NOKIA

Nokia Service Router Linux

QUALITY OF SERVICE GUIDE
RELEASE 22.11

3HE 19041 AAAA TQZZA
Issue 01

November 2022

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

Table of contents

1 About this guide

This document describes configuration details for the Quality of Service (QoS) feature set used with the Nokia Service Router Linux (SR Linux).

This document is intended for network technicians, administrators, operators, service providers, and others who need to understand how the router is configured.

Note:
This manual covers the current release and may also contain some content that will be released in later maintenance loads. See the *SR Linux Release Notes* for information on features supported in each load.

1.1 Precautionary and information messages

The following are information symbols used in the documentation.

DANGER: Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.

WARNING: Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.

Caution: Caution indicates that the described activity or situation may reduce your component or system performance.

Note: Note provides additional operational information.

Tip: Tip provides suggestions for use or best practices.

1.2 Conventions

Nokia SR Linux documentation uses the following command conventions.

- **Bold** type indicates a command that the user must enter.
- Input and output examples are displayed in `Courier` text.
- An open right-angle bracket indicates a progression of menu choices or simple command sequence (often selected from a user interface). Example: **start** > **connect to**.
- A vertical bar (|) indicates a mutually exclusive argument.
- Square brackets ([ ]) indicate optional elements.

- Braces ({ }) indicate a required choice. When braces are contained within square brackets, they indicate a required choice within an optional element.
- *Italic* type indicates a variable.

Generic IP addresses are used in examples. Replace these with the appropriate IP addresses used in the system.

2 What's new

Topic	Location
Multifield classification policies and Ingress DSCP rewrite	Multifield classification policies Ingress DSCP rewrite
Named queues and forwarding classes (7250 IXR only)	Named queues and forwarding classes
Ingress subinterface traffic policing	Ingress subinterface traffic policing
"How QoS works" sections updated for new feature content	How QoS works for transit traffic How QoS works for VXLAN traffic How QoS works for router-terminated traffic How QoS works for router-originated traffic

3 Quality of service overview

Quality of Service (QoS) provides an appropriate level of service for packets as they flow inside the switch and between switches in the network. The required level of service depends on the application that generates the flow of packets, and can be defined by the application's sensitivity to packet loss, delay, and jitter.

QoS functionality is supported on the 7250 IXR, 7220 IXR-D2, D3, and D5, and the 7220 IXR-H2 and H3.

Note: The 7220 IXR-D5 supports the following subset of SR Linux QoS functionality:

- DSCP classifier and rewrite-rule policies (VXLAN not supported)
- Queue depth (unicast only)
- WRED slope
- ECN slope
- WRR
- Strict priority scheduling
- Forwarding class peak rate (unicast only)

You can group packets that require a similar treatment (per-hop behavior) into a Forwarding Class (FC), also known as a behavior aggregate. You can specify up to eight FCs. Traffic is scheduled and can optionally be marked based on its FC.

A configurable drop probability expresses the packet loss sensitivity. Assign a low drop probability to packets that are sensitive to loss. To provide the required congestion management and intelligent discard decisions when congestion occurs, balance the traffic classifications between low, medium, and high drop probability.

3.1 How QoS works for transit traffic

This section describes how QoS applies to transit packets on the SR Linux.

1. Packets are received on a subinterface.
2. Each received packet is classified as belonging to one of eight forwarding classes (corresponding to forwarding class indexes 0 to 7) and one of three drop probabilities (low, medium, or high).
   - **For IP packets**:
     - If the packet matches a multifield classifier policy configured on the ingress subinterface, the forwarding class (FC) and drop probability level are determined entirely from that policy. In addition, if this policy includes a DSCP rewrite action, the DSCP value for the packet is rewritten accordingly.
     - Otherwise, if the packet matches a DSCP classifier policy configured on the ingress subinterface, the forwarding class and drop probability level are determined from that policy.

Note: If there is no entry of this policy matching the received DSCP, the assigned forwarding class index is 0 and the assigned drop probability is low. This FC and drop probability classification corresponds to a best effort treatment.

  - If there is no multifield classifier or DSCP classifier policy bound to the ingress subinterface, the FC and drop probability are determined from the default DSCP classifier policy. See Table 1: System default DSCP classifier policy.

- **For VLAN-encapsulated, non-IP packets**:
  - If the packet matches a dot1p (IEEE 802.1p) classifier policy configured on the VLAN subinterface, the FC and drop probability are determined from that policy.
  - If there is no matching dot1p classifier policy, or no dot1p policy is explicitly bound to the VLAN subinterface, the FC and drop probability are determined from the default dot1p policy.

3. Both IP and non-IP traffic can be directed to a subinterface traffic policer. In this case, packets are metered to determine compliance with a traffic profile. At the output of the policer, every packet is marked with a color (green, yellow, or red) that represents whether it conforms, exceeds, or violates the traffic profile. The drop probability for all packets can then be updated based on their conformance to the policy, and violating (red) packets can be dropped altogether.
4. A forwarding lookup on the packet determines its egress port.
5. On the 7250 IXR, if the packet is a unicast packet, it is associated with a Virtual Output Queue (VOQ) based on the ingress port, egress port, and FC.

   On a 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3, the packet is associated directly with an Egress Queue (EGQ) of the egress port, based on the FC of the packet and its type (either unicast or multicast).
6. While it waits for its VOQ or EGQ to be serviced, the packet is stored in buffer memory. The total amount of buffer memory varies by platform.
7. The packet is dropped if the buffer memory is close to full or if the Maximum Burst Size (MBS) of the VOQ or EGQ is exceeded.

   The MBS is one of the parameters that is configurable in a queue template. When a queue template is applied to a set of queues, all of those queues have the MBS value specified in the template. If the MBS is not specified in a queue template, the default value is platform dependent. The MBS is not a guaranteed allocation of buffer memory.
8. When the packet is Explicit Congestion Notification (ECN)-capable, and ECN is enabled globally with the **qos explicit-congestion-notification** command, and the VOQ or EGQ has an active ECN slope that applies to the packet, the ECN field may be remarked depending on the current (weighted) queue depth.
   - If the current queue depth is below the configured **min-threshold-percent** of the ECN slope, the ECN field of the packet is unchanged.
   - If the current queue depth is above the configured **max-threshold-percent** of the ECN slope, the ECN field of the packet is (re)marked as Congestion Experienced (CE), ECN=11.
   - If the current queue size is between the **min-threshold-percent** and **max-threshold-percent** of the ECN slope, the ECN field of the packet is (re)marked as CE, ECN=11, based on a probability function that increases linearly from 0% at the minimum threshold to *n*% at the maximum threshold, where *n* is the operational **max-probability** of marking the packet.

Note: The operational values of the **max-probability** may be significantly different from the configured values based on internal hardware calculations. You can check the hardware-configured values for any slope calculations.

9. When the packet is non-ECN-capable (the ECN field is zero) and the egress queue has an active WRED slope for the drop probability of the packet, the packet may be dropped by the WRED algorithm, which operates as follows:
   - If the current queue depth is below the configured **min-threshold-percent** of the WRED slope, the packet is admitted to the queue.
   - If the current queue depth is above the configured **max-threshold-percent** of the WRED slope, the packet is dropped.
   - If the current queue size is between the minimum threshold and maximum threshold of the WRED slope, the packet is dropped based on a probability function that increases linearly from 0% at the minimum threshold to *n*% at the maximum threshold, where *n* is the operational **max-probability** of dropping the packet.

   Note: The operational values of the **max-probability** may be significantly different from the configured values based on internal hardware calculations. You can check the hardware configured values for any WRED slope calculations.

10. Each unicast queue and each multicast queue of an egress port is associated with a scheduler node. The mapping of queues to scheduler nodes is platform-dependent and cannot be configured. See Output queue scheduling.
11. Each egress queue can be individually configured with a Peak Information Rate (PIR). The PIR is configured as a percentage of the egress port bandwidth.

    By default, the PIR of each queue is 100%. The operational PIR is stored by the **peak-rate-bps** leaf in bits per second. The bits counted in this rate include the Layer 2 framing of the packet (including the 14 byte Ethernet header, the 4-byte VLAN header, and the 4-byte CRC) but exclude the 20-byte Layer 1 overhead (SFD, preamble, IPG).
12. The DSCP field in the IPv4 or IPv6 header of the outgoing packet can be rewritten. On the 7250 IXR, the DSCP field must be rewritten when ECN is enabled and the packet ECN field is non-zero. When there is a rewrite policy applied, the DSCP in the outgoing packet is based on the FC (and potentially also the drop probability) of the packet. If the FC (and drop-probability) matches an entry in the applied policy, the new DSCP value is based on the policy entry. If there is no matching entry in the applied policy, the new DSCP value is 0.
13. For VLAN-tagged traffic, the PCP field in the 802.1p header of the outgoing packet can be rewritten. When there is a dot1p marking policy applied to a subinterface, the dot1p value in the outgoing packet is based on the FC (and potentially also the drop probability) of the packet. If the FC (and drop-probability) matches an entry in the applied policy, the new PCP value is based on the policy entry.
    - On a bridged subinterface, if there is no matching entry in the applied policy, all pushed 802.1Q VLAN tags on the outgoing frame are marked with a PCP value of 0.
    - On a routed subinterface, if there is no dot1p policy applied, the forwarding class index from the ingress classification is encoded into the PCP field.

System default DSCP classifier policy

Table 1: System default DSCP classifier policy

DSCP values	Included DSCP names	Forwarding class	Drop probability
0, 2 to 7	CS0/BE	fc0	Low
1	LE	fc0	High
8 to 11	CS1, AF11	fc1	Low
12 to 13	AF12	fc1	Medium
14 to 15	AF13	fc1	High
16 to 19	CS2, AF21	fc2	Low
20 to 21	AF22	fc2	Medium
22 to 23	AF23	fc2	High
24 to 27	CS3, AF31	fc3	Low
28 to 29	AF32	fc3	Medium
30 to 31	AF33	fc3	High
32 to 35	CS4, AF41	fc4	Low
36 to 37	AF42	fc4	Medium
38 to 39	AF43	fc4	High
40 to 47	CS5, EF	fc5	Low
48 to 55	CS6/NC1	fc6	Low
56 to 63	CS7/NC2	fc7	Low

3.2 How QoS works for VXLAN traffic

When a 7220 IXR-D2 or D3 receives a terminating VXLAN packet on a subinterface, it classifies the packet to one of eight forwarding classes and one of three drop probabilities (low, medium, or high). The classification is based on the following considerations:

- The outer IP header DSCP is ignored.
- If the payload packet is non-IP, the classified FC index is 0 and the classified drop probability is low.
- If the payload packet is IP, and the **qos classifiers vxlan-default** command references a classifier policy, that policy is used to determine the FC and drop probability from the header fields of the payload packet.
- If the payload packet is IP, and the **qos classifiers vxlan-default** command does not reference a classifier policy, the default DSCP classifier policy is used to determine the FC and drop probability from the header fields of the payload packet.

- If a dot1p policy is applied on the subinterface, then the PCP field is set to 0. If no dot1p policy is applied, then the FC index value from the ingress classification is encoded into the PCP field.

When the 7220 IXR-D2 or D3 adds VXLAN encapsulation to a packet and forwards it out a subinterface, the inner header IP DSCP value is not modified if the payload packet is IP, even if the egress routed subinterface has a DSCP rewrite rule policy bound to it that matches the packet FC and drop probability. If a DSCP rewrite policy is bound to the egress routed subinterface, that policy modifies the outer header IP DSCP.

Note: If transit VXLAN traffic arrives on a subinterface with a configured subinterface traffic policer, it is policed the same as any other transit traffic. But if the VXLAN traffic terminates on the subinterface, the policing does not apply.

