely fib_mgr does not.

- **show controller NP summary {all |np0 |np1 |np2 |np3}location** *node-id*—View if NP/PRM is up.
- **show controller NP drvlog location** *node-id*—View NP driver logs.
- **show cef platform trace common all reverse location** *node-id*—View platform ltrace common messages.
- **show cef platform trace common event reverse location** *node-id*—View platform ltrace common events.
- **show cef platform trace {ipv4 or mpls} error reverse location** *node-id*—View platform ltrace error messages recorded for protocols ipv4 or MPLS.
- **show cef trace all reverse location** *node-id*—View all PI CEF ltrace messages.

Step 1 When **show controller NP summary {all |np0 |np1 |np2 |np3} location** *node-id* or **show controller NP drvlog location** *node-id* tells that either PRM or underlying NP has a problem, then fib_mgr will not come up. Troubleshoot at the PRM layer or NP layer.

Step 2 Collect the core file and decode the tracebacks using the SBT.

Workaround

Step 1 Restart prm_server process.

Step 2 Reboot LC.

CEF Entries Out of Sync

The cef entry on RSP may be pointing to the management interface and as a result the traffic originating from the router may go out on the management interface instead of through the LC interface.

- **show cef trace event reverse location** *node-id*—View PI cef ltrace events.
- **show cef trace error reverse location** *node-id*—View PI cef ltrace errors.
- **show arp-gmp trace location** *node-id*—View arp-gmp traces.
- **show arp trace location** *node-id*—View ARP traces.
- **show cef platform trace common event reverse location** *node-id*—View CEF platform common event traces.

- **show cef platform trace common error reverse location** *node-id*—View CEF platform common error traces.

Step 1 Look for a default route 0.0.0.0/0 configured to go out via management interface.

Step 2 Look for a static ARP configured for the prefix in question. It is possible that ARP is installing two entries via both management interface and also through the LC interface (since the prefix is reachable by both routes).

Step 3 If the above is not the case, see if some one is advertising an ARP entry through management interface. (show arp should tell that). After finding out the culprit, clear the ARP and verify the cef entries again.

Workaround

- Perform shut and no shut on the management interface.
- **clear arp-cache**
- Reboot the LC.

fib_mgr Crashes

- **show cef platform trace common all reverse location** *node-id*—View CEF platform common traces.
- **show cef platform trace common event reverse location** *node-id*—View CEF platform common event traces.
- **show cef platform trace common error reverse location** *node-id*—View CEF platform common error traces.
- **show cef platform trace {ipv4 or mpls} error reverse location** *node-id*—View CEF platform protocol traces for ipv4 or MPLS.

Step 1 If the trigger is a prm restart or crash, this is expected.

Step 2 If the underlying process (prm_server) is down or crashed, it is likely fib_mgr will not come up.

Step 3 Save the core file.

Step 4 Use the SBT to decode the tracebacks= From root of workspace, use ./util/bin/sbt -p (process_name) -f (log_file).

Step 5 Save the console logs.

Workaround

Restart of fib_mgr or reboot LC.

Release 3.7.2 Final Draft?Cisco Confidential

Tracebacks Appearing

This will be a scenario where there will be a few error tracebacks appearing on the console as a result of some trigger (like interface shut/no shut, or any other trigger like this).

- **show cef trace event location** *node-id*—View CEF traces for major events.
- **show cef trace error location** *node-id*—View CEF traces for major errors.
- **show cef platform trace common error location** *node-id*—View CEF platform traces for common errors across all protocols.
- **show cef platform trace {ipv4 or mpls} error location** *node-id*—View CEF platform traces for errors in protocols ipv4 or MPLS.
- **show logging**

Step 1 Decode the tracebacks using the SBT tool From root of workspace, use ./util/bin/sbt -p (process_name_ -f (log_file)

Step 2 Save core files.

Workaround

If the traceback is service-impacting, do the following:

Step 1 Restart the fib_mgr process.

Step 2 Reboot the LC.

Traffic Loss Due to Changing encap on a Subinterface

When traffic is being forwarded through an l3-subinterface and if the encapsulation is changed on that subinterface, it is some times seen that the traffic will not resume until after 15 seconds.

- **show cef trace event reverse location** *node-id*—View CEF trace messages for major events.
- **show cef trace error reverse location** *node-id*—View CEF trace messages for major errors.
- **show cef platform trace common error location** *node-id*—View CEF platform traces for common errors across all protocols.
- **show cef platform trace {ipv4 or mpls} event location** *node-id*—View CEF platform traces for major events in protocols ipv4 or MPLS.
- **show cef platform trace {ipv4 or mpls} error location** *node-id*—View CEF platform traces for major errors in protocols ipv4 or MPLS.
- **show arp trace location** *node-id*—View ARP related traces.
- **show arp-gmp trace location** *node-id*—View arp-gmp related traces.
- **show arp location** *node-id*—View ARP-related information.

When encapsulation changes from **dot1q vlan 300** to **dot1q vlan 200** on the subinterface, fib_mgr deletes all prefixes corresponding to this interface and creates them again. It takes 15 seconds to add all prefixes; traffic does not get forwarded for that time. For example, there is an interface with address 192.0.0.0/8. There is a static ARP entry for 192.2.2.2.

```
RP/0/RSP0/CPU0:router#show run | inc arp
```

The delay is less likely to happen with regular adjacency (not the static ARP).

When VLAN color changes the following occurs:

- The adjacency is deleted, the adjacency route 192.2.2.2 is deleted.
- The connected route is deleted.
- The adjacency is added before the connected route is added. The FIB treats adding an adjacency without a covering connected route as an error, the route 192.2.2.2 is placed in retry.
- The connected route 192.0.0.0/8 is added.
- Because the FIB retry timer is 15 seconds, the adjacency route 192.2.2.2 is added after 15 seconds.

Workaround

Remove the static ARP entry.

Traffic Loss during RSP Fail Over

Sometimes RSP fail over causes traffic loss. This may mean the IGP over which the prefixes are learned is going down. The following assumes OSPF as the IGP.

- **show ospf** *process-id* **trace**—View OSPF major NSF related traces during failover.
- **show process failover**—View process details during failover.
- **debug ospf ha**—Enables OSPF HA related debugs.
- **debug ospf** *instance* **nsf**—Before FO (Fail Over) and collect the debug log.
- **show process failover**—After FO.
- **show ospf trace ha**—After FO.

Step 1 Check if the next hop router had an FO.

a. If so, the OSPF will go down.

a. If not, verify **nsf cisco** is configured under OSPF.

If so, see if the next hop is reachable during FO.
If not, a link may be going down or having negotiation problems.

Workaround

Reload the router.

Release 3.7.2 Final Draft?Cisco Confidential

IP Multicast

IP Multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to thousands of corporate recipients and homes. This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show igmp {global-interface | groups | ifrs | interface | nsf | old-output | snooping | ssm | standby | summary | trace | traffic | vrf** *name***}**
2. **show pim {bgp-safi| bsr | context| df| global | group-map | ifrs| interface | ipv4 | join-prune | mdt | mib-database | mstatic | multicast | neighbor | nsf | old-output | range-list | rpf | safi-all | standby | summary | table-context | topology | trace | traffic | tunnel | unicast | vrf}**
3. **show mrib {client | route | table-info}**
4. **show mfib {connections | counter}**
5. **show mfib hardware {connection | interface | ltrace | resource-counters} location** *node-id*
6. **show mfib hardware route {accept-bitmap | statistics | summary} {* | A.B.C.D | A.B.C.D/length | detail | hex-dump} location** *node-id*
7. **show mfib hardware route summary location** *node-id*
8. **debug mrib errors**
9. **debug mrib events**
10. **debug netio error**
11. **debug mfib warning**
12. **debug mfib errors**
13. **debug mlib errors**
14. **debug mlib warning**
15. **show mrib trace**
16. **show mfib trace**
17. **show mlib trace**

Release 3.7.2 Final Draft?Cisco Confidential

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show igmp {global-interface | groups | ifrs | interface | nsf | old-output | snooping | ssm | standby | summary | trace | traffic | vrf** *name*} **Example:** `RP/0/RSP0/CPU0:router# show igmp groups`	View all Internet Group Management Protocol (IGMP)-related information in the control plane. IGMP is a protocol used by IPv4 systems to report IP multicast memberships to neighboring multicast routers. Filter results using the following parameters and keywords: • **global-interface**—IGMP global Interface Descriptor Block (IDB) data structures, IDBs have information like IP addresses, interface states, and packet statistics. There is one IDB for each interface and one for each subinterface. • **groups**—IGMP group memberships. • **ifrs**—Interfaces in Forward Reference Store. • **interface**—IGMP interface information. • **nsf**—Current multicast NSF state for IGMP, either normal or activated for NSF. The latter state indicates that recovery is in progress due to an IGMP failure. The total NSF timeout and time remaining are displayed until NSF expiration. • **old-output**—Provides backward compatibility. • **snooping**—IGMP snooping parameters. • **ssm**—Source Specific Multicast (SSM)-related information. • **standby**—Standby process. • **summary**—IGMP summary. • **trace**—IGMP ltrace data. • **traffic**—IGMP traffic counters. • **vrf** *name*—Specify a Virtual Private Network (VPN) routing and forwarding (VRF).

	Command or Action	Purpose
Step 2	**show pim {bgp-safi| bsr | context| df| global | group-map | ifrs| interface | ipv4 | join-prune | mdt | mib-database | mstatic | multicast | neighbor | nsf | old-output | range-list | rpf | safi-all | standby | summary | table-context | topology | trace | traffic | tunnel | unicast | vrf}** **Example:** `RP/0/RSP0/CPU0:router# show pim neighbor`	View Protocol Independent Multicast (PIM)-related information in the control plane. Filter results using the following parameters and keywords: • **bgp-safi**—Border Gateway Protocol (BGP) secondary address family (SAFI) database. • **bsr**—PIM Bootstrap Router (BSR) information. • **context**—PIM VRF Contexts. • **df**—Bidirectional Designated Forwarder (DF). • **global**—Interfaces in Global tree. • **group-map**—PIM group-to-protocol mapping information. • **ifrs**—Interfaces in Forward Reference Store. • **interface**—PIM interface information. • **ipv4**—IPv4 Address Family. • **join-prune**—PIM Join/Prune information. • **mdt**—Data MDT information. • **mib-database**—PIM group db for mroute, the mroute state is maintained by multicast routers for the incoming and outgoing interfaces for each source and group (S,G) pair, it is used to determine which packets are discarded or forwarded. • **mstatic**— Multicast Static Route information. • **multicast**—SAFI Multicast. • **neighbor**—PIM neighbor information. • **nsf**—Non-stop forwarding. • **old-output**—Provides backward compatibility. • **range-list**—PIM range-list information. • **rpf**—RPF information. • **safi-all**—SAFI wildcard. • **standby**—Standby process. • **summary**—PIM summary information. • **table-context**—PIM Table context. • **topology**—PIM topology table information. • **trace**—PIM trace data. • **traffic**—PIM traffic counters. • **tunnel**—Tunnel interfaces. • **unicast**—SAFI Unicast. • **vrf**—VRF.

	Command or Action	Purpose

Step 3	`show mrib {client | route | table-info}`
	
	Example:
	`RP/0/RSP0/CPU0:router# show mrib client`
		View Multicast Routing Information Base (MRIB) information. Filter results using the following parameters and keywords:
		• **client**—MRIB client connections.
		• **iir-interface**—MRIB IIR interface database.
		• **ipv4**—IPv4 Address Family.
		• **nsf**—Non-stop forwarding.
		• **old-output**—Provides backward compatibility.
		• **platform**—Platform-specific data.
		• **route**—Routing database.
		• **route-collapse**—MRIB Route Collapse database.
		• **table-info**—MRIB VRF table information.
		• **tlc**—MRIB Table-LC database.
		• **trace**—Trace data.
		• **vrf**—VRF.

