CHAPTER 10

Troubleshooting Quality of Service and Access Control Lists

This chapter describes techniques for troubleshooting *quality of service* (QoS) and access control list (ACL) features.

The system supports the following QoS features:

- 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 IP Precedence bit classification with marking, policing and scheduling, ingress and egress.

This chapter includes the following sections:

- Bundle Members Not Configured with service-policy, page 10-241
- Troubleshooting Access Control Lists, page 10-241

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 interface-name* **[output | input]**
5. **show qos interface** *type interface-name* **[output | input]**
6. **show qos-ea interface** *type interface-name* **[output | input]**
7. **show qos-ea km**
8. **debug qos-ea ?**

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 policy-map 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** *type interface-name* **[output|input]** **Example:** `RP/0/RSP0/CPU0:router# show policy-map interface g0/2/0/0`	View all the statistics, queue IDs and class information.

	Command or Action	Purpose
Step 5	**show qos interface** *type interface-name* **[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 interface-name* **[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	**debug qos-ea ?** **Example:** `RP/0/RSP0/CPU0:router# debug qos-ea ?`	—

Service-Policy Configuration Is Rejected

Step 1 If the service-policy configuration is rejected or failed to commit, check the error message with the **show configuration failed** command.

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

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

Step 3 Check the OOR.

Step 4 Verify resources used.

Step 5 Verify summary information.

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

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 VMR entries for each class.

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

Step 6 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 7 Verify if in ingress QoS lookup occurs before Layer 2 ingress rewrite and that in egress Layer 2 rewrite occurs before QoS lookup.

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

Step 8 `RP/0/RSP0/CPU0:router# show run interface type node-id`

Step 9 `RP/0/RSP0/CPU0:router# show run policy-map policy`

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

Packets in Wrong Queue

Step 1 RP/0/RSP0/CPU0:router# **show qos-ea** *type* ***interface*** **{input | output} location** *node-id*

Step 2 Verify the packets are correctly classified.

Step 3 Verify hash structure.

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

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 type interface {input | output} location node-id
```

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 type interface {input | output} location node-id
```

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 type interface {input | output} location node-id
```

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

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

Step 4 `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 type interface {input | output} location node-id
```

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 type interface {input | output} location node-id
```

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

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

Weighted Random Early Detection Incorrect

Step 1 Ensure that packets are correctly classified.

Step 2 Verify whether the weighted random early detection (WRED) curves are correctly configured with minimum and maximum thresholds of each curve are as per the configured service policy.

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

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

```
RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id
```

Step 4 `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 type interface {input | output} location node-id
```

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 type interface {input | output} location node-id
```

Step 5 `RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member 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 that excess weights are configured as per the bandwidth remaining ratio configuration.

```
RP/0/RSP0/CPU0:router# show qos type interface {input | output} location node-id
```

Step 4 `RP/0/RSP0/CPU0:router# show qos-ea type interface {input | output} location node-id`

Step 5 `RP/0/RSP0/CPU0:router# show policy-map interface filename {input | output} [member filename]`

Step 6 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 type interface {input | output} location node-id`

Step 7 If commit and excess weights are correct:

a. Check queue size of each class.

b. Increase the queue size.

Non-zero Queue(conform) and Queue(exceed) Counters In show policy-map Commands

This section explains what to do if the **show policy-map** command displays non-zero values for Queue(conform) and Queue(exceed) counters.

On the ASR 9000, every hardware queue has a configured committed information rate (CIR) and peak information rate (PIR) value. CIR corresponds to the guaranteed bandwidth for the queue and PIR corresponds to the maximum bandwidth (also known as the shape rate) for the queue. To configure the CIR and PIR, use the **police rate** command. The syntax is:

```
RP/0/RSP0/CPU0:router# police rate {value [units] | percent percentage} [burst burst-size
[burst-units]] [peak-rate value [units]] [peak-burst peak-burst [burst-units]]
```

In this command, CIR is the police rate *value* and PIR is the police peak-rate *value*.

Example