3.3 How QoS works for router-terminated traffic

This section describes how QoS applies to traffic that terminates on the SR Linux.

1. A packet is received on a subinterface and is determined to need extraction toward the CPM. The packet is directed to one of the queues associated with the CPM as a destination "physical port" based on its protocol and type. Different traffic types have their own independent queue, for example:
   - sflow
   - ICMPv4 ping
   - BFD
   - ARP
   - ICMPv6 neighbour solicitation and neighbor advertisement
   - BGP
   - gRPC
   - LLDP
   - IPv4 packets with IP options and IPv6 packets with extension headers
   - DHCPv6
   - IS-IS hello PDUs
   - OSPF/OSPFv3 hello PDUs
2. Some of the queues toward the CPM have a PIR shaping rate designed to prevent an overload of one type of traffic. The PIR shaping rates vary by platform.

3.4 How QoS works for router-originated traffic

This section describes how QoS applies to traffic that originates on the SR Linux.

1. An application on the SR Linux CPM has an IPv4 or IPv6 packet to send to another system.
2. The CPM datapath assigns a DSCP to the self-generated packet based on its protocol and the hard coded mapping shown in Table 2: Default forwarding class and DSCP marking for router-originated traffic.

Except for ICMP and ICMPv6 echo-request packets, the DSCP values cannot be overridden. For originated echo-request packets, the DSCP override value can be configured as an optional parameter of the **ping** command.

3. The CPM datapath looks up the DSCP from the previous step (either the fixed value or the override value for echo-request) in the default DSCP classifier policy (see Table 1: System default DSCP classifier policy) to determine the FC and drop probability level.
4. A forwarding lookup determines the egress port.
5. On the 7250 IXR, the packet is sent to the egress line card and added to a Virtual Output Queue (VOQ) appropriate for its forwarding class and the egress port. The decision to drop or enqueue the packet in the VOQ and the scheduling of the VOQ follows the previous description for transit traffic. There is no scheduling differentiation between router-originated traffic and transit traffic of the same FC on the egress IMM.
6. The packet is directed to the egress queue appropriate for its forwarding class and packet type. On the 7220 IXR-D2, D3, and D5 and the 7220 IXR-H2 and H3, the decision to drop or enqueue the packet in the egress queue and the scheduling of the egress queue follow QoS treatment of transit traffic described in How QoS works for transit traffic.
7. The DSCP field in the IPv4 or IPv6 header is always written based on the hard coded mapping described in Table 2: Default forwarding class and DSCP marking for router-originated traffic. If the packet also matches a DSCP policy rewrite rule or a dot1p rewrite rule applied to the output subinterface, the rewrite-rule policy is ignored.

Default forwarding class and DSCP marking for router-originated traffic

Table 2: Default forwarding class and DSCP marking for router-originated traffic

Protocol / message type	Forwarding class index	Drop probability	DSCP marking
IPv4 ARP request/reply	6	Low	N/A
ICMPv4 including echo-request[1], echo- reply[2], dest-unreachable, redirect, time-exceeded, parameter-problem	0	Medium	0
ICMPv4 echo-request with ToS/DSCP override = *x*	look up *X* in system-default DSCP classifier	look up *X* in system-default DSCP classifier	*x*
ICMPv4 echo-reply to echo-request with non-zero DSCP *x*	look up *X* in system-default DSCP classifier	look up *X* in system-default DSCP classifier	*x*
UDP traceroute	0	Low	0
IPv6 neighbor solicitation	6	Low	48 (CS6/NC1)
IPv6 neighbor advertisement	6	Low	48 (CS6/NC1)

1 Echo-request generated by a ping command with no DSCP parameter specified.

2 Echo-reply to an echo-request packet with DSCP=0.

Protocol / message type	Forwarding class index	Drop probability	DSCP marking
All other ICMPv6 including dest unreachable, packet-too-big, time-exceeded, parameter-problem, echo-request, echo-reply, router-solicitation, redirect	0	Medium	0
ICMPv6 echo-request with DSCP override = *x*	look up *x* in system-default DSCP classifier	look up *x* in system-default DSCP classifier	*x*
ICMPv6 echo-reply to echo-request with non-zero DSCP *x*	look up *x* in system-default DSCP classifier	look up *x* in system-default DSCP classifier	*x*
BFD	6	Low	48 (CS6/NC1)
BGP	6	Low	48 (CS6/NC1)
DNS query	4	Low	34 (AF41)
FTP/TFTP	4	Low	34 (AF41)
gNMI	4	Low	34 (AF41)
JSON RPC	4	Low	34 (AF41)
LLDP	N/A	Low	N/A
NTP	4	Low	34 (AF41)
sFlow	0	Low	0
SNMP	4	Low	34 (AF41)
SSH	4	Low	34 (AF41)
Syslog	4	Low	34 (AF41)
TACACS+	4	Low	34 (AF41)

4 Multifield classification policies

SR Linux supports rule-based QoS multifield classification of IPv4 and IPv6 packets. Each IPv4 and IPv6 multifield classification policy is structurally similar to an IPv4 or IPv6 interface ACL, containing a list of ordered entries, each specifying a set of match conditions and associated actions.

Each multifield classification rule, or entry, has a sequence ID. The policy evaluates packets starting with the entry with the lowest sequence ID, progressing to the entry with the highest sequence ID. Evaluation stops at the first matching entry (that is, when the packet matches all of the conditions specified by the multifield classification entry).

Multifield classification policies are supported only on the 7220 IXR-D2/D2L/D3/D3L.

Match conditions

Each IPv4 or IPv6 policy entry can specify zero or more of the following match conditions.

Table 3: Multifield classification match conditions

Match condition	Description	IPv4 policy support	IPv6 policy support
Destination IP	Matches by prefix or by address and mask	✔	✔
Destination port	Matches by destination TCP or UDP port or range. Comparison operators define whether the matching destination port must be: • equal to the specified value • greater than or equal to the specified value • less than or equal to the specified value	✔	✔
DSCP set	Matches one of the DSCP values listed. This setting matches against the ingress DSCP value (not the rewritten DSCP value). If left empty, any DSCP value matches.	✔	✔
Fragment/first-fragment	Matches a packet that is a fragment, and optionally the first fragment	✔	Not applicable
ICMP type/code	Matches one of the specified ICMP type and code combinations	✔	Not applicable
ICMPv6 type/code	Matches one of the specified ICMPv6 type and code combinations	Not applicable	✔
Next-header number	Matches the first next-header field (in the IPv6 fixed header) if it contains the specified value	Not applicable	✔
Protocol number	Matches the IP protocol type field	✔	Not applicable
Source IP	Matches by prefix or by address and mask	✔	✔

Match condition	Description	IPv4 policy support	IPv6 policy support
Source port	Matches source TCP or UDP port or range. Comparison operators define whether the matching source port must be: • equal to the specified value • greater than or equal to the specified value • less than or equal to the specified value	✔	✔
TCP flags	Matches the TCP flag names: RST, SYN, and ACK based on a logical expression using the &, |, and ! operators	✔	✔

Supported actions

Each IPv4 or IPv6 policy entry supports the following actions:

- Set the forwarding class (mandatory action in each entry)
- Set the drop probability (optional action in each entry, default is low)
- Rewrite the ingress DSCP value (optional action in each entry)

Related topics

Ingress DSCP rewrite

4.1 Supported interfaces: routed, bridged, and IRB

You can bind a multifield classification policy (IPv4, IPv6, or both) to the following subinterface types:

- Routed subinterface of a default or ip-vrf network instance, associated with an Ethernet port, LAG, or IRB
- Bridged subinterface of a mac-vrf network instance, associated with an Ethernet port or LAG

DSCP classification policy and multifield classifier policy on the same subinterface

You can apply both a DSCP classification policy and a multifield classifier policy to the same IP/routed subinterface for a specified protocol (IPv4 or IPv6). If an ingress IPv4 or IPv6 packet matches a multifield classification rule, its forwarding class and drop probability are determined solely by the matching multifield classification rule. If an ingress IPv4 or IPv6 packet does not match any multifield classification rule, forwarding class and drop probability are determined as follows:

- **On 7220 IXR-D2/D2L/D3/D3L**:
  Forwarding class and drop probability are determined by the configured or default DSCP policy.

4.2 Scaling and restrictions

The following describe scaling and restrictions for multifield classification policies.

7220 IXR-D2/D2L/D3/D3L

On the 7220 IXR-D2/D2L/D3/D3L:

- Multifield classifier policies always operate in subinterface-specific mode, with no option available for a shared mode. As a result, the number of TCAM entries required to implement one multifield classifier policy is N * S, where N is the number of TCAM entries required to implement one instance of the policy and S is the number of subinterfaces where the policy is applied.
- SR Linux blocks the binding of a MAC ACL and an IPv4 or IPv6 multifield classifier policy on the same subinterface. MAC ACL and multifield classification are mutually exclusive options in R22.11.

4.3 Ingress DSCP rewrite

Packets arriving on an interface can have IP DSCP markings that are not trusted. For example, when the upstream devices do not classify or mark the packets properly, or when the interface is at the beginning of a service SLA that is defined in terms of application characteristics instead of DSCP. In this case, an ingress DSCP rewrite action in the multifield classification policy can replace the DSCP value for matching IPv4 or IPv6 packets with a new value.

Note: If an egress DSCP rewrite rule is also applied to a Layer 3 subinterface, it does not overwrite the ingress DSCP rewrite action. In this case, the packet is transmitted with the DSCP specified in the ingress DSCP rewrite rule.

The following table provides more details about the packet flows that are supported with ingress DSCP rewrite.

Table 4: Supported packet flows with ingress DSCP rewrite

Ingress packet	Ingress subif type	Ingress subif MF classifier entry action	Forwarding	IRB subif MF classifier entry action	Egress subif(s) DSCP rewrite policy	Egress Packet
IP/ Ethernet	bridged (mac-vrf)	set fc=A dscp-rewrite=B	L2 switched	configured or not configured (no effect in either case)	bridged subif DSCP rewrite policy: NO effect	DSCP=B
IP/ Ethernet	bridged (mac-vrf)	set fc=A dscp-rewrite=B	L3 routed between mac-vrf1 and mac-vrf2 using IRB	not configured	mac-vrf2 IRB subif DSCP rewrite policy: NO effect mac-vrf2 bridged subif DSCP rewrite policy: NO effect	DSCP=B
IP/ Ethernet	bridged (mac-vrf)	set fc=A dscp-rewrite=B	L3 routed between mac-vrf1 and	IRB of mac-vrf1: set fc=C	mac-vrf2 IRB subif DSCP rewrite policy: NO effect	DSCP=D

Ingress packet	Ingress subif type	Ingress subif MF classifier entry action	Forwarding	IRB subif MF classifier entry action	Egress subif(s) DSCP rewrite policy	Egress Packet
			mac-vrf2 using IRB	dscp-rewrite=D	mac-vrf2 bridged subif DSCP rewrite policy: NO effect	
IP/ Ethernet	bridged (mac-vrf)	set fc=A dscp-rewrite=B	L3 routed followed by VXLAN encap (symmetric or asymmetric)	not configured	routed subif DSCP rewrite policy: only changes outer DSCP	VXLAN with outer DSCP based on fc=A lookup in the DSCP rewrite policy, payload DSCP=B
IP/ Ethernet	bridged (mac-vrf)	set fc=A dscp-rewrite=B	L3 routed followed by VXLAN encap (symmetric or asymmetric)	IRB of mac-vrf1: set fc=C dscp-rewrite=D	routed subif DSCP rewrite policy: only changes outer DSCP	VXLAN with outer DSCP based on fc=C lookup in the DSCP rewrite policy, payload DSCP=D
IP/ Ethernet	routed (ip-vrf or default)	set fc=A dscp-rewrite=B	L3 routed	N/A	routed subif DSCP rewrite policy: NO effect	DSCP=B

4.4 Configuring multifield classification policies for input traffic

Procedure

The following examples create IPv4 and IPv6 multifield classifier policies, each containing one entry with multiple match conditions and associated actions.