Step 4	`show mfib {connections | counter | encap-info | hardware | interface | ipv4 | ma | mdt | nsf | route | svd | table-info | trace | vrf)`
	
	Example:
	`RP/0/RSP0/CPU0:router# show mfib nsf`
		View Multicast Forwarding Information Base (MFIB) information in the control plane. Filter results using the following parameters and keywords:
		• **connections**—Status of MFIB connections to servers.
		• **counter**—MFIB global counters.
		• **encap-info**—Multicast Virtual Private Network (MVPN) Encap information.
		• **hardware**—Cisco ASR 9000 Series Router hardware.
		• **interface**—MFIB interface specific information.
		• **ipv4**—IPv4 Address Family.
		• **ma**—Multicast Forwarding (MFWD) management agent process exists on each LC and assigns hardware counters to each (S, G) route.
		• **mdt**—MDT tunnel information.
		• **nsf**—Multicast NSF status.
		• **route**—Routing database.
		• **svd**—Singular Value Decomposition (SVD) events.
		• **table-info**—Table information.
		• **trace**—MFIB traces.
		• **vrf**—VRF.

	Command or Action	Purpose
Step 5	**show mfib hardware {connection | interface | ltrace | resource-counters} location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show mfib hardware connection location node-id`	View all hardware data in the multicast PD. Filter results using the following parameters and keywords: • **connection**—MFIB connections to server. • **interface**—Cisco ASR 9000 Series Router hardware. • **ltrace**—IP Multicast platform specific trace information. • **resource-counters**—Allocated and freed hardware resources. • **route**—Platform-specific information for the routing database. • **location**—Specify the MFIB location. **Note** The output of these commands can be large when there are a large number of routes and olists configured.
Step 6	**show mfib hardware route {accept-bitmap | statistics | summary} {* | A.B.C.D | A.B.C.D/length | detail | hex-dump} location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show mfib hardware route olist location node-id`	View all hardware ROUTE data in the multicast PD. Filter results using the following parameters and keywords: • **accept-bitmap**—Accepting interface list for bidir routes. • **olist**—Output interface list (olist) stored in the hardware. • **statistics**—Per route packets and bytes counters. • **summary**—Summary of routes. • *****—Shared tree entries. • **A.B.C.D**—Source/group IP address. • **A.B.C.D/length**—Group IP address/prefix length. • **detail**—Details of each route (requires 140 columns). • **hex-dump**—Hex dump of the PLU and TLU. • **location**—Specify the MFIB location. **Note** The output of these commands can be large when there are a large number of routes and olists configured.
Step 7	**show mfib hardware route summary location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show mfib hardware route summary location node-id`	View all hardware ROUTE data in the multicast PD. Filter results using the following parameters and keywords: • **summary**—Summary of routes. • **location**—MFIB location.
Step 8	**debug mrib errors** **Example:** `RP/0/RSP0/CPU0:router# debug mrib errors`	To monitor Multicast Routing Information Base (MRIB) internal errors, use the debug mrib errors command in EXEC mode. To disable debugging output, use the no form of this command.
Step 9	**debug mrib events**	—

	Command or Action	Purpose
Step 10	`debug netio error`	—
Step 11	`debug mfib warning`	—
Step 12	`debug mfib errors`	—
Step 13	`debug mlib errors`	—
Step 14	`debug mlib warning`	—
Step 15	`show mrib trace`	—
Step 16	`show mfib trace`	—
Step 17	`show mlib trace`	—

Multicast Pie Installation Fails

Step 1 View detailed information for the specified install id.

```
RP/0/RSP0/CPU0:router# show install log [1-4294967295] detail
```

Step 2 Ensure that the pie file name is correct and reissue the command.

Step 3 Ensure that the location of the pie file is correct and reissue the command

Step 4 Ensure that the pie file has proper permissions (755) and reissue the command.

Step 5 If you are loading from the TFTP directory, ensure that the following are true:

a. The router has network connectivity.

b. The TFTP address is properly configured.

c. The TFTP server has connectivity.

```
RP/0/RSP0/CPU0:router# ping tftp-server-addr
```

d. If loading locally from a router, ensure that the pie file is stored on the router.

Step 6 Verify all nodes are in the “IOS XR RUN State”

```
RP/0/RSP0/CPU0:router# show platform
```

Multicast CLI Unavailable Although Pie Is Installed

`show install active`—View active package information

Ensure that the following are correct:

- Pie file name.
- Pie file location.
- Pie file installation on ALL nodes.

Release 3.7.2 Final Draft?Cisco Confidential

"This command not authorized" Error Message

While issuing certain commands in config or EXEC mode, the "This command not authorized" error message appears, disabling further access. This means the user does not have privileges.

show config run—(from admin mode) View current operating admin configuration of the system.

Dynamic IGMP Failure

Dynamic IGMP failure i.e., Dynamic (*,G) are timing out. The groups and routes are configured and set up correctly but when traffic is sent to Cisco ASR 9000 Series Router from tester, it is not received at the Rx tester port.

debug igmp groups—View IGMP group activity; check if IGMP group times out.

show igmp traffic—View IGMP control plane activity; verify that Cisco ASR 9000 Series Router is sending out packets.

Step 1 One possible cause could be that IGMP group is timing out. One way to check is to create a static route for the (*,G) and see if traffic is now received. If it is, it means that the groups are timing out.

Step 2 Decrease the query interval (resulting in more queries per minute):

```
RP/0/RSP0/CPU0:router# conf t
RP/0/RSP0/CPU0:router(config)# router igmp
RP/0/RSP0/CPU0:router(config-igmp)# query-interval 1
RP/0/RSP0/CPU0:router(config-igmp)# commit
```

Step 3 Ensure that packets are going out of interface at the interval set.

Step 4 Ensure the tester responds with an IGMP membership report.

Workaround

Configure static groups as a temporary workaround.

Multicast Packets Dropping—General Forwarding Plane Debug

Traffic gets dropped in a multicast setup. To localize the problem, debug the counters step by step from ingress to egress.

Step 1 Use the **show counters** command.

Step 2 Send a packet burst of 10000 packets and verify the counters at each stage. While starting multicast traffic for a new route/group, there will always be some drops initially as the packets are punted to RP for lookup and an (S,G) is created if needed. These few initial drops are expected.

Release 3.7.2 Final Draft?Cisco Confidential

Traffic Fails on Some Interfaces

Traffic is failing on some interfaces or channels. The groups and routes are configured and set up correctly. Traffic is sent to Cisco ASR 9000 Series Router from tester. It is received correctly on some interfaces but not on others or some video channels are received correctly on an interface while others are not. Possible causes could be:

- OLIST may not be properly configured.
- uIDB values not correctly set in hardware.
- MGID on Octopus or Bridge not correctly set up.

Step 1 Ensure that packets are going from ing NP -> fabric -> Eg NP

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location [ingress/ egress]
```

Step 2 View olist interfaces for the route.

```
RP/0/RSP0/CPU0:router# show mfib hardware route olist location [egress]
```

Step 3 View NP counters.

```
RP/0/RSP0/CPU0:router# show controllers NP counter all location [egress]
```

Step 4 View the statistics for the specific route and source.

```
RP/0/RSP0/CPU0:router# show mfib hardware route stat [src ip addr] location [ingress/
egress]
```

Traffic Fails on Some Interfaces—MGID

Step 1
```
RP/0/RSP0/CPU0:router# show mfib hardware
```

Step 2 Ensure that packets are going from ing NP -> fabric -> Eg NP

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location [ingress/ egress]
```

Step 3 View olist interfaces for the route.

```
RP/0/RSP0/CPU0:router# show mfib hardware route olist location [egress]
```

Step 4 View packet lookup structure information on the NP.

```
RP/0/RSP0/CPU0:router# show controllers NP struct 22 all location [egress]
```

Step 5 View MGID programming information on the location.

```
RP/0/RSP0/CPU0:router# show controllers mgidprgm mgidindex [MGID VALUE] location
[ingress/ egress]
```

Step 6 Ensure that packets are transmitted out of ingress NP to fabric and received by egress NP from the fabric.

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location <<ingress/ egress>>
```

Step 7 View the MGID for the route.

```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether node-id location node-id
```

Step 8 View the MGID table.

```
RP/0/RSP0/CPU0:router# show controllers mgidprgm mgidindex <MGID VALUE> location
<<egress>>
```

Throughput Loss at Receiver Interfaces

Traffic is sent and received on routes but there is a loss of throughput at the receiver.

Step 1 Ensure that packets are going from ing NP -> fabric -> Eg NP

```
RP/0/RSP0/CPU0:router# show mfib hardware route statistics location [ingress/ egress]
```

Step 2 The above command tells us if packets are punted to RP. If so, check if the source of that channel is setting some IP options or not.

Layer 2 Tunnel Protocol (L2TP)

L2TP is an Internet Engineering Task Force (IETF) standard that combines the best features of two existing tunneling protocols: Cisco's Layer 2 Forwarding (L2F) and Microsoft's Point-to-Point Tunneling Protocol (PPTP). This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show uidb data location ingress**
2. **show vlan trace** *[file file-name | hexdump | last <n> | reverse | stats | tailf | unique | verbose | wrapping}] [location]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show uidb data location ingress**	View micro-IDB index data information for a specified node. The node-id argument is entered in the *rack/slot/module* notation.
Step 2	**show vlan trace** *[file file-name | hexdump | last <n> | reverse | stats | tailf | unique | verbose | wrapping}] [location]*	—

Packets Incorrectly Dropped/Forwarded Based on Filters

Packets with destination MACs which should be dropped/forwarded are not being filtered properly.

Step 1 `RP/0/RSP0/CPU0:router# show version`

Step 2 Verify L2 bridge/xconnect is up.

`RP/0/RSP0/CPU0:router# show running-configuration`

If not, refer to VPLS/EoMPLS debug guides. ltrace output will contain a lot of PI/PD trace messages. Some applying to the filtering configuration being applied to the interface. This is a success case for filtering dot1ad being applied to Gi0/5/0/1. If failed, there will be errors in trace when applying this configuration.

Step 3 `RP/0/RSP0/CPU0:router# show uidb data location node-id node-id node-id ingress`

Step 4 `RP/0/RSP0/CPU0:router# show vlan trace location node-id`

Step 5 `RP/0/RSP0/CPU0:router# show controllers NP counter {NP0 | NP1 | NP2 | NP3} location node-id`

Workaround

Step 1 Try deleting and adding filtering configuration back to running configuration.

Step 2 Reboot LC.

Link Bundles

A link bundle is a group of ports that are bundled together and act as a single link. The advantages of link bundles are as follows:

- Multiple links can span several LCs to form a single interface. Thus, the failure of a single link does not cause a loss of connectivity.
- Bundled interfaces increase bandwidth availability, because traffic is forwarded over all available members of the bundle. Therefore, traffic can move onto another link if one of the links within a bundle fails. Add or remove bandwidth without interrupting packet flow.

This section contains the following:

Release 3.7.2 Final Draft?Cisco Confidential

Bundle Does Not Come Up

Step 1 Ensure that the bundle is up.

```
RP/0/RSP0/CPU0:router# show interface bundle-ether x
```

Step 2
```
RP/0/RSP0/CPU0:router# show interface
```

Step 3 View LACP statistics.

```
RP/0/RSP0/CPU0:router# show lacp counters
```

Step 4
```
RP/0/RSP0/CPU0:router# show lacp
```

Step 5
```
RP/0/RSP0/CPU0:router# show bundle
```

Step 6
```
RP/0/RSP0/CPU0:router# show tech bundle
```

Step 7
```
RP/0/RSP0/CPU0:router# show controller np counters all location node-id
```

a. Ensure that the bundle state is not “shutdown”.

b. Ensure that the member port is not “shutdown”.

c. Ensure that the port’s MAC address is valid.

d. Ensure that the other side of the link is UP (bundle and members).

If running LACP, ensure that LACP packets are able to transmit and receive accordingly. If LACP packets are not able to transmit and receive accordingly, check interface counters to identify at what stage packets are dropped.

e. Check ucode counters to see if there are any drop counts corresponding to the location of the LC.

Bundle Member Not Distributing

Step 1
```
RP/0/RSP0/CPU0:router# show tech bundle
```

Step 2 Ensure that the member is up.

```
RP/0/RSP0/CPU0:router# show interface node-id
```

Step 3 Find why the member is not distributing.