```
RP/0/RSP0/CPU0:router# show policy-map location 0/7/0/30

GigabitEthernet0/7/0/30.1001 output: STS-1

Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :             4167078/4179237900           79024
    Transmitted         :             2224974/2229365484           42017
    Total Dropped       :             1942095/1949863380           36801
    Policy SHAPE-OUT Class BFD-OUT
      Classification statistics          (packets/bytes)     (rate - kbps)
        Matched             :                4786/296732                5
        Transmitted         :                4786/296732                5
        Total Dropped       :                   0/0                     0
      Policing statistics                (packets/bytes)     (rate - kbps)
        Policed(conform)    :                4786/296732                5
        Policed(exceed)     :                   0/0                     0
        Policed(violate)    :                   0/0                     0
        Policed and dropped :                   0/0
        Policed and dropped(parent policer)  : Un-determined
      Queueing statistics
        Queue ID                               : 8
        High watermark  (Unknown)
        Inst-queue-len  (packets)              : 0
        Avg-queue-len   (Unknown)
        Taildropped(packets/bytes)             : 0/0
        Queue(conform)      :                     0/0                     0
        Queue(exceed)       :                  4786/296732                5
        RED random drops(packets/bytes)        : 0/0
```

A non-zero value displayed for Queue(exceed) does not mean that there is a packet drop, but rather the number of packets above the configured (or system selected) CIR rate on that queue. Although you could change the Queue(exceed) behavior by explicitly configuring a bandwidth and/or a shape rate on each queue, it is not necessary to do so. You can treat these counters as informational or simply ignore them.

In the **police rate** command, if you do not explicitly configure a value for the police rate (the CIR), the system automatically assigns one. The Queue(conform) counter in the **show policy-map** command is the number of packets/bytes that were transmitted within this CIR value, and the Queue(exceed) value is the number of packets/bytes that were transmitted within the PIR value. The Queue(exceed) counter is based on whether the **parent bandwidth is exceed or conform**. If there is no parent bandwidth, all traffic is counted as excess.

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 the ACL 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 Restart the qos_ma_ea process.

After QoS EA Restarts, show policy-map interface Fails

- **show running-config**
- **show qos-ea ha chkpt all info location** *node-id*
- **show qos-ea ha chkpt if-qos all location** *node-id*

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 node-id
```

Step 2 If there is no error, do the following:

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

b. 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 node-id
```

Step 2 If there is no error, do the following:

a. RP/0/RSP0/CPU0:router# **debug generic**

b. Collect the debugs by performing 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} location** *node-id*

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 {qos | qos-ea} interface {output | input} location** *node-id*

Troubleshooting Access Control Lists

This section explains how to troubleshooting problems with *access control lists* (ACLs). 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 topics:

Using show and debug Commands

SUMMARY STEPS

1. **show access-lists ipv4 [rp-access [hardware {ingress | egress}** {*sequence-number* **| location** *node-id* **| summary [rp-access] | maximum [detail] [usage** {*pfilter location node-id*}]
2. **debug feature-ea-dll {all | error | info | resmgr | vmr}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show access-lists ipv4** [**rp-access** [**hardware** {**ingress** | **egress**} {*sequence-number* | **location** *node-id* | **summary** [**rp-access**] | **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 line card (LC). • **pfilter**—Packet filtering for the LC.
Step 2	**debug feature-ea-dll {all | error | info | resmgr | vmr}**	View error messages at various levels.

ACL Messages Not Appearing

Step 1 View ACEs in the ACLs.

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

Step 2 View TCAM entries in the ACLs..

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

Step 3 Configure the logs in the ACL.

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

Workaround

If an entry with the fragment flag is not present, remove the access-list from all the interfaces and reapply it.

Note Fragmented packets are not matched against the deny ACE without **fragment** keyword. Add the explicit **fragment** keyword in the ACE to deny the fragment packet. See the workaround commands in the "Fragmented Packets Being Accepted" section on page 10-243.

Fragmented Packets Being Accepted

Step 1 View ACEs in the ACLs.

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

Step 2 View IPv4 counters, for example, fragment.

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

Step 3 View TCAM entries in the ACLs.

Step 4 Ensure that the **fragment** keyword is in the ACE.

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

Step 5 Check the fragment packet count received by the device.

Step 6 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, perform 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 Counter Incorrect or Not Working

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
```

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

View errors encountered when the configuration was applied.

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

Workaround

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

Single ACE Using Many TCAMs

Step 1 View ACEs in the ACLs.

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

Step 2 Check the number of ranges in the ACE.

ACL Using Varying TCAM Space

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

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

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 ACEs in the ACLs.

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

Step 3 View trace log for pfilter_ea.

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

View ACEs configured for the ACL.

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

Workaround

Remove the old ACL before applying the new one.

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 IPv4 trace information.

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

Step 4 View the Ethernet services trace.

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

DF Bit Not Supported

The Do Not Fragment (DF) bit is not supported as match criteria in the current release.

Max ACL Limit Reached

The maximum number of ACL IDs per network processor (NP) is 2048. Interfaces share TCAM entries for the ACL name and direction.

Unsupported Combinations in ACL

There may be unsupported field combinations in the access-list. Verify that the combinations in the access-list are currently supported. 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.

TCAMs Out of Resources

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