Example: Configure IPv4 multifield classification policy

```
# info qos classifiers multifield ipv4-policy multifield-test
    qos {
        classifiers {
            multifield {
                ipv4-policy multifield-test {
                    entry 10 {
                        match {
                            fragment true
                            first-fragment true
                            protocol tcp
                            tcp-flags syn&ack
                            dscp-set [
                                AF11
                            ]
```

```
                            destination-ip {
                                prefix 10.10.20.0/24
                            }
                            icmp {
                                type 4
                                code [
                                    3
                                ]
                            }
                            source-ip {
                                address 10.10.10.1
                                mask 255.255.255.0
                            }
                            destination-port {
                                operator eq
                                value 25
                            }
                            source-port {
                                operator ge
                                value 2526
                            }
                        }
                        action {
                            forwarding-class fc6
                            drop-probability low
                            rewrite {
                                set-dscp 40
                            }
                        }
                    }
                }
            }
        }
    }
```

Example: Configure IPv6 multifield classification policy

```
# info qos classifiers multifield ipv6-policy multifield-test-v6
    qos {
        classifiers {
            multifield {
                ipv6-policy multifield-test-v6 {
                    entry 100 {
                        match {
                            next-header icmp
                            tcp-flags ack
                            dscp-set [
                                CS7
                            ]
                            destination-ip {
                                address fd92:834d:fae9:fa28::
                                mask ffff:ffff:ffff:fc00::
                            }
                            icmp6 {
                                type 1
                                code [
                                    3
                                ]
                            }
                            source-ip {
                                prefix fc00::/7
                            }
                            destination-port {
```

```
                            operator eq
                            range {
                                start 800
                                end 1000
                            }
                        }
                        source-port {
                            operator le
                            value 700
                        }
                    }
                    action {
                        forwarding-class fc7
                        drop-probability medium
                        rewrite {
                            set-dscp 56
                        }
                    }
                }
            }
        }
    }
```

4.5 Applying a multifield classification policy to a subinterface

Procedure

The following example applies the IPv4 and IPv6 multifield classification policies to inbound traffic on subinterface ethernet-1/1.1.

Example: Apply multifield classification policy to subinterface

```
# info interface ethernet-1/1 subinterface 1 qos
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                input {
                    classifiers {
                        multifield {
                            ipv4-policy multifield-test
                            ipv6-policy multifield-test-v6
                        }
                    }
                }
            }
        }
    }
```

5 DSCP classifier policy configuration for input traffic

When a DSCP classifier policy is applied to a subinterface, the policy attempts to match the 6-bit DSCP value in the IP header of incoming packets to one of its entries. If there is a match, the incoming packet is assigned to the specified forwarding class and drop probability; otherwise, the assigned forwarding class is 0 and the assigned drop probability is low.

Packets that require a similar treatment (per-hop behavior) are grouped into an FC, also known as a behavior aggregate. The SR Linux differentiates up to eight forwarding classes.

The drop probability can be one of high, medium, or low. If a queue-template with different WRED slopes is bound to a queue, then packets in that queue with a high drop probability are the first to be dropped when the queue experiences congestion, followed by packets with a medium drop probability, then by packets with a low drop probability. The default is low.

5.1 Configuring DSCP classifier policies

Procedure

The following example creates a DSCP classifier policy:

Example

```
--{ candidate shared default }--[  ]--
# info qos classifiers
 qos {
        classifiers {
            dscp-policy new-policy {
                dscp 0 {
                    forwarding-class fc0
                    drop-probability high
                }
                dscp 8 {
                    forwarding-class fc1
                    drop-probability high
                }
            }
        }
    }
```

Note: To create a new DSCP classification policy based on the default policy, you can copy the default policy from state in candidate mode, as shown in the following example:

```
# copy from state /qos classifiers dscp-policy default to /qos classifiers dscp-
policy test
```

Related topics

DSCP classifier policy application to subinterfaces

5.2 Using a DSCP classifier for VXLAN traffic

Procedure

On a 7720 IXR-D2 and D3, you can use a classifier policy to classify ingress packets received from any remote VXLAN VTEP. The policy applies to payload packets after VXLAN decapsulation is performed.

The following example shows how the DSCP classifier policy created in the previous example (**new-policy**) can be used for VXLAN traffic:

Example

```
--{ candidate shared default }--[  ]--
# info qos classifiers
 qos {
        classifiers {
            vxlan-default new-policy
        }
    }
```

5.3 DSCP classifier policy application to subinterfaces

If you apply a DSCP classifier policy to input traffic on a subinterface, incoming packets are evaluated against the policy, and matching packets are assigned to the forwarding class and drop probability specified by the policy. If no classifier policy is applied to the subinterface, the system default DSCP classifier (with the reserved name *default*) is used.

5.3.1 Applying a DSCP classifier policy to input traffic (7250 IXR)

Procedure

The following example applies a DSCP classifier policy to inbound IPv6 traffic on a subinterface with a 7250 IXR system:

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                input {
                    classifiers {
                        ipv6-dscp new-policy
                    }
                }
            }
        }
    }
```

5.3.2 Applying a DSCP classifier policy to input traffic (7220 IXR)

Procedure

The following example applies a DSCP classifier policy to inbound traffic on a subinterface with a 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 system:

Note: The 7220 IXR-D2, D3, and D5 and 7220 IXR-H2 and H3 systems do not support separate classifier policies for IPv4 and IPv6 traffic, but you can apply a common policy that applies to both IPv4 and IPv6 traffic.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                input {
                    classifiers {
                        dscp new-policy
                    }
                }
            }
        }
    }
```

6 DSCP rewrite-rule policy configuration for output traffic

When a DSCP rewrite-rule policy is applied to a subinterface, the policy attempts to match the forwarding class (and optionally the drop-probability) of outbound packets to one of its entries. If there is a match, the DSCP value of the outbound packet is changed to the value specified by the policy. If the forwarding class of the packet does not match a rule of the rewrite-rule policy, the DSCP value is changed to 0.

On 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 systems, if no DSCP rewrite-rule policy is applied to a subinterface, the incoming packet's DSCP remains unchanged at egress.

6.1 Configuring DSCP rewrite-rule policies

Procedure

The following example creates a rewrite-rule policy:

Example

```
--{ candidate shared default }--[  ]--
# info qos rewrite-rules
 qos {
        rewrite-rules {
            dscp-policy normalize {
                map fc0 {
                    dscp 1
                map fc0 {
                    dscp 7
                map fc1 {
                    dscp 10
                    drop-probability low {
                        dscp 11
                    }
                    drop-probability high {
                        dscp 13
                map fc2 {
                    dscp 23
                map fc3 {
                    dscp 31
                }
            }
        }
    }
```

Related topics

Rewrite-rule policy application to subinterfaces

6.2 Rewrite-rule policy application to subinterfaces

When a rewrite-rule policy is applied to output traffic on a subinterface, outbound packets are evaluated against the policy. The policy subjects all packets to remarking, with some exceptions. If no rewrite-rule policy is applied to the subinterface, the DSCP marking of the traffic leaving the subinterface is unchanged, unless it is ECN-capable traffic forwarded by a 7250 IXR system or VXLAN traffic originated by a 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 system. For these exceptions, DSCP may be remarked even in the absence of a rewrite-rule policy applied to the egress subinterface.

On all platforms, rewrite-rule policies do not affect DSCP marking of self-generated traffic.

6.2.1 Applying a rewrite-rule policy to output traffic (7250 IXR)

Procedure

The following example applies a rewrite-rule policy to outbound IPv4 traffic on a subinterface with a 7250 IXR system:

Note: 7250 IXR systems support separate rewrite policies for IPv4 and IPv6 egress traffic.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                output {
                    rewrite-rules {
                        ipv4-dscp new-rule
                    }
                }
            }
        }
    }
```

6.2.2 Applying a rewrite-rule policy to output traffic (7220 IXR)

Procedure

The following example applies a rewrite-rule policy to outbound traffic on a subinterface with a 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 system:

Note: Common rewrite policies that apply to both IPv4 and IPv6 traffic are supported on 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 systems.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
```

```
                output {
                    rewrite-rules {
                        dscp new-rule
                    }
                }
            }
        }
    }
```

7 Dot1p classification and marking

SR Linux supports IEEE 802.1p (dot1p) classification and marking using the Priority Code Point (PCP) field. When one or more IEEE 802.1Q VLAN tags are added to an Ethernet frame, the Class of Service (CoS) of the frame can be set using the PCP field in the outermost VLAN tag. The 3-bit PCP field can specify eight different classes of service, allowing Ethernet frames that carry non-IP or non-MPLS payloads to be assigned a required service level.

Supported platforms

Dot1p classification and marking is supported on the following platforms:

- 7220 IXR-D2/D2L: R22.6.3 (targeted release)
- 7220 IXR-D3/D3L: R22.6.3 (targeted release)

7.1 Dot1p classification

Dot1p classification refers to classification of a frame based on the PCP field in the outermost VLAN. The system assigns a forwarding-class and drop-probability to every packet at an early point in the packet forwarding pipeline. These assignments determine which packets to schedule or drop first when congestion occurs.

Each dot1p classifier policy can contain up to eight mapping rules. Each rule binds one of the eight possible PCP values (0 to 7) to a forwarding-class (fc0 to fc7) and to a drop-probability level (low, medium, or high).

For a dot1p classifier policy to take effect, you must apply the policy to at least one bridged subinterface. SR Linux supports dot1p classifier policies on any bridged subinterface of any Ethernet port or LAG. No limit exists on the number of bridged subinterfaces that can apply the same policy. (Routed subinterfaces are not supported.) Dot1p classification is applicable for non-IP packets only.

When a dot1p classifier policy is applied to a subinterface, if the PCP value for an incoming Ethernet frame does not match any configured dot1p rule, the frame is classified as `fc0` and `drop-probability low`.

Default dot1p classifier policy

SR Linux supports a default dot1p classifier policy, which always exists and is not modifiable. It is invisibly applied to all bridged subinterfaces that do not have a configured dot1p classifier policy applied. The following table describes the rules of the default policy.

Table 5: Default dot1p classifier policy

Dot1p	FC	Drop probability
0	fc0	low
1	fc1	low
2	fc2	low

Dot1p	FC	Drop probability
3	fc3	low
4	fc4	low
5	fc5	low
6	fc6	low
7	fc7	low

Dot1p classifier policy effects

The following table describes the effects of applying the dot1p classifier policy in relation to other configuration.