```
RP/0/RSP0/CPU0:router# show bundles reasons
```

Step 4 Ensure that the remote side is up.

Step 5 Ensure that the LACP parameters are same on both sides.

a. If LACP is enabled, check its status.

```
RP/0/RSP0/CPU0:router# show lacp bundle
```

b. Ensure that bundle members have the same characteristics.

```
RP/0/RSP0/CPU0:router# show running interface interface-name
```

c. If the bundle members have different characteristics, make them all the same.

d. Ensure that LACP packets are transmitted and received.

```
RP/0/RSP0/CPU0:router# debug bundlemgr local packets port node-id
```

Workaround

If the bundle with LACP cannot come up, do the following:

- Use the bundle in non-lacp mode.

```
RP/0/RSP0/CPU0:router# bundle id x mode on
```

- Use one side of bundle in passive mode, the other in active mode. At least one side must be active.

Bundle Not Using MAC-Address From Backplane

Step 1 Ensure that the backplane MAC is programmed.

```
RP/0/RSP0/CPU0:router# show diag chassis eeprom-info
```

Step 2 View a summary of the MAC address.

```
RP/0/RSP0/CPU0:router# show bundle mac-allocation distrib
```

Step 3 Ensure that the lag table is programmed.

```
RP/0/RSP0/CPU0:router# show bundle trace process distrib buffer all location node-id
```

Step 4 `RP/0/RSP0/CPU0:router# show controllers backplane bpe-trace`

Step 5 `RP/0/RSP0/CPU0:router# show tech bundle`

Release 3.7.2 Final Draft?Cisco Confidential

L3 Data Traffic Not Flowing

Regular Interface (No Subinterfaces)

Step 1 View the Address Resolution Protocol (ARP).

```
RP/0/RSP0/CPU0:router# show arp
```

Step 2
```
RP/0/RSP0/CPU0:router# show interface bundle-ether x
```

Step 3 View the running configuration information.

```
RP/0/RSP0/CPU0:router# show running-config
```

Step 4 View the router's micro-IDB index data information.

```
RP/0/RSP0/CPU0:router# show uidb data location node-id bundle-ether egress
```

Step 5 View information about packets forwarded by Cisco Express Forwarding (CEF).

```
RP/0/RSP0/CPU0:router# show cef
```

Step 6
```
RP/0/RSP0/CPU0:router# show cef hardware ingress location node-id
```

Step 7
```
RP/0/RSP0/CPU0:router# show cef hardware egress location node-id
```

Step 8 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller np counters all location node-id
```

Step 9
```
RP/0/RSP0/CPU0:router# show controllers np stats all location node-id
```

a. If traffic is still not flowing check the trace to make sure the lag table is programmed.

```
RP/0/RSP0/CPU0:router# show bundle trace process adjacency buffer all location node-id
```

b. Verify the lag table is programmed properly in hardware.

```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether node-id location node-id
```

Subinterface

Step 1 Troubleshoot L3 IPv4 traffic.

Step 2 Ensure that VLAN traffic coming in matches that on the incoming interface.

Step 3 View ucode counters.

```
RP/0/RSP0/CPU0:router# show controller np counters {np1 | np2 | np3 | all} location node-id
```

Release 3.7.2 Final Draft?Cisco Confidential

Ping Failed over Bundle

Step 1 View the ARP.

```
RP/0/RSP0/CPU0:router# show arp
```

Step 2 View the ARP information on the particular LC or RSP.

```
RP/0/RSP0/CPU0:router# show arp location node-id
```

Step 3 View route entries on the RSP.

```
RP/0/RSP0/CPU0:router# show ip route
```

Step 4 `RP/0/RSP0/CPU0:router# show cef hardware detail location node-id ingress`

Step 5 `RP/0/RSP0/CPU0:router# show interface`

Step 6 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller np counters all location node-id
```

Step 7 Use the hash calculator to determine which bundle member (interface) to test.

Step 8 Remove the interface from the bundle.

Step 9 Assign the interface an IP address.

Step 10 Ping the interface.

Step 11 Ensure that ARP is resolved between the router and the node being pinged.

Step 12 Ensure that the MAC address in the ARP table of the other side corresponds to that on the router.

Step 13 Ensure that the MAC address of the bundle is valid.

Step 14 Ensure that the routing and hardware routing table has an entry to the next hop.

Step 15 Check interface counters to see if ping packets are transmitted and being received on the router member port of the bundle.

Step 16 Check ucode counters to see where packets are dropped on the incoming or outgoing member of the bundle.

Workaround

Step 1 View the ARP.

```
RP/0/RSP0/CPU0:router# static arp
```

Step 2 View the route.

```
RP/0/RSP0/CPU0:router# static route
```

Step 3 Try a different port.

L3 Packets Not Synching Over Bundle

Step 1 View counter information about the Open Shortest Path First (OSPF) routing processes.

```
RP/0/RSP0/CPU0:router# show ospf counters
```

Step 2
```
RP/0/RSP0/CPU0:router# show interface
```

Step 3 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller np counters all location node-id
```

Step 4 Turn on debug of that protocol or look at protocol counters to see if the protocol packets are being sent and received.

Step 5 If the protocol packets are not being sent or received, check the interface counters to see if interface indicates packet in and out.

Step 6 If the interface level indicates that packets are coming in and out but not reaching protocol, check ucode counters to see if there are any drops.

L2 Traffic Not Flowing

VPLS

Step 1 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller np counters all location node-id
```

Step 2 Verify the AC is up.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain
```

Step 3 Verify the bridge domain is up.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain
```

Step 4 Look for MTU mismatches.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

Step 5 Ensure that the lag table is programmed.

```
RP/0/RSP0/CPU0:router# show bundle trace process adjacency buffer all location node-id
```

Step 6 Ensure that the lag table is programmed in the hardware.

```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether node-id location node-id
```

Step 7 Verify the bundle AC is programmed in the bridge member table in the hardware.

```
RP/0/RSP0/CPU0:router# show controllers np struct 28 all location node-id
```

Release 3.7.2 Final Draft?Cisco Confidential

Step 8 Verify the XID is programmed in the hardware.

```
RP/0/RSP0/CPU0:router# show controllers np struct 15 all location node-id
```

Step 9 Check ingress UIDB to ensure that the VP bytes are set.

```
RP/0/RSP0/CPU0:router# show uidb data location node-id Bundle-Ether # ingress
```

a. If VP bytes are all zero collect information.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls all location node-id
```

b. Remove the bundle from the bridge, commit, re-add the bundle and commit again.

Step 10 Ensure that traffic is forwarded to the egress LC.

```
RP/0/RSP0/CPU0:router# show controllers np stats all location node-id
```

VPWS

Step 1 View brief information on configured cross-connects.

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect summary
```

Step 2
```
RP/0/RSP0/CPU0:router# show l2vpn xconnect state
```

Step 3 Ensure that the lag table is programmed.

```
RP/0/RSP0/CPU0:router# show bundle trace process adjacency buffer all location node-id
```

Step 4
```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether node-id location node-id
```

Step 5
```
RP/0/RSP0/CPU0:router# show controllers np struct 15 all location node-id
```

Step 6
```
RP/0/RSP0/CPU0:router# show uidb data location node-id Bundle-Ether
```

Step 7 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller np counters all location node-id
```

Load Balancing Over Bundle

L2 Traffic Does Not Load Balance

Step 1
```
RP/0/RSP0/CPU0:router# show running-config
```

Step 2
```
RP/0/RSP0/CPU0:router# show bundle
```

Step 3
```
RP/0/RSP0/CPU0:router# show interface bundle-ether x
```

Step 4
```
RP/0/RSP0/CPU0:router# show uidb data location node-id type instance ingress
```

Step 5
```
RP/0/RSP0/CPU0:router# show arp router-ids
```

Step 6
```
RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether node-id location node-id
```

Release 3.7.2 Final Draft?Cisco Confidential

L3 Traffic Does Not Load Balance

Step 1 `RP/0/RSP0/CPU0:router# show arm router-id`

Step 2 `RP/0/RSP0/CPU0:router# show bundle`

Step 3 `RP/0/RSP0/CPU0:router# show interface bundle-ether x`

Step 4 `RP/0/RSP0/CPU0:router# show iir interface bundle-ether x`

Step 5 `RP/0/RSP0/CPU0:router# collect { src-ip | dst-ip-of-traffic }`

Step 6 `RP/0/RSP0/CPU0:router# show controllers bundle bundle-ether node-id location node-id`

Step 7 `RP/0/RSP0/CPU0:router# show running-config`

Step 8 `RP/0/RSP0/CPU0:router# show arm router-id`

Step 9 `RP/0/RSP0/CPU0:router# show bundle to`

Step 10 Find out which member should be carrying the traffic out.

```
RP/0/RSP0/CPU0:router# bundle-hash bundle-ether x
```

Step 11 View each member of the bundle to see which member is actually carrying the traffic out.

```
RP/0/RSP0/CPU0:router# show interface x
```

Bundle Statistic

L2 statistics are not supported in the **show interface accounting** command for bundle interfaces in the current release.

Multi Protocol Label Switching (MPLS)

MPLS carries different kinds of traffic, like IP packets, native ATM, SONET, and Ethernet frames. This section contains the following:

Release 3.7.2 Final Draft?Cisco Confidential

Using Show and Debug Commands

SUMMARY STEPS

1. **debug mpls ldp transport event**
2. **debug mpls ldp transport connection**
3. **show mpls forwarding private**
4. **show mpls forwarding tunnels**
5. **debug mpls ea platform {info | errors | events | all } [location]**
6. **debug cef ea errors location <..>**
7. **debug cef ea info location <..>**
8. **debug cef ea event location <..>**
9. **debug cef ea verbose location <..>**
10. **show ltrace af-ea**
11. **debug cef platform mpls all**
12. **debug cef platform mpls error**
13. **debug cef platform mpls events**
14. **debug cef platform mpls info**
15. **debug cef platform mpls trace**
16. **debug cef platform mpls verbose**
17. **debug cef platform common**
18. **debug cef platform adj errors location <..>**
19. **debug cef platform common all location node-id**
20. **debug cef platform common trace location <..>**
21. **debug cef platform common errors location <..>**
22. **debug cef platform common info location <..>**
23. **debug cef platform common events location <..>**
24. **debug cef platform common verbose location <..>**
25. **show ltrace commands**
26. **show cef platform trace [common|ipv4|ipv6|mpls|te|all]**
27. **show cef platform [ipv4|ipv6|mpls|common|te|adj|all]**
28. **show mpls forwarding label hardware egress location**
29. **show mpls forwarding labels private hardware egress location**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`debug mpls ldp transport event`	Discovery and connection setup/shutdown events.
Step 2	`debug mpls ldp transport connection`	Connections setup/shutdown events.
Step 3	`show mpls forwarding private`	View MPLS prefixes with local and outgoing tags and the number of bytes switched for each tagged prefix.
Step 4	`show mpls forwarding tunnels`	—
Step 5	`debug mpls ea platform {info | errors | events | all } [ location]`	—
Step 6	`debug cef ea errors location <..>`	—
Step 7	`debug cef ea info location <..>`	—
Step 8	`debug cef ea event location <..>`	—
Step 9	`debug cef ea verbose location <..>`	—
Step 10	`show ltrace af-ea`	—
Step 11	`debug cef platform mpls all`	—
Step 12	`debug cef platform mpls error`	—
Step 13	`debug cef platform mpls events`	—
Step 14	`debug cef platform mpls info`	—
Step 15	`debug cef platform mpls trace`	—
Step 16	`debug cef platform mpls verbose`	—
Step 17	`debug cef platform common`	—
Step 18	`debug cef platform adj errors location <..>`	—
Step 19	`debug cef platform common all location node-id`	—
Step 20	`debug cef platform common trace location <..>`	—
Step 21	`debug cef platform common errors location <..>`	—
Step 22	`debug cef platform common info location <..>`	—
Step 23	`debug cef platform common events location <..>`	—
Step 24	`debug cef platform common verbose location <..>`	—
Step 25	`show ltrace commands`	—
Step 26	`show cef platform trace [com-mon|ipv4|ipv6|mpls|te|all]`	—
Step 27	`show cef platform [ipv4|ipv6|mpls|com-mon|te|adj|all]`	—
Step 28	`show mpls forwarding label hardware egress location`	—
Step 29	`show mpls forwarding labels private hardware egress location`	—

IP Packets Not Forwarded to LSP

Step 1 Find the NP to which the interface belongs.

```
RP/0/RSP0/CPU0:router# show controllers np ports all
```

Step 2 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller np counters location node-id
```

Step 3 Check the hardware label FIB.

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels label hardware egress location node id
```

Step 4 Find out whether the ARP resolves for the next hop prefix

```
RP/0/RSP0/CPU0:router# show ARP <prefix> location node id
```

Workaround

Change the remote directly connected interface MAC address to trigger adjacency changes on the router.