Table 6: Effect of dot1p classifier policy in relation to other configuration

Ingress sub-interface type	Ingress packet type	default fc of ingress sub-interface	default drop-probability of ingress subif	default dot1p-policy	explicit dot1p-policy (bound to ingress sub-interface)	default dscp-policy	explicit dscp-policy (bound to ingress sub-interface)
routed, untagged	IPv4/IPv6 untagged	—	—	not applicable	not supported	used if no explicit dscp-policy	used
routed, untagged	IPv4/IPv6 priority-tagged	—	—	not applicable	not supported	used if no explicit dscp-policy	used
routed, single-tagged	IPv4/IPv6 single-tagged	—	—	not applicable	not supported	used if no explicit dscp-policy	used
bridged, untagged	IPv4/IPv6 untagged	—	—	—	—	used if no explicit dscp-policy	used
bridged, untagged	IPv4/IPv6 priority-tagged	—	—	—	—	used if no explicit dscp-policy	used
bridged, single-tagged	IPv4/IPv6 single-tagged	—	—	—	—	used if no explicit dscp-policy	used
bridged, untagged	non-IP untagged	used	used	—	—	—	—

Ingress sub-interface type	Ingress packet type	default fc of ingress sub-interface	default drop-probability of ingress subif	default dot1p-policy	explicit dot1p-policy (bound to ingress sub-interface)	default dscp-policy	explicit dscp-policy (bound to ingress sub-interface)
bridged, untagged	non-IP priority-tagged	—	—	used if no explicit dot1p-policy	used	—	—
bridged, single-tagged	non-IP single-tagged	—	—	used if no explicit dot1p-policy	used	—	—
bridged, single-tagged	non-IP double-tagged	—	—	used if no explicit dot1p-policy	used	—	—

7.1.1 Configuring dot1p classifiers for input traffic

Procedure

The following example creates a dot1p classifier policy:

Note: The dot1p-policy name can be any name string other than `default`.

Example

```
--{ candidate shared default }--[  ]--
# info qos classifiers
 qos {
        classifiers {
            dot1p-policy new-dot1p-policy {
                dot1p 0 {
                    forwarding-class fc0
                    drop-probability high
                }
                dot1p 7 {
                    forwarding-class fc7
                    drop-probability low
                }
            }
        }
    }
```

7.1.2 Applying a dot1p policy to a subinterface

Procedure

The following example applies a dot1p policy to inbound traffic on a subinterface:

Example

```
# info interface ethernet-1/1 subinterface 1 qos input classifiers
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                input {
                    classifiers {
                        dot1p-policy new-dot1p-policy
                    }
                }
            }
        }
    }
```

7.2 Dot1p marking

Dot1p marking refers to rewriting of the PCP value in the outermost VLAN tag. The node rewrites the value in the PCP field before a packet is transmitted out an egress interface. Downstream nodes handle the remarked traffic based on the updated code point. SR Linux implements dot1p marking using dot1p rewrite policies.

Each dot1p rewrite policy contains up to eight mapping rules, and each rule associates one of the eight possible internal forwarding classes (fc0 to fc7) to a PCP value (0 to 7).

Unless explicitly configured otherwise, a new mapping rule applies to all drop-probability levels. You can optionally configure a mapping rule for a specific forwarding class and drop probability combination, as required.

For a dot1p rewrite policy to take effect, you must apply the policy to at least one subinterface. SR Linux supports rewrite policies on any bridged or routed subinterface of any Ethernet port or LAG. No limit exists on the number of subinterfaces that can apply the same policy.

When a dot1p rewrite policy is applied to a subinterface, if the forwarding class of an outgoing packet does not match any configured dot1p rewrite rule, all pushed 802.1Q VLAN tags on the outgoing frame are marked with a PCP value of 0.

If a dot1p rewrite policy is not applied to a subinterface:

- For a routed subinterface, the dot1p value is taken from the forwarding class
- For bridged subinterface, the dot1p value is 0

No default dot1p rewrite policy

No default dot1p rewrite policy exists.

Effect of dot1p rewrite policy

Table 7: Effect of dot1p rewrite policy in relation to other configuration

Egress sub-interface type	Egress packet type	Behavior with dot1p-policy applied to egress sub-interface	Behavior without dot1p-policy applied to egress sub-interface
Untagged	Any	None, can be blocked	—
Routed, single-tagged	Non-originated, forwarded IPv4/IPv6	Rewrites the PCP in the VLAN tag pushed at egress by the subinterface definition.	PCP = fc in the VLAN tag pushed at egress when no policy attached.
Routed, single-tagged	Self-originated IPv4/IPv6	None, ignored for self-originated traffic.	—
Routed, single-tagged	Non-originated, forwarded after VXLAN encapsulation, Ethernet untagged payload	PCP = fc in the VLAN tag pushed at egress by the subinterface definition.	PCP = fc in the VLAN tag pushed at egress when no policy attached.
Bridged, single-tagged	Non-originated, L2 forwarded	Rewrites the PCP in the VLAN tag pushed at egress by the subinterface definition. Does not modify the PCP of VLAN tags in the payload. A VLAN tag is carried as payload when the ingress subinterface is configured with vlan-tagging = true and vlan-id = any.	PCP = 0 in the VLAN tag pushed at egress when no policy attached. Does not modify the PCP of VLAN tags in the payload. A VLAN tag is carried as payload when the ingress subinterface is configured with vlan-tagging = true and vlan-id = any.
Bridged, single-tagged	Self-originated IPv4/IPv6	None, ignored for self-originated traffic.	—
Bridged, single-tagged	Non-originated, L3 forwarded (IRB)	Rewrites the PCP in the VLAN tag pushed at egress by the subinterface definition.	PCP = fc in the VLAN tag pushed at egress when no policy attached.
Bridged, single-tagged	Non-originated, forwarded after VXLAN decapsulation, Ethernet untagged payload	Rewrites the PCP in the VLAN tag pushed at egress by the subinterface definition.	PCP = fc in the VLAN tag pushed at egress when no policy attached.

7.2.1 Configuring dot1p rewrite rules for output traffic

Procedure

The following example creates a dot1p rewrite-rule policy:

Example

```
--{ candidate shared default }--[  ]--
# info qos rewrite-rules
    qos {
        rewrite-rules {
            dot1p-policy rewrite-dot1p-example {
                map fc0 {
                    dot1p 0
                }
                map fc1 {
                    dot1p 3
                    drop-probability low {
                        dot1p 1
                    }
                    drop-probability high {
                        dot1p 2
                    }
                }
                map fc3 {
                    dot1p 3
                }
                map fc7 {
                    dot1p 7
                }
            }
        }
    }
```

7.2.2 Applying a dot1p rewrite rule to a subinterface

Procedure

The following example applies a dot1p rewrite-rule policy to outbound traffic on a subinterface:

Example

```
# info interface ethernet-1/1 subinterface 1 qos output rewrite-rules
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                output {
                    rewrite-rules {
                        dot1p-policy rewrite-dot1p-example
                    }
                }
            }
        }
    }
```

8 Queue templates configuration

Queue templates are groups of configuration information that apply to a set of queues. On 7250 IXR systems, the controlled set of queues are VOQs; on 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 systems, the controlled set of queues are egress queues.

The maximum number of queue templates per system varies by platform. On 7250 IXR systems, the maximum is eight queue templates; on 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 systems, the maximum is 62 queue templates.

The following parameters are configurable inside a queue template:

- The MBS of each queue; this essentially defines the length of each queue. When the queue builds to the MBS level, further packets are dropped. Be aware that discards may occur before the queue reaches MBS (for example, resulting from shared buffer exhaustion, or from the effects of WRED slopes defined for the queue).
- WRED slopes that define probability curves for discarding packets as a function of (weighted) average queue depth. WRED slopes are not supported for multicast queues.
- ECN slopes that define probability curves for marking ECN-capable packets as having experienced congestion, instead of discarding them. ECN slopes are not supported for multicast queues.

If a queue (VOQ or egress queue) does not have a queue template binding, it inherits the settings of the default queue template. The default queue template has a platform-specific MBS default value, no defined queue utilization thresholds, no WRED slopes, and no ECN slopes. You cannot display the default queue template, but its effect is visible by reading the state of individual queues that lack a queue template binding.

8.1 Configuring queue templates

Procedure

The following example creates a queue template that you could use for any of the following:

- a set of VOQs on a 7250 IXR
- an egress queue on a 7220 IXR-D2, D3, and D5
- an egress queue on a 7220 IXR-H2 and H3

Example

```
--{ candidate shared default }--[  ]--
# info qos
 qos {
        queue-templates {
            queue-template wred-ecn-1 {
            }
        }
    }
```

Note: This example is only the starting point of a full configuration. Subsequent sections build on this example to create a full configuration.

8.2 Queue depth (maximum burst size)

In a queue-template, the **maximum-burst-size** parameter sets the maximum length of an egress queue or set of VOQs. The queue depth is also known as the Maximum Burst Size (MBS). You must set the **maximum-burst-size** parameter to a non-zero value to configure WRED slope and ECN slope parameters.

On the 7250 IXR, the **maximum-burst-size** parameter applies to a set of VOQs. If the parameter is not configured, or is set to 0, the effective MBS of these VOQs is 256MB.

On the 7220 IXR-D2, D3, and D5 or the 7220 IXR-H2 and H3, the **maximum-burst-size** parameter applies to a set of egress queues. If the parameter is not configured or is set to 0, the effective MBS of these egress queues is calculated based on a fair allocation algorithm. You can assign a non-zero MBS value to multicast queues, but Nokia does not recommend this configuration (especially if multicast traffic is being shaped by configuring **peak-rate-percent**), because it can lead to a shortage of multicast-related buffering resources on 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 systems.

8.2.1 Configuring queue depth (maximum burst size)

Procedure

The following example specifies the queue depth with a set **maximum-burst-size**:

Example

```
--{ candidate shared default }--[  ]--
# info qos queue-templates
    qos {
        queue-templates {
            queue-template wred-ecn-1 {
                queue-depth {
                    maximum-burst-size 20
                }
            }
        }
    }
```

8.3 WRED slope

In a queue template, you can configure WRED policies to handle congestion when queue space is depleted. Without WRED, when a queue reaches its maximum fill size, the queue discards any packets arriving at the queue (known as tail drop).

WRED policies manage queue depth. They help to prevent congestion by starting random discards when the queue reaches a user-configured threshold value. This avoids the impact of discarding all the new incoming packets. By starting random discards at this threshold, an end-system can adjust its sending rate to the available bandwidth.

The WRED curve algorithm is based on two user-configurable thresholds (**min-threshold-percent** and **max-threshold-percent**) and a discard probability factor (**max-probability**).

On the 7220 IXR-D2, D3, and D5 or the 7220 IXR-H2 and H3, you can configure a WRED slope to apply only to TCP or to non-TCP traffic. This can be useful because TCP has built-in mechanisms to adjust its sending rate in response to packet drops. TCP-based senders lower the packet transmission rate when some of the packets fail to reach the far end.

8.3.1 Configuring a WRED slope (7250 IXR)

Procedure

The following example specifies a WRED slope for low drop probability traffic flowing through a set of VOQs on a 7250 IXR. This WRED slope applies to both TCP and non-TCP traffic.

Example

```
--{ * candidate shared default }--[  ]--
# info qos
    qos {
        queue-templates {
            queue-template wred-ecn-1 {
                active-queue-management {
                    wred-slope all drop-probability low {
                        min-threshold-percent 10
                        max-threshold-percent 25
                        max-probability 50
                    }
                }
            }
        }
    }
```

8.3.2 Configuring a WRED slope (7220 IXR)

Procedure

The following example specifies a WRED slope for TCP traffic that is classified as low drop probability flowing through an egress queue on the 7220 IXR-D2, D3, and D5 or the 7220 IXR-H2 and H3.

Example

```
--{ * candidate shared default }--[  ]--
# info qos
    qos {
        queue-templates {
            queue-template wred-ecn-1 {
                active-queue-management {
                    wred-slope tcp drop-probability low {
                        min-threshold-percent 10
                        max-threshold-percent 25
                        max-probability 50
                    }
                }
            }
        }
```

```
    }
```

8.4 ECN slope

Some IP applications support the ECN mechanism. With ECN, IP packets originated by such applications are not discarded when they enter a congested queue; instead, they are marked in a special way. The marking uses the two ECN bits in the traffic class field of the IPv4 or IPv6 packet header. The receiver of IP packets marked as having experienced congestion can signal to the sender (through Layer 4 or higher protocols) that it should reduce its sending rate. The advantage of this feedback mechanism is that the sending rate can drop more gradually than the normal response of a TCP sender to packet discards. A more gradual back-off can result in higher effective throughput in the network.

An ECN slope is similar to a WRED slope. It is based on two user-configurable thresholds (**min-threshold-percent** and **max-threshold-percent**) and a marking probability factor (**max-probability**).

To use an ECN slope, you must configure **explicit-congestion-notification**.

8.4.1 Configuring an ECN slope (7250 IXR)

Procedure

On 7250 IXR systems, the configuration requires you to specify an ECN DSCP policy; this is the DSCP rewrite policy that is used when an ECN field rewrite must be performed. In addition, you can only have one ECN slope per queue and it applies to all drop-probability levels.

Example

The following example specifies an ECN slope applicable to a 7250 IXR system:

```
--{ candidate shared default }--[  ]--
# info qos
 qos {
        explicit-congestion-notification {
            ecn-dscp-policy normalize
        }
        queue-templates{
            queue-template wred-ecn-1 {
                queue-depth{
                    maximum-burst-size 20{
                }active-queue-management{
                    ecn-slope{
                        ecn-drop-probability all{
                        ecn-min-threshold-percent 50
                        ecn-max-threshold-percent 50
                        max-probability 100{
    }
```

8.4.2 Configuring an ECN slope (7220 IXR)

Procedure

On the 7220 IXR-D2, D3, and D5 or the 7220 IXR-H2 and H3, you can have one ECN slope per drop-probability level of traffic flowing through an egress queue.

Example

The following example specifies an ECN slope applicable to a 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 system:

```
--{ candidate shared default }--[  ]--
# info qos
    qos {
        explicit-congestion-notification {
        }
        queue-templates {
            queue-template 2 {
                queue-depth {
                    maximum-burst-size 100
                }
                active-queue-management {
                    ecn-slope high {
                        min-threshold-percent 0
                        max-threshold-percent 80
                        max-probability 90
                    }
                }
            }
        }
    }
```

8.5 Queue utilization thresholds

When a router receives a burst of traffic, and the incoming rate exceeds the available transmission rate, the router queues the excess traffic. If the burst lasts long enough, or it is followed by additional bursts, the queues may overflow, resulting in traffic loss.

To respond to onsets of congestion, you can subscribe to telemetry information that generates an event when specific queues exceed a specified occupancy level.

To assign a utilization threshold to a queue, you must apply a non-default queue template to the queue, and that queue template must specify a non-zero **high-threshold-bytes** value. When the utilization of the queue crosses the specified **high-threshold-bytes** value, a hardware interrupt is raised. XDP records the current system-time and clears the interrupt. In a scaled setup, XDP may take 10 to 15 ms to process and clear each interrupt, meaning multiple threshold crossings within a very short period of time across one or more queues using the same queue template may appear as only a single event in the telemetry stream. When the **high-threshold-bytes** value is 0, the functionality is disabled and no threshold events are generated for the queues covered by the queue template.

SR Linux supports queue utilization thresholds on 7250 IXR, 7220 IXR-D2 and D3, and 7220 IXR-H2 and H3 systems; however, the behavior varies by system.

Note: You can only configure queue utilization thresholds for unicast queues; multicast queues do not support queue utilization thresholds.

8.5.1 Configuring queue utilization thresholds on 7250 IXR systems

Procedure

On a 7250 IXR system, binding a queue template with a non-zero **high-threshold-bytes** value to an egress queue assigns that threshold value to all the VOQs that logically feed this egress queue.

You can configure each queue template that the system supports with a different **high-threshold-bytes** value as needed.

Example: Configuring high-threshold-bytes

The following example configures the **high-threshold-bytes** value to 256255:

```
--{ candidate shared default }--[  ]--
# qos queue-templates queue-template 2 queue-depth high-threshold-bytes 256255
--{ candidate shared default }--[  ]--
# commit stay
All changes have been committed. Starting new transaction.
```

Each configured threshold value is rounded up to the nearest multiple of 256 bytes, up to a maximum capped value of MBS. You can observe the rounding (on a per VOQ-set basis) using the **info from state interface queue-statistics queue** <*queue-name*> **virtual-output-queue queue-depth** output. (A VOQ-set consists of the VOQ for core 0 and the VOQ for core 1.)

Example: Rounding high-threshold-bytes

In the following example, the **high-threshold-bytes** value was configured to 256255, but is rounded to the lower 256000 value (that is, a multiple of 256 bytes):

```
--{ candidate shared default }--[  ]--
# info from state interface ethernet-2/1 queue-statistics queue unicast-0 virtual-output-
queue queue-depth
    interface ethernet-2/1 {
        queue-statistics {
            queue unicast-0 {
                virtual-output-queue {
                    queue-depth {
                        maximum-burst-size 1203200768
                        high-threshold-bytes 256000
                    }
                }
            }
        }
    }
```

Note: To support named queues, the **unicast-queue 0** parameter previously included in this example (in R22.6 and earlier) is now updated to **queue unicast-0** (in R22.11 and later).

The state tree maintains the time of the last threshold crossing in the **interface queue-statistics queue virtual-output-queue queue-depth** leaf. This represents the last time when either VOQ in the VOQ-set (core0/core1) exceeded the operational threshold. The value of this leaf is not cleared when you delete or modify the queue template that is bound to the queue/VOQs or the **high-threshold-bytes** configuration in the applied queue template.

8.5.2 Configuring queue utilization thresholds on 7220 IXR-D2 and D3 systems

Procedure

On 7220 IXR-D2 and D3 systems, binding a queue template with a non-zero **high-threshold-bytes** value to an egress queue causes that threshold value to be used for that specific queue, as long as it is a unicast queue. The configuration of this leaf is ignored when this queue template is attached to a multicast queue.

No more than seven different configured **high-threshold-bytes** values are allowed across all the queue templates used. The management server rejects a commit that would leave more than seven different values after all adds, deletes, and modifies are processed.

Example: Configuring high-threshold-bytes

The following example configures the **high-threshold-bytes** value to 2048999:

```
--{ candidate shared default }--[  ]--
A# qos queue-templates queue-template 2 queue-depth maximum-burst-size 2049024
 high-threshold-bytes 2048999
--{ candidate shared default }--[  ]--
# commit stay
All changes have been committed. Starting new transaction.
```

Each configured threshold value (that the management server accepts) is rounded up to the nearest multiple of 2048 bytes, up to a maximum capped value of MBS. For this reason, do not configure values that round to the same multiple of 2048 bytes. This causes duplication among the **high-threshold-bytes** values, of which only seven are allowed. You can display the effect of this rounding using the **info from state interface qos output unicast-queue queue-depth** command.

Example: Rounding high-threshold-bytes

In the following example, the **high-threshold-bytes** value was configured to *2048999*, but is rounded to a lower *2048000* value (that is, a multiple of 2048 bytes):

```
--{ candidate shared default }--[  ]--
A:# info from state interface ethernet-1/3 qos output unicast-queue 0 queue-
depth
    interface ethernet-1/3 {
        qos {
            output {
                unicast-queue 0 {
                    queue-depth {
                        maximum-burst-size 2049024
                        high-threshold-bytes 2048000
                    }
                }
            }
        }
    }
```

The state tree maintains the time of the last threshold crossing in the **interface qos output unicast-queue queue-depth last-high-threshold-time** leaf. This represents the last time the queue exceeded the operational threshold. The value of this leaf is not cleared when you delete or modify the queue template that is bound to the queue or the **high-threshold-bytes** configuration in the applied queue-template.

8.5.3 Configuring queue utilization thresholds on 7220 IXR-H2 and H3 systems

Procedure

On 7220 IXR-H2 and H3 systems, binding a queue template with a non-zero **high-threshold-bytes** value to an egress queue causes that threshold value to be used by each ITM that serves the queue. For a high-threshold event, the queue utilization threshold must be exceeded on either ITM.

No more than seven different configured **high-threshold-bytes** values are allowed across all the queue templates used. The management server rejects a commit that would leave more than seven different values after all adds, deletes, and modifies are processed.

Example: Configuring high-threshold-bytes

The following example configures the **high-threshold-bytes** value to 254255:

```
--{ candidate shared default }--[  ]--
A# qos queue-templates queue-template 2 queue-depth maximum-burst-size 2049024
 high-threshold-bytes 254255
--{ candidate shared default }--[  ]--
# commit stay
All changes have been committed. Starting new transaction.
```

Each configured threshold value (that the management server accepts) is rounded up to the nearest multiple of 254 bytes, up to a maximum capped value of MBS. For this reason, do not configure values that round to the same multiple of 254 bytes. This causes duplication among the **high-threshold-bytes** values, of which only seven are allowed. You can display the effect of this rounding using the **info from state interface qos output unicast-queuequeue-depth** command.

Example: Rounding high-threshold-bytes

In the following example, the **high-threshold-bytes** value was configured to 254255, but is rounded to a lower 254000 value (that is, a multiple of 254 bytes):

```
--{ candidate shared default }--[  ]--
A:# info from state interface ethernet-1/3 qos output unicast-queue 0 queue-
depth
    interface ethernet-1/3 {
        qos {
            output {
                unicast-queue 0 {
                    queue-depth {
                        maximum-burst-size 2049024
                        high-threshold-bytes 254000
                    }
                }
            }
        }
    }
```

The state tree maintains the time of the last threshold crossing in the **interface qos output unicast-queue queue-depth last-high-threshold-time** leaf. This represents the last time when either ITM exceeded the operational threshold. The value of this leaf is not cleared when you modify or delete the queue-template that is bound to the queue or the **high-threshold-bytes** configuration in the applied queue-template.

9 Named queues and forwarding classes

On 7250 IXR systems, SR Linux provides support for both named queues and named forwarding classes.

In R22.6 and earlier, forwarding classes had fixed names, `fc0` to `fc7`, and each queue of an egress port was distinguished by a simple numerical index `0` to `7` that mapped directly to those forwarding classes.

To align with OpenConfig, SR Linux R22.11 and later provides the flexibility to do the following:

- Assign each queue a string name
- Assign each forwarding class a string name and forwarding class index
- Map the named forwarding class to the named queue

Note: There are no changes to the factory default configuration.

Implementation details

The following implementation details apply to named queues and forwarding classes:

- Named queues and named forwarding classes are *not* automatically created under the /`qos` container configuration.
- Even though they do not appear as named forwarding classes in the configuration, the default forwarding class names `fc0` to `fc7` always exist and are reserved names.
- Even though they do not appear as named queues in the configuration, the default queue names `unicast-0` to `unicast-7` always exist and are reserved names. (On applicable platforms, default multicast queues are also reserved names.)
- Every interface always has a full set of egress queues; only the names of the queues are variable.
- When a named queue is associated with a forwarding class, the queue index value is determined by the `forwarding-class-index` value defined in the forwarding class configuration. The specified queue name then serves as an alias for the associated unicast or multicast queue index. For example, if `queue-X` is mapped to `forwarding-class-index 5` as a unicast output queue, then `queue-X` serves as the alias for queue `unicast-5`.
- If an interface has no explicit configuration for a default queue, and no named queue associated with that queue index, SR Linux displays the queue name in the output as the default value (`unicast-0` to `unicast-7`) with default parameters.
- If you configure a named forwarding class (for example, `forwarding-class-A`) and assign it a forwarding class index (for example, `forwarding-class-index 3`), any subsequent configuration that references the default forwarding class name (in this case, `fc3`) fails. You must always reference the named forwarding class when it is configured.