IP Packets Not Forwarded to MPLS TE Tunnel

Step 1 Find the NP to which the interface belongs.

```
RP/0/RSP0/CPU0:router# show controllers np ports all
```

Step 2 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller np counters location node id
```

Step 3
```
RP/0/RSP0/CPU0:router# show stats counters
```

Step 4 Check the tunnel adjacency of the prefix.

```
RP/0/RSP0/CPU0:router# show cef prefix hardware egress location node id
```

Step 5 Ensure that the MPLS traffic tunnel is up.

```
RP/0/RSP0/CPU0:router# mpls traffic tunnel tunnel-id
```

Step 6 Check the hardware TE label FIB.

```
RP/0/RSP0/CPU0:router# show cef adjacency tunnel-te tunnel-id hardware egress location
node id
```

Workaround:

Enter the **shut** command and the **no shut** command on the tunnel interface to reprogram the hardware.

MPLS Packets Not Forwarded to MPLS TE Tunnel

Step 1 Ensure that the label is pointed to tunnel adjacency (tt1).

```
RP/0/RSP0/CPU0:router# show mpls for private
```

Step 2 Ensure that transmit adjacency is complete.

```
RP/0/RSP0/CPU0:router# show mpls for labels 16004 ha eg location node-id
```

Step 3 Ensure that hardware tunnel adjacency is complete.

```
RP/0/RSP0/CPU0:router# show cef adj tunnel-te 1 hardware eg location node-id
```

Workaround

Perform shut and no shut of the tunnel interface to reprogram the hardware.

Label is Out of Hardware Supported Range

This means the label cannot be inserted into hardware label FIB due to hardware limitation.

LC/node-id:Jul 10 13:08:18.379: fib_mgr[137]: Label is out of hardware supported range label :1048561

LC/node-id:Jul 10 13:08:18.379 : fib_mgr[137]: %PLATFORM-PLAT_FIB_MPLS-3-ERR_STR : hw gtrie leaf insert: Failed to insert leaf to EZError = Invalid argument(22)

LC/node-id:Jul 10 13:08:18.380 : fib_mgr[137]: %ROUTING-FIB-3-PD_FAIL : fib_leaf_insert 4532 Cannot insert in switching leaf 1048561/1 [82909024] type 5 flags 40001 refcnt 1 prot mpls table tableid 0 vrfid 0 ---LDI: [8352d87c] type 6 flags 4101000 refcnt 0 type 4 depth 1 num_slots 1 num_buckets 1 LDI pl 83c20588 ---PL [83c20588] type 7 flags c00d02 refcnt 3 path_cnt 1 max_depth 1 ldi 8352d87c ---: 0x16 Invalid argument : pkg/bin/fib_mgr : (PID=110681) : -Traceback= 4823ac68 4823f16c 48248554 4824b0bc 4828e6e0 48290290 4824b288 4824cc68 48252960 4825369c 48291d24 48292104 4820584c fc1f0694 fc1ee4c0 48200f4c

LC/node-id:Jul 10 13:08:18.384 : fib_mgr[137]: Label is out of hardware supported range label :1048562

LC/node-id:Jul 10 13:08:18.388 : fib_mgr[137]: Label is out of hardware supported range label :1048563

MPLS TE Tunnels Do Not Come Up

Step 1 Look for reasons the tunnel does not come up.

```
RP/0/RSP0/CPU0:router# show mpls tr tunnels
```

Step 2 Ensure that the tunnel egress interface is configured in RSVP

rsvp

interface Bundle-Ether1

Release 3.7.2 Final Draft?Cisco Confidential

```
 bandwidth 100000
!
interface GigabitEthernet0/1/0/2
 bandwidth 100000
!
interface GigabitEthernet0/4/0/8  <<---- tunnel egress interface
 bandwidth 100000
!
interface GigabitEthernet0/4/0/20
 bandwidth 100000
!
```

mpls traffic-eng **<<---- Ensure that the tunnel egress interface is configure in mpls traffic-engineering config**

```
interface Bundle-Ether1
!
interface GigabitEthernet0/1/0/2
!
interface GigabitEthernet0/4/0/8 <<---- tunnel egress interface
!
 interface GigabitEthernet0/4/0/20
!
```

Step 3 Ensure that traffic engineering is configured in OSPF

```
router ospf te
 log adjacency changes detail
 router-id 192.1.1.30
 area 0
  mpls traffic-eng
```

Step 4 Ensure that ping is successful on tunnel destination IP.

FRR-Protected Tunnel Goes Down After Triggering FRR

FRR is a mechanism for protecting MPLS Traffic Engineering (TE) label-switched paths (LSPs) from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

Step 1 Ping the address in an updated sender template.

```
RP/0/RSP0/CPU0:router# ping PLR_Address
```

Step 2 Ensure that the MP address is reachable. Check forwarding over the backup tunnel is working.

```
RP/0/RSP0/CPU0:router# ping backup_tunnel_destination
```

Step 3 Ensure that the backup tunnel is in Up, Up state.

```
RP/0/RSP0/CPU0:router# show mpls traffic-engineering tunnels
```

Step 4 Check RSVP traces to find out why the tunnel went down.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng trace events
```

MPLS TE FRR Database Not Built

Step 1 Ensure that the protected tunnel is fast reroutable.

a. `RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnel property fastreroute`

b. `RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnel protection`

Step 2 Ensure that backup does not pass through protected interface.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnel backup protected-interface
```

Step 3 Ensure that backup has enough backup bandwidth (if configured).

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels backup
```

Step 4 Ensure that backup and protected tunnels have a merge point (check hop information).

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels
```

Step 5 Ensure that protected and backup tunnels are in Up, Up state.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels brief
```

Note Protected tunnels with 0 signaled bandwidth cannot be protected by limited backup-bw tunnels.

Step 6 Enable debugs, remove, and reapply backup tunnel-te.

```
RP/0/RSP0/CPU0:router# debug mpls traffic-eng frr
```

Step 7 Shut and no shut the backup tunnel and/or protected tunnel (if possible). This resets backup tunnel assignments.

Step 8 Ensure that fast-reroute option is not configured on the backup tunnel.

```
RP/0/RSP0/CPU0:router# show running-config interface tunnel-te15
```

Step 9 Find out if backup is assigned to a protected LSP.

a. `RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database`

b. `RP/0/RSP0/CPU0:router# show mpls traffic-eng forwarding`

c. `RP/0/RSP0/CPU0:router# show rsvp fast-reroute`

Step 10 Ensure that the pool-type of the protected LSP bandwidth and backup-bw of the backup tunnel matches.

MPLS FRR Switch Time Debugging

Step 1 Ensure that the FRR database is build and in ready state.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database
```

Step 2 Upon FRR triggered, ensure that FRR is in the active state.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database
```

Step 3 Check FRR switch time of LC that the primary tunnel is failed.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute log location node-id
```

Step 4 Ensure that both primary and backup tunnels on the LC received the FRR trigger.

```
RP/0/RSP0/CPU0:router# show cef platform trace te all location node-id
```

MPLS lDP/TE/FRR OOR Debugging

Step 1 Check label resource utilization.

```
RP/0/RSP0/CPU0:router# show controllers np struct 26 all location node-id
```

Step 2 Check next hop adjacency resource utilization.

```
RP/0/RSP0/CPU0:router# show controllers np struct 13 all location node-id
```

Step 3 Check transmit adjacency resource utilization.

```
RP/0/RSP0/CPU0:router# show controllers np struct 12 all location node-id
```

Multiple Spanning Tree (MST)

MST is an IEEE standard inspired from the Cisco proprietary Multiple Instances Spanning Tree Protocol (MISTP) implementation. This section contains the following:

- RL2GP Access Network Does Not Recognize RL2GP Node as Root, page 62
- Traffic Not Switching Through RL2GP Node(s), page 63

Using Show and Debug Commands

SUMMARY STEPS

1. **show spanning-tree mst**
2. **show spanning-tree ring-termination**
3. **show l2vpn bridge-domain [bd-name** *bridge-domain name* **| brief | detail | group** *bridge-domain group name* **| interface** {*type interface-id*} **| neighbor** *IP address* **[pw-id** *value***] | summary]**
4. **debug spanning-tree mst controller**
5. **debug spanning-tree mst io**
6. **debug spanning-tree mst packet**
7. **debug spanning-tree mst protocol-state**
8. **show spanning-tree mst id trace controller verbose**
9. **show spanning-tree mst id trace io verbose location** *location-of-problem-interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`show spanning-tree mst`	View MST status. Filter results using the following parameters and keywords: • **name**—Protocol instance name. • **blocked-ports**—MST information for blocked ports. • **bpdu interface** *interface-name*—MST Bridge protocol data units (BPDUs). • **brief** —MST information summary. • **configuration**—MST-related configuration information. • **errors**—MST configuration errors. • **instance-id**—MST instance. • **interface** *interface-name*—Per-interface MST information. • **internal**—MST internal information.
Step 2	`show spanning-tree ring-termination`	Verifies Spanning Tree Protocol (STP) is enabled. Filter results using the following parameters and keywords: • **name**—Protocol instance name. • **bpdu interface** *interface-name*—MST BPDUs. • **interface** *interface-name*.—Per-interface MST information

	Command or Action	Purpose
Step 3	**show l2vpn bridge-domain [bd-name** *bridge-domain name* | **brief** | **detail** | **group** *bridge-domain group name* | **interface** {*type interface-path-id*} | **neighbor** *IP address* **[pw-id** *value*] | **summary]** **Example:** `RP/0/RSP0/CPU0:router# show l2vpn bridge-domain`	View information for the bridge ports such as attachment circuits and pseudowires for the specific bridge domains. Filter results using the following parameters and keywords: • **bd-name**—Bridge domain name • **brief**—Brief information • **detail**—Detailed information • **group**—Bridge domain group name • **hardware**—Hardware information • **interface**—Filter on interface • **neighbor**—Filter on neighbor • **private**—Private information • **summary**—Bridge domain summary information
Step 4	**debug spanning-tree mst controller**	• **all**—Debugs all modules • **comms**—Comms module debugging • **config**—Configuration module debugging • **database**—Database module debugging • **edm**—EDM module debugging • **engine**—Engine module debugging • **l2vpn**— L2VPN module debugging • **pfi**— PFI module debugging • **rl2gp**—RL2GP module debugging
Step 5	**debug spanning-tree mst io**	• **all**—Debugs all modules • **comms**—Comms module debugging • **config**—Configuration module debugging • **database**—Database module debugging • **edm**— EDM module debugging • **engine**—Engine module debugging • **main**—Init/Cleanup debugging • **packet-io**—Packet input or output debugging • **pfi**—PFI module debugging • **rl2gp**— RL2GP module debugging
Step 6	**debug spanning-tree mst packet**	• **brief**—View brief output • **full**—View full output • **raw**—View raw packet output
Step 7	**debug spanning-tree mst protocol-state**	View debug for a specific MSTI interface.

	Command or Action	Purpose
Step 8	`show spanning-tree mst id trace controller verbose`	Use this trace as soon after the problem is observed as possible.
Step 9	`show spanning-tree mst id trace io verbose location [location of problem interface]`	Use this trace as soon after the problem is observed as possible.

MSTP Incorrectly or Inconsistently Formed

When the spanning tree is misformed, it is often due to misconfiguration or BPDU loss. This generally manifests as more than one node showing itself as ROOT, but can also result in disagreement on which nodes are ROOT.

MSTP Incorrectly or Inconsistently Formed—Misconfiguration

Ensure that the following match on the nodes:

- Configuration name
- Bridge revision
- Provider-bridge mode
- Instance to VLAN mapping

```
show run spanning-tree mst name
```

MSTP Incorrectly or Inconsistently Formed—BPDU Loss

Determine if node A is sending BPDUs to node B. Run the command several times for each interface connecting the nodes.

show spanning-tree mst *name* **internal io interfaces** *interface-name*

Only designated ports will send periodic BPDUs, but non-designated ports send updates on topology changes and startup. Ensure that BPDUs sent and received are going up as appropriate.

MSTP Correctly Formed, but Traffic Flooding

Intermittent BPDU loss may mean the Spanning-Tree will not show up incorrectly in the show commands, but will send out topology change notifications. These notifications cause a MAC flush, forcing traffic to flood until the MAC addresses are re-learned.

Look for topology change notifications. Run the following command and look for TC 1:

Note This option is verbose.