9.1 Configuring queue names

Procedure

On 7250 IXR systems, use the **qos queues queue** <*name*> command to assign a name to a queue. When you configure a named queue, it remains an inactive configuration that cannot be referenced by any interface until you map it to a forwarding class. The index value for the named queue is derived from the forwarding class mapping (based on the forwarding class index).

Example: Configure queue name (7250 IXR only)

```
# info qos queues
    qos {
        queues {
            queue test-queue-1 {
            }
        }
    }
```

9.2 Configuring forwarding class names and queue associations

Procedure

On 7250 IXR systems, use the **qos forwarding-classes forwarding-class** <*name*> command to assign a name, index value, and output queue to a forwarding class.

Forwarding classes with a higher forwarding class index are generally serviced more preferentially than forwarding classes with a lower forwarding class index (subject to scheduler configuration).

You must associate the forwarding class with a unicast queue and a multicast queue. All of the following parameters are mandatory: the **forwarding-class** *name*, the **forwarding-class-index**, the **unicast-output-queue**, and the **multicast-output-queue**.

Example: Configure forwarding class name, index, and queue association (7250 IXR only)

```
# info qos forwarding-classes
    qos {
        forwarding-classes {
            forwarding-class test-fc {
                forwarding-class-index 1
                output {
                    unicast-output-queue test-queue-1
                    multicast-output-queue test-queue-1
                }
            }
        }
    }
```

You can reference the named forwarding class in policies including DSCP classifier and rewrite, dot1p classifier and rewrite, multifield classifier and rewrite, MPLS traffic-class and rewrite, and the ingress subinterface policer template.

10 Output queue scheduling

Each unicast queue and each multicast queue of an egress port is associated with a scheduler node. The mapping of queues to scheduler nodes is platform-dependent and cannot be configured.

On 7250 IXR systems, there are two scheduling nodes per port; one for unicast traffic and one for multicast traffic. The two scheduling nodes have a WRR relationship, but the parameters cannot be adjusted. There is one PIR scheduling loop per scheduling node. The scheduling loop serves the strict priority classes first (in descending order of FC), followed by the WRR classes (by weight), limiting each forwarding class to its PIR (expressed as a percentage of the egress port bandwidth). By default, the PIR of each forwarding class is 100%. Note that multicast traffic handled by the multicast scheduler node is unscheduled and is not subject to the ingress VOQ buffering that applies to unicast traffic.

On 7220 IXR-D2 and D3 systems, the unicast queue and multicast queue for a particular forwarding class make up a queue pair. Each of the eight possible queue pairs of an egress port are associated with a scheduler node. Each scheduler node is served as strict priority (SP) or weighted round robin (WRR). If it is served as WRR, the scheduler node also has an associated weight. The scheduling loop serves the SP nodes first, followed by the WRR nodes by weight. The serving order of SP queues is in descending order of FC: fc7 first, then fc6, then fc5, and so on.

On 7220 IXR-H2 and H3 and 7220 IXR-D5 systems, there is a one-to-one mapping of queues to scheduler nodes. Each scheduler node can be served as SP or WRR. A WRR node has a configurable weight. The scheduling loop serves the SP nodes first, followed by the WRR nodes by weight. The serving order of SP queues is as follows:

- unicast queue 7 serving forwarding class index 7
- unicast queue 6 serving forwarding class index 6
- multicast queue 3 serving forwarding class index 6 and 7
- unicast queue 5 serving forwarding class index 5
- unicast queue 4 serving forwarding class index 4
- multicast queue 2 serving forwarding class index 4 and 5
- unicast queue 3 serving forwarding class index 3
- unicast queue 2 serving forwarding class index 2
- multicast queue 1 serving forwarding class index 2 and 3
- unicast queue 1 serving forwarding class index 1
- unicast queue 0 serving forwarding class index 0
- multicast queue 0 serving forwarding class index 0 and 1

Note: On the 7220 IXR-D5 systems, only unicast queues are supported.

10.1 Configuring strict priority (7250 IXR)

Procedure

The following example configures a queue or scheduler node for strict priority. When strict priority is set to false, the associated queue or scheduler node is configured as WRR. When strict priority is set to true, any configured weight is ignored.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        qos {
            output {
                queue unicast-0 {
                    scheduling {
                        strict-priority true
                    }
                }
            }
        }
    }
```

Note: To support named queues, the **unicast-queue 0** parameter previously included in this example (in R22.6 and earlier) is now updated to **queue unicast-0** (in R22.11 and later).

10.2 Configuring strict priority (7220 IXR)

Procedure

The following example configures a queue or scheduler node for strict priority on the 7220 IXR-D2, D3, and D5 or the 7220 IXR-H2 and H3. Note that when strict priority is set to false, the associated queue or scheduler node is configured as WRR. When strict priority is set to true, any configured weight is ignored.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        qos {
            output {
                scheduler {
                    tier 1 {
                        node 0 {
                            strict-priority true
                        }
                    }
                }
            }
        }
    }
```

10.3 Configuring WRR (7250 IXR)

Procedure

The following example configures a queue or scheduler-node for WRR. Queues or scheduler nodes that you do not configure with a specific weight have a weight of 1.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        qos {
            output {
                queue unicast-0 {
                    scheduling {
                        strict-priority false {
                            weight 20
                        }
                    }
                }
            }
        }
    }
```

Note: To support named queues, the **unicast-queue 0** parameter previously included in this example (in R22.6 and earlier) is now updated to **queue unicast-0** (in R22.11 and later).

10.4 Configuring WRR (7220 IXR)

Procedure

The following example configures a queue or scheduler node for WRR on the 7220 IXR-D2, D3, and D5 or the 7220 IXR-H2 and H3. Queues or scheduler nodes that you do not configure with a specific weight have a weight of 1.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        qos {
            output {
                scheduler {
                    tier 1 {
                        node 0 {
                            strict-priority false
                            weight 20
                        }
                    }
                }
            }
        }
    }
```

10.5 Configuring forwarding class peak rate

Procedure

The following examples set the maximum percentage of port bandwidth that is available to traffic of a particular FC. By default, traffic belonging to any FC can use up to 100% of the port bandwidth.

Example: Configure forwarding class peak rate (7220 IXR)

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        qos {
            output {
                unicast-queue 0 {
                    scheduling {
                        peak-rate-percent 75
                    }
                }
            }
        }
    }
```

Example: Configure forwarding class peak rate (7250 IXR)

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        qos {
            output {
                queue unicast-0 {
                    scheduling {
                        peak-rate-percent 75
                    }
                }
            }
        }
    }
```

Note: To support named queues, the **unicast-queue 0** parameter in this example is now updated to **queue unicast-0** (in R22.11 and later).

11 Ingress subinterface traffic policing

Some SR Linux-compatible hardware platforms (7220 IXR-D2/D3/D2L/D3L) support the ability to direct selected traffic flows to hardware policers. Traffic directed to a policer is metered to determine compliance with a traffic profile. At the output of the policer, every packet is marked with a color (green, yellow, or red) that represents whether it conforms, exceeds, or violates the traffic profile.

With a two-rate-three-color marker (RFC 2698), the traffic profile is defined using two traffic rates and their associated burst sizes:

- Committed information rate (CIR) and committed burst size (CBS)
- Peak information rate (PIR) and maximum burst size (MBS)

11.1 Token buckets

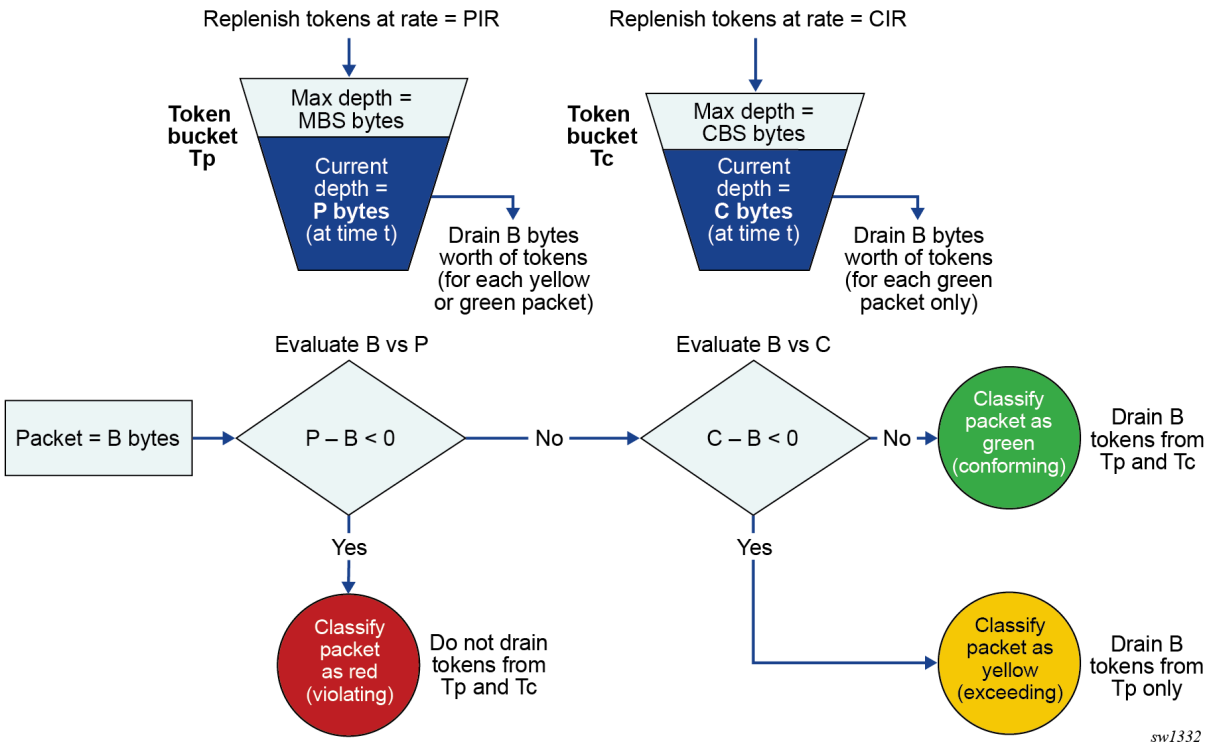
To determine compliance with the traffic profile, each policer uses two token buckets:

- **CIR bucket (Tc)**
  Tc has a fill rate equal to the CIR and a maximum depth of CBS bytes (with current depth at time t of C bytes).
- **PIR bucket (Tp)**
  Tp has a fill rate equal to the PIR and a maximum depth of MBS bytes (with current depth at time t of P bytes).

Initially (at time 0) the token buckets Tp and Tc are full, so that P=MBS and C=CBS. From then onwards, each bucket is continuously refilled at the rate of PIR and CIR.

The following diagram shows the token bucket process for each packet that arrives at the policer.

Figure 1: Token buckets (trTCM)



Each policer instance operates in a non-configurable color-aware mode. When a packet of size B bytes arrives at time t, the policer processes the packet as follows:

- If the packet is precolored as red or if P-B < 0, the packet is red (violating) and no tokens are drained from Tp or Tc. The policer either drops the packet or updates its drop probability as defined in the policer template (low, medium, or high).
- If the packet is precolored as yellow or if C-B < 0, the packet is yellow (exceeding), and B bytes are drained from Tp. The policer assigns the packet an updated drop probability as defined in the policer template (low, medium, or high).
- Otherwise, the packet is green (conforming), and B bytes are drained from Tp and Tc. The policer forwards the packet with no modifications, and drop probability remains unchanged (low).

Note: A drop probability of medium or high increases the chance that the packet is discarded (or ECN marked) when it enters the egress queue, if that egress queue has a WRED/ECN slope.

Pre-coloring based on drop probability

All packets arrive at the input of the policer with an assigned drop probability, based on the DSCP classifier policy. Each policer treats these packets as pre-colored as described in the preceding section. The following table describes the colors associated with each packet based on the drop probability at input.

Table 8: Drop probability to color mapping

Drop probability at input	Color associated at input
Low	Green (conforming)

Drop probability at input	Color associated at input
Medium	Yellow (exceeding)
High	Red (violating)

11.2 Policer template

To assign policers to subinterfaces, you must first configure policer templates. A policer template specifies a group of 1 to 32 policers, each with a specified sequence ID. Policers with lower sequence IDs are evaluated before policers with higher sequence IDs. You can configure each policer to match a forwarding class and optionally, forwarding type.

You can apply policer templates to the following subinterface types:

- Bridged subinterfaces of Ethernet ports or LAGs on a mac-vrf network-instance
- Routed subinterfaces of Ethernet ports or LAGs on either the default network-instance or an ip-vrf network-instance

Note:

- On routed subinterfaces, all traffic is considered to match the unicast forwarding type, even if it is received with a broadcast destination IP.
- Classification to forwarding class and drop probability occurs before policing in the ingress pipeline.
- There is no ingress policing of traffic of a particular forwarding class and forwarding type if that traffic has no match in the associated policer template.

IRB subinterface

Attachment of a policer template to an IRB subinterface is not currently supported.

Multiple subinterfaces referring to same policer template

Subinterfaces cannot share the same policer. If two or more different subinterfaces (routed or bridged) of the same port, same line card, or same chassis refer to the same policer template, each subinterface applies a separate instance of the template, consuming an equal number of TCAM entries.

Policing on LAG subinterfaces

Policers applied to subinterfaces are instantiated on each pipeline. The 7220 IXR-D2/D3/D2L/D3L each have two pipelines, with half the ports mapping to pipeline 0 and the other half of the ports mapping to pipeline 1. This impacts policing of ingress traffic on LAG subinterfaces as follows:

- The actual PIR for LAG subinterface traffic is N times the configured/quantized PIR, where N is the number of pipelines spanned by the LAG.
- The actual CIR for LAG subinterface traffic is N times the configured/quantized CIR, where N is the number of pipelines spanned by the LAG.

11.3 Policer statistics

For each policer template, you can choose between two statistics modes:

- **Violating-focus**

  Collects the number of:

  – Accepted (not dropped) packets and octets (counting all drop probabilities at policer output)

  – Violating packets and octets

- **Forwarding-focus**

  Collects the number of:

  – Committed packets and octets (conforming traffic only)

  – Accepted (not dropped) exceeding packets and octets

11.4 TCAM resources and scale

Note the following considerations related to traffic policers and Ternary Content Addressable Memory (TCAM) resources:

- If a policer template is configured on a subinterface, and any linecard supporting that subinterface cannot program all the TCAM rules of all the policers defined in that policer template, then policing is not activated on the subinterface. In this case, the **info from state** output for the subinterface shows no policer template bound to the subinterface.
- When a policer template bound to a subinterface is in a failed state due to TCAM resource exhaustion, all further configuration of the policer template will fail except for deletion of policers from the policer template and unbinding the policer template from a subinterface.

11.5 Configuring a subinterface traffic policer template

Procedure

To configure a policer template, use the **qos policer-templates** command.

Note: For PIR and CIR, the configured value and the operational value can differ as a result of rate quantization in the hardware. The actual operational PIR and CIR rates used in the hardware are available in the state representation of each policer instance created from the template, using the following command: **info from state interface** <*name*> **subinterface** <*index*> **qos input policers policer** <*sequence-id*>.

Example: Configure subinterface traffic policer

The following example configures a policer template containing one policer with sequence ID 100 that has a defined PIR, CIR, MBS, and CBS, and that matches unicast FC1 traffic. Yellow packets are marked with a drop-probability of medium, and red packets are dropped. The statistics-mode is set to violating-focus.

```
--{ candidate shared default }--[  ]--
# info qos
```

```
qos {
    policer-templates {
        policer-template test-policer-1 {
            statistics-mode violating-focus
            policer 100 {
                peak-rate-kbps 15000
                committed-rate-kbps 10000
                maximum-burst-size 100000
                committed-burst-size 20000
                forwarding-class fc1 {
                    forwarding-type [
                        unicast
                    ]
                }
                exceed-action {
                    drop-probability medium
                }
                violate-action {
                    drop
                }
            }
        }
    }
}
```

*Table 9: Parameters for **qos policer-templates***

Parameter	Definition
policer-template *<name>*	Assigns a name to the policer template.
statistics-mode {**violating-focus** | **forwarding-focus**}	(Optional) Defines the statistics mode (default: **violating-focus**).
policer *<sequence-id>*	Assigns the sequence ID for the policer.
peak-rate-kbps *<0 to 4294967295>*	Sets PIR in kbps. The minimum supported PIR is 8 kbps.
committed-rate-kbps: *<0 to 4294967295>*	Sets CIR in kbps. The minimum supported CIR is 8 kbps.
maximum-burst-size *<512 to 4294967295>*	Sets MBS in bytes (4294967295 bytes = 268 MB).
committed-burst-size *<512 to 4294967295>*	Sets CBS in bytes.(4294967295 bytes = 268 MB).
forwarding-class *<fc>* [**forwarding-type** {**broadcast**|**multicast**|**unicast**|**unknown-unicast**}]	(Optional) Matches the policer to the specified forwarding class and optionally, forwarding type. If no forwarding class is specified, all traffic is matched. If traffic of a specific forwarding class has no mapping in your policer-template, it is not policed at ingress. To match any traffic that is not explicitly mapped, include a policer with no forwarding class specified (for example, as the lowest-priority policer in the template). This ensures that the policer template matches all traffic at ingress.
exceed-action drop-probability {**high** | **low** | **medium**}	(Optional) Applies a drop-probability to yellow packets (default: **drop-probability medium**).

Parameter	Definition
violate-action {**drop** | **drop-probability** {**high** | **low** | **medium**}}	(Optional) Applies an action (drop packets or assign a drop-probability) to red packets (default: **drop-probability high**).

11.6 Assigning a traffic policer template to a subinterface

Procedure

The following example applies policer template 100 to subinterface 1/2.1

Example: Assign traffic policer template to a subinterface

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/2
    interface ethernet-1/2 {
        subinterface 1 {
            qos {
                input {
                    policers {
                        policer-template test-policer-1
                    }
                }
            }
        }
    }
```

11.7 Displaying subinterface traffic policer statistics

Procedure

The following example displays traffic policer statistics.

Example: Display traffic policer subinterface statistics

```
--{ candidate shared default }--[  ]--
# info from state interface ethernet-1/1 subinterface 1 qos input policers policer 1
 statistics
```

You can use the following options to narrow the scope of the statistics output.

- In violating-focus mode only:
  - **accepted-octets**
  - **accepted-packets**
  - **violating-octets**
  - **violating-packets**
- In forwarding-focus mode only:
  - **committed-octets**
  - **committed-packets**

- **exceeding-octets**
- **exceeding-packets**

11.8 Clearing subinterface traffic policer statistics

Procedure

You can reset the policer statistics counters for a subinterface.

Example: Reset all policer statistics counters on a subinterface

The following example resets all policer statistics counters on a subinterface:

```
--{ running }--[  ]--
# tools interface ethernet-1/1 subinterface 1 qos input policers clear
```

Example: Reset statistics counters for specific policer

The following example resets statistics counters for policer 10 on ethernet-1/1.1:

```
--{ running }--[  ]--
# tools interface ethernet-1/1 subinterface 1 qos input policers policer 10 clear
```

12 MPLS QoS overview

SR Linux supports QoS capabilities in MPLS networks using traffic classification and marking.

MPLS traffic classification and marking

SR Linux supports EXP-inferred LSPs as described in RFC 3270. This allows multiple classes of service to be transported by a single LSP, with the EXP marking of each packet determining the correct per-hop behavior (PHB) to apply to each router.

On SR Linux, the mapping between an EXP value and a PHB is provided by an MPLS traffic-class classifier policy. A single router can have one or more of these policies so that some subinterfaces can have one policy applied and other subinterfaces can have another policy applied. Each traffic-class classifier policy consists of multiple mapping entries, each of which maps one unique EXP value to a (forwarding class, drop probability) tuple.

SR Linux also supports MPLS traffic-class rewrite policies. If MPLS-encapsulated packets are transmitted out an egress subinterface with such a policy bound to it, the EXP field in all the pushed labels of these packets is based on the mapping rules of the policy. MPLS traffic-class rewrite rules associate a forwarding class or a (forwarding class, drop probability) tuple with an EXP rewrite value.

SR Linux does not support the short-pipe model of RFC 3270.

12.1 Ingress LER

When an SR Linux router that is acting as an ingress LER matches an IP packet to an LDP tunnel or a static MPLS forwarding entry, the following apply.

- The ingress LER determines the forwarding class and drop probability of the packet from the IP DSCP of the received unlabeled packet, based on the DSCP classifier policy applied to the ingress subinterface (or the default DSCP classifier policy if there is no explicit association). If an MPLS TC classifier policy is applied to the ingress subinterface, it has no effect.
- If a DSCP rewrite policy is applied to the egress subinterface, the IP header DSCP value is rewritten before the egress MPLS encapsulation is applied.
- If no MPLS TC rewrite policy is associated with the egress subinterface, EXP=0 is written into all pushed labels.
- If an MPLS TC rewrite policy is associated with the egress subinterface, and it matches the forwarding class (and possibly also the drop probability) of the packet, the EXP provided by the mapping rule is written into the EXP field of all pushed labels.
- If ECN is enabled globally, and the packet hits an ECN slope in a congested queue such that the ECN marking should be '11', the ECN field of the packet is modified accordingly, and the DSCP field is also remarked according to the ECN DSCP policy.

12.2 Transit LSR

When an SR Linux router that is acting as a transit LSR matches an MPLS packet to a swap ILM entry, the following apply.

- The transit LSR determines the forwarding class and drop probability of the packet from the EXP in the topmost label stack entry of the received labeled packet (before popping), based on the MPLS TC classifier policy applied to the ingress subinterface (or the default MPLS TC classifier policy, if there is no explicit association).
- If a DSCP classifier policy is applied to the ingress subinterface, it has no effect on the packet classification.
- If a DSCP rewrite policy is applied to the egress subinterface, it has no effect on the transmitted MPLS packet.
- If no MPLS TC rewrite policy is associated with the egress subinterface, the classified FC of the packet is written as a value 0 to 7 into the EXP field of all pushed labels. This does not guarantee that the EXP of the popped labels matches the EXP of the pushed labels (that is, if a non-default MPLS TC classifier policy is applied to the ingress subinterface).
- If an MPLS TC rewrite policy is associated with the egress subinterface, and it matches the forwarding class (and possibly also the drop probability) of the packet, the EXP provided by the mapping rule is written into the EXP field of all pushed labels.
- If ECN is enabled globally, it has no effect on the MPLS packet. The MPLS packet is considered non-ECT capable, even if the buried IP ECN bits indicate otherwise. The IP ECN field is not modified.

12.3 PHP LSR

When an SR Linux router that is acting as a PHP LSR matches an MPLS packet to a pop and swap-to-implicit-null ILM entry, the following apply.

- The PHP LSR determines the forwarding class and drop probability of the packet from the EXP in the topmost label stack entry of the received labeled packet (before popping), based on the MPLS TC classifier policy applied to the ingress subinterface (or the default MPLS TC classifier policy, if there is no explicit association). If a DSCP classifier policy is applied to the ingress subinterface, it has no effect on the classification of the packet.
- If an MPLS TC rewrite policy is applied to the egress subinterface, it has no effect on the transmitted IP packet.
- If no DSCP rewrite policy is associated with the egress subinterface, the DSCP field of the IP payload packet is transmitted unchanged. There is no attempt to copy the EXP field into the IP DSCP of the IP payload packet.
- If a DSCP rewrite policy is associated with the egress subinterface, and it matches the forwarding class (and possibly also the drop probability) of the packet, the DSCP provided by the mapping rule is written (as an override) into the DSCP field in the transmitted IP packet. This is consistent with the uniform model of RFC 3270.
- If ECN is enabled globally, it has no effect on the PHP packet. The PHP packet is considered non-ECT capable even if the IP ECN bits indicate otherwise. The IP ECN field is not modified.

12.4 Egress LER

When an SR Linux router that is acting as an egress LER matches an MPLS packet to a pop ILM entry that leads to all labels being popped, the following apply.

- The egress LER determines the forwarding class and drop probability of the packet from the EXP in the topmost label stack entry of the received labeled packet (before popping), based on the mpls-tc classifier policy applied to the ingress subinterface (or the default mpls-tc classifier policy, if there is no explicit association).

  If a DSCP classifier policy is applied to the ingress subinterface, it has no effect on the classification of the packet.
- If an mpls-tc rewrite policy is applied to the egress subinterface, it has no effect on the transmitted IP packet.
- If no DSCP rewrite policy is associated with the egress subinterface, the DSCP field of the IP payload packet is transmitted unchanged. There is no attempt to copy the EXP field into the IP DSCP of the IP payload packet. This is consistent with the pipe model of RFC 3270.
- If a DSCP rewrite policy is associated with the egress subinterface, and it matches the forwarding class (and possibly also the drop probability) of the packet, the DSCP provided by the mapping rule is copied into the IP DSCP of the transmitted IP packet, overwriting the previous value. This is consistent with the uniform model of RFC 3270.
- If ECN is enabled globally, it has no effect on the terminating MPLS packet. The terminating packet is considered non-ECT capable even if the IP ECN bits indicate otherwise. The IP ECN field is not modified.

Note: Note that the DSCP marking of terminating MPLS traffic cannot be decoupled from the DSCP marking of transit IP traffic through the same egress subinterface.

12.5 Default MPLS traffic-class classifier policy

The following table shows the default MPLS TC classifier policy.

Table 10: Default MPLS TC classifier policy

Traffic class (EXP)	Forwarding class	Drop probability
0	0	low
1	1	low
2	2	low
3	3	low
4	4	low
5	5	low
6	6	low

Traffic class (EXP)	Forwarding class	Drop probability
7	7	low

13 MPLS QoS configuration

MPLS QoS configuration on SR Linux involves the following tasks:

- Configuring MPLS traffic-class policy
- Applying MPLS traffic-class policy to input traffic
- Configuring MPLS rewrite rules
- Applying MPLS rewrite rules to output traffic

13.1 Configuring MPLS traffic-class policy

Procedure

The following example creates an MPLS traffic-class policy:

Example