```
debug spanning-tree mst packet full {received | sent}
```

Run the command on both nodes with “brief” verbosity to check for missing BPDUs. Look at the timestamps, and look for gaps greater than or equal to six seconds. Gaps of that size would cause a topology change.

```
debug spanning-tree mst packet brief {received | sent}
```

MSTP Shows Wrong Port State

When the STP attempts to change the port state it uses L2VPN. Check the value of "Sent Update." If this is Yes, then STP is awaiting an update from L2VPN.

show spanning-tree mst name internal l2vpn

Packet Forwarding Does Not Match MSTP State

Shut down redundant links, remove MSTP configuration, and ensure that basic bridging works.

show spanning-tree mst *name*

show interface *interface-name*

Ensure that the state of each port as calculated by MSTP, and compare it with packet transmit and receive counts on ports and EFPs that are controlled by MSTP. Normal data packets should be sent/received only on ports that are in forwarding (FWD) state. BPDUs should be sent/received on all ports that are MSTP controlled. Ensure that BPDUs are flowing and that Root bridge selection is correct. See those related scenarios first.

show l2vpn bridge-domain [detail]

Will show the status of members of the bridge domain. Ensure that the relevant bridge domain members are up.

show uidb data location [location] *interface-name*

Check forwarding state as programmed in hardware.

RL2GP Access Network Does Not Recognize RL2GP Node as Root

show running-config spanning-tree ring-termination name

There are two ways of configuring RL2GP:

- Advertise as though both nodes are separate—requires each node have a unique bridge id and the configurations complement each other.
- Advertise as though each node is a different port on the same node—configuration is identical except for the port id.

Commands for RL2GP must target the untagged EFP instead of the base interface.

Step 1 Send the output on both RL2GP nodes.

```
RP/0/RSP0/CPU0:router# show span ring-termination test bpdu interface interface-name
```

Step 2 Debug on both nodes and include the output.

```
RP/0/RSP0/CPU0:router# debug spanning-tree mst packet full sent interface interface-name
```

Traffic Not Switching Through RL2GP Node(s)

Step 1 Collect l2vpn and UIDB data to ensure the datapath is healthy.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain [detail]
```

Step 2 Ensure that bridge domain members are up.

```
RP/0/RSP0/CPU0:router# show uidb data location [location] interface-name
```

Step 3 Ensure that the forwarding state is set as it was programmed in the hardware.

Quality of Service (QoS)

The system offers the following types of QoS:

- Multilevel priority scheduling for voice and video applications with minimal jitter, latency and packet loss.
- Priority propagation to ensure service integrity for voice and video throughout all hierarchy layers, even at peak hours with high traffic load.
- Differentiated Service Code Point (DSCP), MPLS experimental bit (EXP) and IEEE 802.1p classification with marking, policing and scheduling, ingress and egress.

This section contains the following:

Release 3.7.2 Final Draft?Cisco Confidential

Using Show and Debug Commands

SUMMARY STEPS

1. **show run policy-map**
2. **show run classmap**
3. **show run interface**
4. **show policy-map interface** *type instance* [**output** | **input**]
5. **show qos interface** *type instance* **[output | input]**
6. **show qos-ea interface** *type instance* **[output | input]**
7. **show qos-ea km**
8. **show qoshal entity**
9. **show qoshal queue**
10. **show qoshal [wfq | shape | wred | police | wred-scale] np tm level profile [sw] location**
11. **show qoshal tm-config**
12. **debug qos-ea ?**
13. **debug prm hal ?**
14. **show qos-ea trace**
15. **show qoshal trace**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show run policy-map** **Example:** `RP/0/RSP0/CPU0:router# show run policy-map l1-all`	View policymap with name.
Step 2	**show run classmap** **Example:** `RP/0/RSP0/CPU0:router# show run class-map c2`	View class-map configuration with name.
Step 3	**show run interface** **Example:** `RP/0/RSP0/CPU0:router# show run interface g0/2/0/0`	View the service-policy binding for a given port/subinterface.
Step 4	**show policy-map interface g0/2/0/0 [output|input]** **Example:** `RP/0/RSP0/CPU0:router# show policy-map inter g0/2/0/0`	View all the statistics, queue IDs and class information.

	Command or Action	Purpose
Step 5	**show qos interface** *type instance* **[output|input]** **Example:** `RP/0/RSP0/CPU0:router# show qos int g0/2/0/0 out`	View all the configuration of each class in hardware.
Step 6	**show qos-ea interface** *type instance* **[output|input]** **Example:** `RP/0/RSP0/CPU0:router# show qos-ea int g0/2/0/0 out`	View all the class information structures.
Step 7	**show qos-ea km** **Example:** `RP/0/RSP0/CPU0:router# show qos-ea km policy l2-all vmr interface g0/2/0/0 sw`	View the key manager (TCAM key manager) related fields associated to a policy-map/interface binding.
Step 8	**show qoshal entity** **Example:** `RP/0/RSP0/CPU0:router# show qoshal entity np 0 tm 1 level 4 ?`	View all the hardware configuration of the TM entity.
Step 9	**show qoshal queue** **Example:** `RP/0/RSP0/CPU0:router# show qoshal queue np 0 tm 1 qid 1286 8 loca 0/2/CPU0`	View the current number of packets in the queue.
Step 10	**show qoshal [wfq|shape|wred|police|wred-scale] np tm level profile [sw] location** **Example:** `RP/0/RSP0/CPU0:router# show qoshal wfq np 0 tm 1 le 4 pr 4 2 loca 0/2/CPU0`	View hardware profiles various queueing/non-queuing features.
Step 11	**show qoshal tm-config** **Example:** `RP/0/RSP0/CPU0:router# show qoshal tm-config all np 0 tm 1 l 0/2/CPU0`	View global tm config information.
Step 12	**debug qos-ea ?** **Example:** `RP/0/RSP0/CPU0:router# debug qos-ea ?`	—
Step 13	**debug prm hal ?** **Example:** `RP/0/RSP0/CPU0:router# debug prm hal ?`	Debugs HAL wrapper module errors.

	Command or Action	Purpose
Step 14	`show qos-ea trace`	View all events/errors in the QOS EA component.
Step 15	`show qoshal trace`	View all events in qoshal.

Service-policy Configuration Is Rejected

Step 1 Verify if the failure is caused by resource allocation.

```
RP/0/RSP0/CPU0:router# show qoshal resource summary np np location filename
```

Step 2 If resource usage is more than what is configured, verify how many are checkpointed.

a. `RP/0/RSP0/CPU0:router# show qos-ea ha chkpt all info location location`

b. `RP/0/RSP0/CPU0:router# show qoshal ha chkpt all info location location`

Step 3 Check the OOR.

Step 4 Verify resources used.

Step 5 Verify trace and summary information.

a. `RP/0/RSP0/CPU0:router# show qos-ea trace errors location filename`

b. `RP/0/RSP0/CPU0:router# show qos-ea trace events location filename`

c. `RP/0/RSP0/CPU0:router# show qos-ea trace internal location filename`

d. `RP/0/RSP0/CPU0:router# show qoshal trace all location filename`

e. `RP/0/RSP0/CPU0:router# show qos summary {queue | police | policy}`

Step 6 Collect trace information and resource summary.

a. `RP/0/RSP0/CPU0:router# show qoshal resource summary location filename`

b. `RP/0/RSP0/CPU0:router# show qos-ea trace errors location filename`

c. `RP/0/RSP0/CPU0:router# show qos-ea trace events location filename`

d. `RP/0/RSP0/CPU0:router# show qos-ea trace internal location filename`

e. `RP/0/RSP0/CPU0:router# show qoshal trace all location filename`

Packets are Incorrectly Classified

Step 1 Verify packets are arriving on the correct interface.

Step 2 Verify the packet fields are as expected.

Step 3 Note the packet type.

Step 4 Verify the KM policy information matches UIDB configuration.

```
RP/0/RSP0/CPU0:router# show qos-ea km policy policy info location filename
```

```
RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename hw detail
```

Step 5 Verify UIDB has qos enabled. The format ID and qos ID should match what is in the **show qos-ea km policy info location** command.

```
RP/0/RSP0/CPU0:router# show uidb data shadow location filename interface filename
{ingress|egress}
```

Step 6 Verify VMR entries for each class.

```
RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename hw detail
```

Step 7 Verify which class the packets are actually matching. If packet fields should match different class, then NP Microcode needs to debug this further.

```
RP/0/RSP0/CPU0:router# show policy-map interface filename {output | input} [member
filename]
```

Step 8 Verify if in ingress QOS lookup occurs before L2 ingress rewrite and that in egress L2 rewrite occurs before QOS lookup.

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Step 9 `RP/0/RSP0/CPU0:router# show run interface filename`

Step 10 `RP/0/RSP0/CPU0:router# show run policy-map policy`

Step 11 `RP/0/RSP0/CPU0:router# show run class-map classmap`

Packets in Wrong Queue

Step 1 `RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename] detail`

Step 2 Verify the packets are correctly classified.

Step 3 Verify hash structure.

```
RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename]
detail
```

Step 4 Verify the hash key for the class and hash result of the class has correct Queue ID.

Packets Incorrectly Marked

Step 1 Verify packets are classified correctly.

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Step 2 `RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename hw`

Step 3 `RP/0/RSP0/CPU0:router# show qos-ea km policy policy vmr interface filename sw`

Step 4 `RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member filename]`

Step 5 Verify marking value.

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Packets Incorrectly Policed

Step 1 Ensure that packets are correctly classified.

Step 2 Verify whether policer CIR/CBS/PIR/PBS are set correctly as per configured service-policy. Also verify the rate at which traffic is coming to match against policed rate.

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Step 3 Get the token bucket and police node index of the class.

```
RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename]
detail
```

Step 4 Verify the policer profile of this token bucket is same as what is shown in show qos interface output. np and index values can be get by 'show qos-ea command' for that class.

```
RP/0/RSP0/CPU0:router# show qoshal police-node np np index index count location filename
```

Step 5

```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

Shaping Incorrect

Step 1 Ensure that packets are correctly classified.

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Step 2 Verify whether shaper CIR/CBS/PIR/PBS are set correctly as per configured service-policy. Get the shape profile ID and entity handle information (np, tm, level, index, offset).

```
RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename]
detail
```

Step 3 Verify the shaper profile in hardware if they are correctly configured.

Step 4 Verify the entity hierarchy in the TM to check whether the TM hierarchy is configured correctly.

```
RP/0/RSP0/CPU0:router# show qoshal shape np np tm tm level level profile id count location
filename
```

Step 5

```
RP/0/RSP0/CPU0:router# show qoshal entity np np tm tm level level index id offset hierarchy
location filename
```

Step 6

```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

Weighted Random Early Detection (WRED) Incorrect

Step 1 Ensure that packets are correctly classified.

Step 2 Verify whether WRED curves are correctly configured with minimum and maximum thresholds of each curve are as per configured service-policy.

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Step 3 Get the WRED profile ID and entity handle information (np, tm, level, index, offset).

```
RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename]
detail
```

Step 4 Ensure that the WRED profiles in hardware are correctly configured.

```
RP/0/RSP0/CPU0:router# show qoshal wred np np tm tm level level profile id count location
filename
```

Step 5 Verify the entity hierarchy in the TM to check whether the TM hierarchy is configured correctly.

```
RP/0/RSP0/CPU0:router# show qoshal entity np np tm tm level level index id offset hierarchy
location filename
```

Step 6
```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

Bandwidth Not Guaranteed

Step 1 Ensure that packets are correctly classified.

Step 2 Verify whether the weights of each class are configured correctly as per the bandwidth ratio among classes.

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Step 3
```
RP/0/RSP0/CPU0:router# show run policy-map policy
```

Step 4 If its correctly configured, then get the WFQ profile ID and entity handle information (np, tm, level, index, offset) of the class.

```
RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename]
detail
```

Step 5 Ensure that the TM hierarchy is configured correctly using the **show qoshal entity np tm level index hierarchy location** command.

Step 6 Verify the parent and child entities are configured correctly, both shaping and WFQ weights.

```
RP/0/RSP0/CPU0:router# show qoshal wred np np tm tm level level profile id count location
filename
```

Step 7
```
RP/0/RSP0/CPU0:router# show qoshal entity np np tm tm level level index id offset hierarchy
location filename
```

Step 8
```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

Step 9 Ensure that the WFQ profiles in hardware are correctly configured.

```
RP/0/RSP0/CPU0:router# show qoshal wred np np tm tm level level profile index count
location filename
```

Bandwidth Ratio Not Working

Step 1 Ensure that packets are correctly classified.

Step 2
```
RP/0/RSP0/CPU0:router# show run policy-map policy
```

Step 3 Verify whether the commit weights of each class is configured correctly as per the bandwidth ratio among classes. Also verify excess weights are configured as per bandwidth remaining ratio configuration (the ration of excess weights should be in the ratio of excess weights).

```
RP/0/RSP0/CPU0:router# show qos interface filename {input | output} [member filename]
```

Step 4
```
RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename]
detail
```

Step 5
```
RP/0/RSP0/CPU0:router# show qoshal wred np np tm tm level level profile id count location
filename
```

Step 6 Ensure that the TM hierarchy is configured correctly.

```
RP/0/RSP0/CPU0:router# show qoshal entity np np tm tm level level index id offset hierarchy
location filename
```

Step 7
```
RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member
filename]
```

Step 8 Get the WFQ profile ID and entity handle information (np, tm, level, index, offset) of the class.

```
RP/0/RSP0/CPU0:router# show qos-ea interface filename {input | output} [member filename]
```

Step 9 Verify the WFQ profile in hardware is correctly configured.

```
RP/0/RSP0/CPU0:router# show qoshal wred np np tm tm level level profile index count
location filename
```

Step 10 If commit and excess weights are correct:

a. Check queue size of each class.

b. Increase the queue size.

Unable to Modify or Delete policy-map or class-map

Step 1 Verify the policy is applied on an interface.

```
RP/0/RSP0/CPU0:router# show running-config
```

Step 2 Remove service-policy on the interfaces.

Step 3 Modify the policy-map.

Unable to Modify or Delete class-map ACL

- **show config failed**
- **show running-config**

Step 1 Verify theaACL is part of a match statement in a class-map.

Step 2 Verify the class-map is part of any policy-map that is applied on an interface.

Step 3 If the policy-map is applied on interface, ACL modification/deletion is not allowed.

Step 4 Remove all the service-policy configuration of this policy-map and modify ACLs

Unable to Delete service-policy

Step 1 `RP/0/RSP0/CPU0:router# show config failed`

Step 2 `RP/0/RSP0/CPU0:router# show qos-ea trace all location filename`

Step 3 Restart the qos_ma_ea process.

After QoS EA Restarts, show policy-map interface Fails

- **show running-config**
- **show qos-ea trace all location filename**
- **show qos-ea ha chkpt all info location filename**
- **show qos-ea ha chkpt if-qos all location filename**

Step 1 Verify if the state of QoS EA is in in_sync (state = 2).

```
RP/0/RSP0/CPU0:router# show qos-ea ha state location filename
```

Step 2 Verify the trace of QoS EA to see if the service-policy is restored for the interface.

Step 3 If there is no error, do the following:

a. `RP/0/RSP0/CPU0:router# debug qos-ea all`

b. `RP/0/RSP0/CPU0:router# debug generic`

c. Collect debugs by performing the failing command.

After QoS EA Restarts, service-policy config Fails

Step 1 Verify the state of QoS EA is in in_sync (state = 2).

```
RP/0/RSP0/CPU0:router# show qos-ea ha state location filename
```

Release 3.7.2 Final Draft?Cisco Confidential

Step 2 Verify the trace of QoS EA to see if the service-policy is restored.

Step 3 If there is no error, do the following:

a. `RP/0/RSP0/CPU0:router# debug qos-ea all`

b. `RP/0/RSP0/CPU0:router# debug generic`

c. Collect the debugs by issuing the failing command.

show policy-map interface Output Error

For bundles, specify member interface. Policy information for bundle-interface is not available in the current release.

- **show policy-map interface {output | input} member**
- **show {qos|qos-ea} interface {output | input} member**

Bundle Members Not Configured with service-policy

For bundles, specify member interface. Policy information for bundle-interface is not available in the current release.

- **show policy-map interface {output | input} member**
- **show {qo s| qos-ea} interface {output | input} member**

Reverse Path Forwarding (RPF)

RPF ensures loop-free forwarding of multicast packets in multicast routing. This section contains the following:

Using Show and Debug Commands

show cef ipv4 interface—View IPv4 Cisco Express Forwarding (CEF)-related information for an interface

Packets from Wrong IP Address—Loose RPF

In loose RPF, the packets incoming on that particular interface will be checked whether the source IP of the packet is reachable through some interface on the box. If not, the packet should be dropped.

Step 1 Ensure that loose RPF is configured on the interface by checking the RPF flags.

`RP/0/RSP0/CPU0:router# show uidb data location node-id filename ingress`

Step 2 Ensure that that the system returns RPF list: uidb1: 12 uidb2: 0 uidb3: 0 uidb4: 0. The RPF list should have at least one UIDB which is a non-zero index. If all are zeros, then it means the loose RPF is not properly set inside the hardware. If you see all zeros, then it is possible that the loose RPF config is appended with 'allow-default' option. This means the RPF check will pass even if there is a default route configured in the system.

```
RP/0/RSP0/CPU0:router# show cef {ipv4} prefix hardware egress detail location node-id
```

Step 3 Check ucode statistics to ensure STATS_STATIC_IPV4_URPF_DROP_PKT is incremented.

```
RP/0/RSP0/CPU0:router# show controller NP counter {np0 | np1 | np2 | np3} location node-id
```

Workaround

Unconfigure and configure loose RPF on the interface.

```
ipv4 verify unicast source reachable-via any
```

Packets Forwarded with Wrong IP Address—Strict RPF

In strict RPF, the packets incoming on that particular interface will be checked whether the source IP of the packet is reachable through the same interface on the box on which the packet came in. If not, the packet should be dropped.

Step 1 Ensure that strict RPF is configured on the interface by checking the RPF flags.

```
RP/0/RSP0/CPU0:router# show uidb data location node-id filename ingress
```

Step 2 Ensure that the system returns RPF list: uidb1: 12 uidb2: 0 uidb3: 0 uidb4: 0v. The RPF list should have at least one UIDB which is a non-zero index. And the non-zero index should be the same index corresponding to the ingress interface of the packet.

```
RP/0/RSP0/CPU0:router# show cef {ipv4} <prefix> hardware egress detail location node-id
```

Step 3 Check for the ingress index. If the RPF list has all zero indexes, it means strict RPF is not properly set inside the hardware. If you see all zeros, the strict RPF config is appended with 'allow-default' option. This means the RPF check will pass even if there is a default route configured in the system.

```
RP/0/RSP0/CPU0:router# show uidb index location node-id
```

Step 4 Check ucode statistics to ensure STATS_STATIC_IPV4_URPF_DROP_PKT is incrementing.

```
RP/0/RSP0/CPU0:router# show controller NP counter {np0 | np1 | np2 | np3} location node-id
```

Workaround

Unconfigure and configure strict rpf on the interface.

```
ipv4 verify unicast source reachable-via rx
```

Release 3.7.2 Final Draft?Cisco Confidential

Virtual Private Local Area Network Service (VPLS)

VPLS allows geographically separate sites to share an Ethernet broadcast domain by connecting them using pseudowires. This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show l2vpn bridge-domain [bd-name** *bridge-domain name* **| brief | detail | group** *bridge-domain group name* **| interface {***type interface-id***} | neighbor** *IP address* **[pw-id** *value***] | summary]**
2. **show l2vpn forwarding bridge-domain [***bridge-domain-name***] {detail | hardware {egress | ingress}} {location** *node-id***}**
3. **debug l2vpn forwarding platform vpls all location** *node-id*
4. **show l2vpn platform trace vpls all location** *node-id*
5. **show controller np counters all location** *node-id*
6. **show controller np struct 17 det all all**
7. **show controller np struct 15 det all all**
8. **show controller np struct 28 det all all**
9. **show controller np struct 16 det all all**
10. **show controller np struct 18 det all all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show l2vpn bridge-domain [bd-name bridge-domain name | brief | detail | group bridge-domain group name | interface {type interface-id} | neighbor IP address [pw-id value] | summary]** **Example:** `RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name d1`	View the bridge-domain bridge-ports (ACs and PW) . Filter results using the following parameters and keywords: • **bd-name** *bridge-domain name*—(Optional) Displays the bridges by the bridge ID. The bridge-domain name argument is used to name a bridge domain. • **brief**—(Optional) Displays brief information about the bridges. • **detail**—(Optional) Displays the output for the Layer 2 VPN (L2VPN) to indicate whether or not the MAC withdrawal feature is enabled and the number of MAC withdrawal messages that are sent or received from the pseudowire. • **group** *bridge-domain group name*—(Optional) Displays filter information on the bridge-domain group name. The bridge-domain group name argument is used to name the bridge domain group. • **interface**—(Optional) Displays the filter information for the interface on the bridge domain. • *type*—Interface type. • *interface-id*—Identifies a physical interface or a virtual interface. • **neighbor** *IP address*—(Optional) Displays only the bridge domain that contains the pseudowires to match the filter for the neighbor. The IP address argument is used to configure IP address of the neighbor. • **pw-id** *value*—(Optional) Displays the filter for the pseudowire ID. The range is from 1 to 4294967295. • **summary**—(Optional) Displays the summary information for the bridge domain.
Step 2	**show l2vpn forwarding bridge-domain** **Example:** `RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain ABC mac-address interface Gi0/1/2/1.2 detail hardware location node-id Bridge`	View forwarding bridge domain information. Filter results using the following parameters and keywords: • *bridge-domain-name*—(Optional) Name of a bridge domain. • **detail**—Displays all the detailed information on the attachment circuits and pseudowires. • **hardware**—Displays the hardware location entry. • **egress**—Reads information from the egress PSE. • **ingress**—Reads information from the ingress PSE. • **location** *node-id*—Displays the bridge-domain information for the specified location. The node-id argument is entered in the rack/slot/module notation.

	Command or Action	Purpose
Step 3	**debug l2vpn forwarding platform vpls all location node-id** **Example:** RP/0/RSP0/CPU0:router# debug l2vpn forwarding platform common ?	—
Step 4	**show l2vpn platform trace vpls all location node-id** **Example:** RP/0/RSP0/CPU0:router# show l2vpn platform trace common ?	—
Step 5	**show controller np counters all location node-id** **Example:** RP/0/RSP0/CPU0:router# show controller np counter np0 location node-id	—
Step 6	**show controller np struct 17 det all all** RP/0/RSP0/CPU0:router# show controller np struct 17 det all all	—
Step 7	**show controller np struct 15 det all all** **Example:** RP/0/RSP0/CPU0:router# show controller np struct 15 det all all	—
Step 8	**show controller np struct 28 det all all** **Example:** RP/0/RSP0/CPU0:router# show controller np struct 15 det all all	—
Step 9	**show controller np struct 16 det all all** **Example:** RP/0/RSP0/CPU0:router# show controller np struct 16 det all all	—
Step 10	**show controller np struct 18 det all all** **Example:** RP/0/RSP0/CPU0:router# show controller np struct 18 det all all	—

AC Is Down

Step 1 RP/0/RSP0/CPU0:router# **show interface**

Step 2 RP/0/RSP0/CPU0:router# **show l2vpn bridge interface detail**

Step 3 Ensure that the AC interface has l2transport configured.

Step 4 Ensure that the AC interface is up.

Step 5 Ensure that the MTUs match.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain interface type instance detail
```

Pseudowire Is Down

Step 1 View OSPF neighbor information.

```
RP/0/RSP0/CPU0:router# show ospf neighbor
```

Step 2 View MPLS LDP neighbor information.

```
RP/0/RSP0/CPU0:router# show mpls ldp neighbor neighbor
```

Step 3 View the bridge pseudowire state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain neighbor
```

Step 4 Ensure that pseudowires are properly configured on both PEs.

Step 5 Ensure that the MPLS package is installed.

Step 6 Ensure that the core interface is up.

Step 7 Ensure that OSPF is the routing protocol.

Step 8 Ensure that an LDP session is established with the PE peer.

Step 9 Ensure that the MTUs match.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

VPLS Not Forwarding Flooding Traffic

Step 1 View OSPF neighbor information.

```
RP/0/RSP0/CPU0:router# show ospf neighbor
```

Step 2 View MPLS LDP neighbor information.

```
RP/0/RSP0/CPU0:router# show mpls ldp neighbor neighbor
```

Step 3 Ensure that pseudowires are properly configured on both PEs.

Step 4 Ensure that the MPLS package is installed.

Step 5 Ensure that the core interface is up.

Step 6 Ensure that OSPF is the routing protocol.

Step 7 Ensure that an LDP session is established with the PE peer.

Step 8 Ensure that the MTUs match.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

VPLS Not Forwarding Flooding Traffic from AC to Pseudowire

Step 1 View ingress UIDB and XID for the segment.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface hardware ingress detail location
```

Step 2 View pseudowire hardware information.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding neighbor 192.12.12.5 pw-id 100 hardware
egress location node-id0
```

Step 3 View MPLS leaf information.

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels hardware egress detail location
```

Step 4 View bridge information about Broadcast, Multicast and Unknown Unicast.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name 1 det
```

Step 5 Ensure that the MAC limit has not been exceeded.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain 1:1 detail location
```

Step 6 View platform error traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location
```

Step 7 View platform event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls event location
```

Step 8 View PI event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

Step 9 Ensure that the pseudowires and AC are up.

Step 10 Verify the hardware is programmed for both ACs.

Step 11 If the AC is not correctly programmed, check for XID programming errors.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location node-id
```

Step 12
```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEtherne0/5/0/2 hardware
ingress detail location node-id
```

Step 13 Verify the hardware is programmed for pseudowires.

VPLS Not Forwarding Flooding Traffic from Pseudowire to AC

Step 1 View ingress UIDB and XID for the segment.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface hardware ingress detail location
```

Step 2 View MPLS leaf information.

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels hardware egress detail location
```

Step 3 View bridge information about Broadcast, Multicast and Unknown Unicast.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name 1 det
```

Step 4 Ensure that the MAC limit has not been exceeded.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain 1:1 detail location
```

Step 5 View platform error traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location
```

Step 6 View platform event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls event location
```

Step 7 View PI event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

Step 8 Ensure that the pseudowires and AC are up.

Step 9 Verify the hardware is programmed for both ACs.

Step 10 If the AC is not correctly programmed, check for XID programming errors.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location node-id
```

Step 11
```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEtherne0/5/0/2 hardware
ingress detail location node-id
```

Step 12 Verify the hardware is programmed for pseudowires.

VPLS Not Forwarding Unicast Traffic from AC to AC

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 View MAC information.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location
```

Step 3 View MAC programmed in hardware.

```
RP/0/RSP0/CPU0:router# show controllers np struct 18 search key all location
```

Step 4 Ensure that the hardware is programmed for both ACs.

Step 5 Ensure that the destination MAC entry is programmed for the LC's destination interface.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location node-id0
```

Step 6 Ensure that the MAC is programmed on the NP.

```
RP/0/RSP0/CPU0:router# show controllers np struct 18 search key 000002000100 all location
node-id0
```

VPLS Not Forwarding Unicast Traffic from AC to Pseudowire

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 View MAC information.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location
```

Step 3 View MAC programmed in hardware.

```
RP/0/RSP0/CPU0:router# show controllers np struct 18 search key all location
```

Step 4 Ensure that the hardware is programmed for both AC and pseudowire.

Step 5 Ensure that the destination MAC entry is programmed for the LC's destination interface.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain mac-address location node-id0
```

Step 6 Ensure that the MAC is programmed on the NP.

```
RP/0/RSP0/CPU0:router# show controllers np struct 18 search key 000002000100 all location
node-id0
```

VPLS Not Forwarding Flooding Traffic from Pseudowire to AC

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 Ensure that the hardware is programmed for both AC and pseudowire.

Pseudowire Up but Ping Fails

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 Ensure that both CEs are on the same subnet.

Step 3 Ensure that the MTUs match.

Step 4 Ensure that the end-to-end encapsulations match.

Traffic Loss

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 View NP counters.

```
RP/0/RSP0/CPU0:router# show controllers NP counter {np0 | np1 | np2 | np3} location node-id
```

Step 3 View segment counters to see if the packet and byte switched count increased.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet? node-id detail
location node-id
```

Step 4 Ensure that the bandwidth rates match between the CEs.

Pseudowire Flap Causing Traffic Loss

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 View NP counters.

```
RP/0/RSP0/CPU0:router# show controllers NP counter {np0 | np1 | np2 | np3} location node-id
```

Step 3 View segment counters to see if the packet and byte switched count increased.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet? node-id detail
location node-id
```

Step 4 View platform error traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location
```

Step 5 View platform event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls event location
```

Step 6 View PI event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

Release 3.7.2 Final Draft?Cisco Confidential

Step 7 View which queues control and data packets are going into.

```
RP/0/RSP0/CPU0:router# show qoshal default-queue port 19 location node-id0
```

Step 8 Check to see if pseudowire is going down due to an UNBIND from PI.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls all and show l2vpn trace
```

Traffic Loss During RSP Fail Over

Step 1 View the state of the xconnect.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge detail
```

Step 2 View the various counters in the respective NPs.

```
RP/0/RSP0/CPU0:router# show controllers NP counter {np0 | np1 | np2 | np3} location node-id
```

Step 3 View counter for the segment.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernetnode-id? detail
location node-id
```

Step 4 View vpls pd traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls all location node-id
```

Step 5 View l2vpn PI traces.

```
RP/0/RSP0/CPU0:router# show l2vpn trace location node-id
```

Step 6 View the state of the bridge domain.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name detail
```

Step 7 View the various counters in the respective NPs.

```
RP/0/RSP0/CPU0:router# show controllers NP counter {np0 | np1 | np2 | np3} location node-id
```

Step 8 View ingress UIDB.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface interface hardware ingress detail
location node-id
```

Step 9 Display l2vpn_ldi structure.

```
RP/0/RSP0/CPU0:router# show controllers np struct 16 detail all all location
```

Step 10 Display nr-ldi structure.

```
RP/0/RSP0/CPU0:router# show controllers np struct 7 detail all all location
```

Step 11 Check all routers in the MPLS path to ensure the following are configured:

a. MPLS LDP graceful restart

b. OSPF NSF

Step 12 Look for an UNBIND event.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls all location node-id
```

Step 13
```
RP/0/RSP0/CPU0:router# show tech l2vpn
```

Step 14 View segment counters to see if the packet and byte switched count increased.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet? node-id detail
location node-id
```

Step 15 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller NP counters {NP0 | NP1 | NP2 | NP3} location node-id
```

Preferred Path Not Working

Step 1 View the state of the bridge domain.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name detail
```

Step 2 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller NP counters {NP0 | NP1 | NP2 | NP3} location node-id
```

Step 3 View ingress UIDB.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface interface hardware ingress detail
location node-id
```

Step 4 View l2vpn_ldi structure.

```
RP/0/RSP0/CPU0:router# show controllers np struct 16 detail all all location
```

Step 5 View nr-ldi structure.

```
RP/0/RSP0/CPU0:router# show controllers np struct 7 detail all all location
```

Virtual Private Wire Services (VPWS)

VPWS connects geographically separate sites by emulating a set of wires between them using the underlying MPLS tunnel. This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show l2vpn xconnect [detail | group | interface | neighbor | state | summary | type | state unresolved]**
2. **show l2vpn forwarding {detail | hardware | interface | location | message | resource | summary | unresolved} location** *node-id*
3. **show MPLS forwarding [detail | {label** *label number*} **| interface** *interface-id* **| labels value | location | prefix [***network/mask* | *length*] **| private | summary | tunnels** *tunnel-id*]
4. **debug l2vpn forwarding platform atom {all | error | events| updates} location** *node-id*
5. **debug l2vpn forwarding platform common {all | error | events| updates} location** *node-id*
6. **show l2vpn platform atom {all | error | events| updates} location** *node-id*
7. **show l2vpn platform common {all | error | events| updates} location** *node-id*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`show l2vpn xconnect [detail | group | interface | neighbor | state | summary | type | state unresolved]`
	
	Example:
	`RP/0/RSP0/CPU0:router# show l2vpn xconnect`
		View brief information on configured cross-connects. Filter results using the following parameters and keywords:
		• **detail**—Detailed information
		• **group**—All cross-connects in a specified group
		• **interface**—Interface and subinterface
		• **neighbor**—Neighbor
		• **state**—Xconnect state types: up, down
		• **summary**—AC information from the AC Manager database
		• **type**—Xconnect types: ac-pw, locally switched
		• **state unresolved**—Unresolved cross-connects

Step 2	`show l2vpn forwarding {detail | hardware | interface | location | message | resource | summary | unresolved} location node-id`
	
	Example:
	`RP/0/RSP0/CPU0:router# show l2vpn forwarding location 0/2/cpu0`
		View the matching AC subinterface.

	Command or Action	Purpose
Step 3	**show mpls forwarding [detail | {label label number} | interface interface-id | labels value | location | prefix [network/mask | length] | private | summary | tunnels tunnel-id]**	View the MPLS Label Forwarding Information Base (LFIB) entries with a local labels range.
Step 4	**debug l2vpn forwarding platform atom {all | error | events| updates} location node-id** RP/0/RSP0/CPU0:router# debug l2vpn forwarding platform atom ?	—
Step 5	**debug l2vpn forwarding platform common {all | error | events| updates} location node-id** **Example:** RP/0/RSP0/CPU0:router# debug l2vpn forwarding platform common ?	—
Step 6	**show l2vpn platform atom {all | error | events| updates} location node-id** **show l2vpn platform common {all | error | events| updates} location node-id**	—
Step 7	**show l2vpn platform common {all | error | events| updates} location node-id**	—

AC Is Down

Step 1 RP/0/RSP0/CPU0:router# **show interface**

Step 2 RP/0/RSP0/CPU0:router# **show l2vpn bridge interface detail**

Step 3 Ensure that the AC interface has l2transport configured.

Step 4 Ensure that the AC interface is up.

Step 5 Ensure that the MTUs match.

RP/0/RSP0/CPU0:router# **show l2vpn bridge-domain interface** *type instance* **detail**

Pseudowire Is Down

Step 1 View OSPF neighbor information.

RP/0/RSP0/CPU0:router# **show ospf neighbor**

Step 2 View MPLS LDP neighbor information.

RP/0/RSP0/CPU0:router# **show mpls ldp neighbor** *neighbor*

Step 3 View the bridge pseudowire state.

RP/0/RSP0/CPU0:router# **show l2vpn bridge-domain neighbor**

Step 4 Ensure that pseudowires are properly configured on both PEs.

Step 5 Ensure that the MPLS package is installed.

Step 6 Ensure that the core interface is up.

Step 7 Ensure that OSPF is the routing protocol.

Step 8 Ensure that an LDP session is established with the PE peer.

Step 9 Ensure that the MTUs match.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail
```

VPWS Not Forwarding Traffic from AC to Pseudowire

Step 1 View ingress UIDB and XID for the segment.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface hardware ingress detail location
```

Step 2 View pseudowire hardware information.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding neighbor 192.12.12.5 pw-id 100 hardware
egress location node-id0
```

Step 3 View MPLS leaf information.

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels hardware egress detail location
```

Step 4 View bridge information about Broadcast, Multicast and Unknown Unicast.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name 1 det
```

Step 5 Ensure that the MAC limit has not been exceeded.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain 1:1 detail location
```

Step 6 View platform error traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location
```

Step 7 View platform event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls event location
```

Step 8 View PI event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

Step 9 Ensure that the pseudowires and AC are up.

Step 10 Verify the hardware is programmed for both ACs.

Step 11 If the AC is not correctly programmed, check for XID programming errors.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location node-id
```

Step 12 `RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEtherne0/5/0/2 hardware ingress detail location` *node-id*

Step 13 Verify the hardware is programmed for pseudowires.

VPWS Not Forwarding Traffic from Pseudowire to AC

Step 1 View ingress UIDB.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface interface hardware ingress detail
location node-id
```

Step 2 View MPLS leaf information.

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels hardware egress detail location
```

Step 3 View platform error traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls error location
```

Step 4 View platform event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn platform trace vpls event location
```

Step 5 View PI event traces.

```
RP/0/RSP0/CPU0:router# show l2vpn trace location
```

Step 6 Ensure that the pseudowires and AC are up.

Step 7 Find the local label associated with this pseudowire.

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect detail
```

Step 8 Find the XID associated with the AC.

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect detail
```

Step 9 View MPLS leaf information.

```
RP/0/RSP0/CPU0:router# show mpls forwarding labels hardware egress detail location
```

Step 10 `RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEtherne0/5/0/2 hardware ingress detail location node-id`

Pseudowire Up but Ping Fails

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 Ensure that both CEs are on the same subnet.

Step 3 Ensure that the MTUs match.

Step 4 Ensure that the end-to-end encapsulations match.

Traffic Loss

Step 1 View the bridge domain state.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name node-id detail
```

Step 2 View NP counters.

```
RP/0/RSP0/CPU0:router# show controllers NP counter {np0 | np1 | np2 | np3} location node-id
```

Step 3 View segment counters to see if the packet and byte switched count increased.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface GigabitEthernet? node-id detail
location node-id
```

Step 4 Ensure that the bandwidth rates match between the CEs.

Traffic Loss During RSP Fail Over

When RSP fail over is performed, some times it is seen that the traffic loss is experienced. This may be because the IGP over which the prefixes are learned is going down. The following assumes OSPF as the IGP.

- **show ospf** *process-id* **trace**—View OSPF major NSF related traces during failover
- **show process failover**—View process details during failover
- **debug ospf ha**—Enables OSPF HA related debugs
- **debug ospf** *instance* **nsf**—Before FO (Fail Over) and collect the debug log
- **show process failover**—After FO
- **show ospf trace ha**—After FO

Step 1 One thing to check immediately is if the next hop router also experienced an FO mechanism (Similar to what is done on this router). If so, the OSPF may go down.

Step 2 If not, verify that ‘nsf cisco’ is configured under the OSPF. If ‘nsf cisco’ is configured, see if the next hop is reachable during FO. If not, there may be a reachability issue like a link going down or negotiation problems.

Preferred Path Not Working

Step 1 View the state of the bridge domain.

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name detail
```

Step 2 View NP counters.

```
RP/0/RSP0/CPU0:router# show controller NP counters {NP0 | NP1 | NP2 | NP3} location node-id
```

Step 3 View ingress UIDB.

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding interface interface hardware ingress detail
location node-id
```

Step 4 View l2vpn_ldi structure.

```
RP/0/RSP0/CPU0:router# show controllers np struct 16 detail all all location
```

Step 5 View nr-ldi structure.

```
RP/0/RSP0/CPU0:router# show controllers np struct 7 detail all all location
```

Virtual Router Redundancy Protocol (VRRP)

VRRP enables a group of routers to form a single virtual router. This section contains the following:

Using Show and Debug Commands

SUMMARY STEPS

1. **show vrrp** [*interface* [*vrid*]] **[brief]**
2. **show vrrp** [*interface* [*vrid*]] **detail**
3. **show vrrp** [*interface* [*vrid*]] **statistics [all]**
4. **show controllers** *interface*
5. **debug vrrp**
6. **debug ether-ctrl config**

Release 3.7.2 Final Draft?Cisco Confidential

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show vrrp** [interface filename [vrid]] [brief] **Example:** RP/0/0/CPU0:# show vrrp	View all VRRP groups status.
Step 2	**show vrrp** [interface filename [vrid]] detail **Example:** RP/0/0/CPU0:# show vrrp detail	View detailed information of VRRP groups.
Step 3	**show vrrp** [interface filename [vrid]] statistics [all] **Example:** RP/0/0/CPU0:# show vrrp statistics	View VRRP statistics.
Step 4	**show controllers** filename **Example:** RP/0/0/CPU0:# show controllers gigabitEthernet 0/3/0/9	View the VRRP group MAC addresses as part of unicast filter list.
Step 5	**debug vrrp**	—
Step 6	**debug ether-ctrl config**	—

VRRP Fails to Reach Active State

On both routers:

Step 1 RP/0/RSP0/CPU0:router# **show vrrp detail**

Step 2 RP/0/RSP0/CPU0:router# **show vrrp trace**

Step 3 RP/0/RSP0/CPU0:router# **show eth-output trace location** *interface-running-vrrp*

Misconfiguration

Step 1 Ensure that the interface with VRRP configured is up.

Step 2 Ensure that an IP address is configured, on the same subnet as the interface, and delay is configured.

RP/0/RSP0/CPU0:router# **show vrrp detail**

Higher Priority Router Already Active

Examine the output of the **show vrrp** command:

- If the Master address for VRRP shows an IP address instead of local, the router with that IP address is Active.
- If preemption is enabled, but the other router has higher priority then it will remain Active.

Operational priority may not match the configured priority. If interfaces are down, this negatively impacts operational priority.

Preemption is Disabled and Another Router Already Active

Examine the output of the **show vrrp** command:

- If the preemption is disabled.
- If the router has higher priority, it will not take over unless preemption is enabled.

Tracked Interface Failing, Router State Not Changed

On both routers:

Step 1 `RP/0/RSP0/CPU0:router# show vrrp detail`

If preemption is enabled but this router has higher operational priority than the other router, this router will remain active. Configured priority or the decrement for tracked interfaces needs to be configured appropriately such that the state transition takes place. If the IP address is same as the interface IP address, router state will not change to Standby.

Step 2 `RP/0/RSP0/CPU0:router# show vrrp trace`

Step 3 `RP/0/RSP0/CPU0:router# show eth-output trace location interface-running-vrrp`

VRRP State Flapping

On both routers:

Step 1 `RP/0/RSP0/CPU0:router# show vrrp detail`

Step 2 `RP/0/RSP0/CPU0:router# show vrrp trace`

Step 3 `RP/0/RSP0/CPU0:router# show eth-output trace location interface-running-vrrp`

Step 4 `RP/0/RSP0/CPU0:router# debug vrrp packets`

Check timestamps to determine whether there is a delay in sending or receiving packets. Check the CPU usage to see if some process is hogging the system resources.

Step 5 `RP/0/RSP0/CPU0:router# show spp node-counters location interface-running-vrrp`

Release 3.7.2 Final Draft?Cisco Confidential

More Than One VRRP Router Active

Step 1 Verify that the same IP is configured on both ends.

```
RP/0/RSP0/CPU0:router# show vrrp detail
```

Step 2
```
RP/0/RSP0/CPU0:router# show vrrp trace
```

Step 3
```
RP/0/RSP0/CPU0:router# show eth-output trace location interface-running-vrrp
```

Step 4 Check timestamps to determine whether there is a delay in sending or receiving packets.
Check the CPU usage to see if a process is overusing resources.
Check for lines similar to **`RP/0/RSP0/CPU0:Sep 8 14:16:39.217 : vrrp[357]: Gi0/5/0/0: VR1: Pkt: ADVER: IN: pri 100 src 192.0.0.11`**. This means advertisement packets are being received by VRRP. If these are absent, no packets are being received and VRRP becomes active.

```
RP/0/RSP0/CPU0:router# debug vrrp packets
```

Step 5 Enter debug VRRP packets on the peer.

Look for lines similar to **`RP/0/RSP0/CPU0:Sep 8 14:18:47.876 : vrrp[357]: Gi0/5/0/0: VR1: Pkt: ADVER: Out: pri 100 src 192.0.0.11`**. This means the peer is sending VRRP packets.

Step 6 Check the output of the **show spp node-counters location** *interface-running-vrrp* on both routers, and look for packet drops.

```
RP/0/RSP0/CPU0:router# show spp node-counters location interface-running-vrrp
```

VRRP Active Router Not Forwarding Traffic

On both routers:

Step 1 Find the virtual MAC address for the group.

```
RP/0/RSP0/CPU0:router# show vrrp detail
```

Step 2
```
RP/0/RSP0/CPU0:router# show vrrp trace
```

Step 3
```
RP/0/RSP0/CPU0:router# show eth-output trace location interface-running-vrrp
```

Step 4
```
RP/0/RSP0/CPU0:router# show ether-ctrl trace
```

Step 5 Ensure that the virtual MAC address is in the unicast address filter list. Verify the router is receiving traffic.

```
RP/0/RSP0/CPU0:router# show controller interface-running-vrrp
```

Step 6
```
RP/0/RSP0/CPU0:router# show controller np struct VRRP_MAC all
```

Traffic Loss or Unexpected VRRP State After Interface shut/no shut

In case of shut / no shut on a VRRP-enabled interface, the following has been observed:

- If preemption is enabled, recovery times are higher than failover times. This means higher traffic loss has occurs when the interface is no shut.
- If preemption is disabled, some VRRP groups are preempted after no shut of an interface.

If you observe either of the above after an interface no shut, follow the steps below.

On both routers:

Step 1 `RP/0/RSP0/CPU0:router# show vrrp detail`

Step 2 `RP/0/RSP0/CPU0:router# show vrrp trace`

Step 3 `RP/0/RSP0/CPU0:router# show eth-output trace location` *interface-running-vrrp*

Step 4 `RP/0/RSP0/CPU0:router# show ether-ctrl trace`

Step 5 `RP/0/RSP0/CPU0:router# show controller` *interface-running-vrrp*

Step 6 `RP/0/RSP0/CPU0:router# show controller np struct` *VRRP_MAC* `all`

Step 7 `RP/0/RSP0/CPU0:router# debug vrrp packets` *interface* for the interface on which no shut is being performed.

Step 8 Perform the no shut.

Step 9 Observe the console logs and look for lines similar to **- `RP/0/RSP0/CPU0:Sep 8 14:16:39.217 : vrrp[357]: Gi0/5/0/0: VR1: Pkt: ADVER: IN: pri 100 src 192.0.0.11`**. Note the time lag between the no shut and the first such message seen. For that amount of time, there is traffic loss between two routers.

Step 10 If there is no traffic flowing between two routers after no shut event, check STP configuration on the Cisco ASR 9000 Series Router. Lowering the fwd delay timer might help in reducing the traffic loss.

Step 11 For preemption disabled case, if the groups still preempt after reducing the fwd delay timer, repeat steps <1 to 4> as listed above, and find out the time period of traffic loss between the two routers. The preemption can be avoided by configuring the minimum delay to be higher than the time period of traffic loss. Minimum delay can be configured as follows -

```
RP/0/RSP0/CPU0:router# router vrrp interface gigabitEthernet 0/2/0/10 vrrp delay minimum
10 reload 5
```