```
--{ candidate shared default }--[  ]--
# info qos classifiers
    qos {
        classifiers {
            mpls-traffic-class-policy mpls-policy-1 {
                traffic-class 7 {
                    forwarding-class fc7
                    drop-probability medium
                }
            }
        }
    }
```

13.2 Applying MPLS traffic-class policy to input traffic

Procedure

The following example applies an MPLS traffic-class policy to inbound traffic on a subinterface.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                input {
                    classifiers {
                        mpls-traffic-class mpls-policy-1
                    }
                }
            }
```

```
        }
    }
```

13.3 Configuring MPLS rewrite rules

Procedure

The following example creates an MPLS rewrite-rule policy:

Example

```
--{ candidate shared default }--[  ]--
# info qos rewrite-rules
    qos {
        rewrite-rules {
            mpls-traffic-class-policy mpls-rewrite-2 {
                map fc7 {
                    traffic-class 7
                }
            }
        }
    }
```

13.4 Applying MPLS rewrite rules to output traffic

Procedure

The following example applies a rewrite-rule policy to outbound traffic on a subinterface.

Example

```
--{ candidate shared default }--[  ]--
# info interface ethernet-1/1
    interface ethernet-1/1 {
        subinterface 1 {
            qos {
                output {
                    rewrite-rules {
                        mpls-traffic-class mpls-rewrite-2
                    }
                }
            }
        }
    }
```

14 Buffer utilization display

The following table describes the buffer utilization differences between the 7250 IXR, 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3.

Table 11: Buffer utilization

Hardware	Buffer memory
7250 IXR	• SRAM size = 32MB • DRAM (HBM) size = 8 GB
7220 IXR-D2 and D3	• Total Buffer size = 32MB • Reserved Buffer size = 4.65MB
7220 IXR-D5	• Total Buffer size = 132MB • Reserved Buffer size = 3.7MB
7220 IXR-H2 and H3	• Total Buffer size = 64MB • Reserved Buffer size = 6.7MB

14.1 Displaying buffer utilization

Procedure

The following examples show overall buffer usage. The output varies depending on the hardware deployed.

Example: Displaying buffer utilization (7250 IXR)

```
# info from state platform linecard 1 forwarding-complex 0 buffer-memory
    platform {
        linecard 1 {
            forwarding-complex 0 {
                buffer-memory {
                    sram {
                        used 15808512 >> in bytes
                        free 17745920  >> in bytes
                    }
                    dram {
                        used 48   >>> it is in % of DRAM
                    }
                }
            }
        }
    }
```

Example: Displaying buffer utilization (7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3)

```
# info from state platform linecard 1 forwarding-complex 0 buffer-memory
    platform {
        linecard 1 {
            forwarding-complex 0 {
                buffer-memory {
                   used  2097152
                   free  27263246
                   reserved 4194034
                }
            }
        }
    }
```

15 Displaying QoS statistics

Procedure

To display traffic statistics for each output queue on an interface, use the **show interface** <*id*> **queue-detail** command in running or candidate mode.

The following example displays output queue statistics for an interface on a 7250 IXR system. The output on a 7220 IXR-D2, D3, and D5 or 7220 IXR-H2 and H3 system is similar, but shows slightly different information.

Example

```
# show interface ethernet-2/6 queue-detail
===============================================================================
Interface: ethernet-2/6
-------------------------------------------------------------------------------
  Description     : <None>
  Oper state      : up
  Last change     : 35s ago, No flaps since last clear
  Speed           : 100G
  Loopback mode   : false
  MTU             : 9232
  VLAN tagging    : false
  MAC address     : 12:12:02:FF:00:00
  Last stats clear: never
===============================================================================
Scheduler details for for ethernet-2/6
-------------------------------------------------------------------------------
Tier 1
Node   Scheduling   Weight                          Serving
0      SP           -        queue multicast-1, multicast-2, multicast-3,
                             multicast-4, multicast-5, multicast-6, multicast-7,
                             openconfig-multicast-0
1      SP           -        queue openconfig-unicast-0, unicast-1, unicast-2,
                             unicast-3, unicast-4, unicast-5, unicast-6,
                             unicast-7
===============================================================================
Queue           : multicast-1
Forwarding class: fc1
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
```

```
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : multicast-2
Forwarding class: fc2
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : multicast-3
Forwarding class: fc3
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : multicast-4
Forwarding class: fc4
-------------------------------------------------------------------------------
```

```
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : multicast-5
Forwarding class: fc5
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : multicast-6
Forwarding class: fc6
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
```

```
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : multicast-7
Forwarding class: fc7
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : openconfig-multicast-0
Forwarding class: openconfig-fc0
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : -
Scheduler node : Tier 1, Node 0
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 0
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
```

```
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
===============================================================================
Queue           : openconfig-unicast-0
Queue template  : default
Forwarding class: openconfig-fc0
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 1
PIR (bps)      : 1000003000
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-         Max probability
                              threshold(%MBS)    threshold(%MBS)
1       all                   0                  0                 0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
VOQ Id         Fwd-         Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1        0/0/0              0/0/0              0/0/0              0/0/0
2        0/0/0              0/0/0              0/0/0              0/0/0
3        0/0/0              0/0/0              0/0/0              0/0/0
4        0/0/0              0/0/0              0/0/0              0/0/0
===============================================================================
Queue           : unicast-1
Queue template  : default
Forwarding class: fc1
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : 101370431000
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
```

```
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-         Max probability
                              threshold(%MBS)    threshold(%MBS)
1       all                   0                  0                 0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
VOQ Id         Fwd-         Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1        0/0/0              0/0/0              0/0/0              0/0/0
2        0/0/0              0/0/0              0/0/0              0/0/0
3        0/0/0              0/0/0              0/0/0              0/0/0
4        0/0/0              0/0/0              0/0/0              0/0/0
===============================================================================
Queue           : unicast-2
Queue template  : default
Forwarding class: fc2
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : 101370431000
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-         Max probability
                              threshold(%MBS)    threshold(%MBS)
1       all                   0                  0                 0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
```

```
-------------------------------------------------------------------------------
VOQ Id         Fwd-         Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1        0/0/0              0/0/0             0/0/0              0/0/0
2        0/0/0              0/0/0             0/0/0              0/0/0
3        0/0/0              0/0/0             0/0/0              0/0/0
4        0/0/0              0/0/0             0/0/0              0/0/0
===============================================================================
Queue           : unicast-3
Queue template  : default
Forwarding class: fc3
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : 101370431000
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-         Max probability
                             threshold(%MBS)    threshold(%MBS)
1       all                 0                  0                  0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
VOQ Id         Fwd-         Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1        0/0/0              0/0/0             0/0/0              0/0/0
2        0/0/0              0/0/0             0/0/0              0/0/0
3        0/0/0              0/0/0             0/0/0              0/0/0
4        0/0/0              0/0/0             0/0/0              0/0/0
===============================================================================
Queue           : unicast-4
Queue template  : default
Forwarding class: fc4
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : 101370431000
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
```

```
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-          Max probability
                           threshold(%MBS)    threshold(%MBS)
1       all                0                  0                  0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
VOQ Id          Fwd-          Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1         0/0/0               0/0/0              0/0/0              0/0/0
2         0/0/0               0/0/0              0/0/0              0/0/0
3         0/0/0               0/0/0              0/0/0              0/0/0
4         0/0/0               0/0/0              0/0/0              0/0/0
===============================================================================
Queue           : unicast-5
Queue template  : default
Forwarding class: fc5
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : 101370431000
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-          Max probability
                           threshold(%MBS)    threshold(%MBS)
1       all                0                  0                  0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
```

```
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
VOQ Id         Fwd-         Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1        0/0/0              0/0/0              0/0/0              0/0/0
2        0/0/0              0/0/0              0/0/0              0/0/0
3        0/0/0              0/0/0              0/0/0              0/0/0
4        0/0/0              0/0/0              0/0/0              0/0/0
===============================================================================
Queue           : unicast-6
Queue template  : default
Forwarding class: fc6
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : 101370431000
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-          Max probability
                           threshold(%MBS)    threshold(%MBS)
1       all                0                  0                  0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
VOQ Id         Fwd-         Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1        0/0/0              0/0/0              0/0/0              0/0/0
2        0/0/0              0/0/0              0/0/0              0/0/0
3        0/0/0              0/0/0              0/0/0              0/0/0
4        0/0/0              0/0/0              0/0/0              0/0/0
===============================================================================
Queue           : unicast-7
Queue template  : default
Forwarding class: fc7
-------------------------------------------------------------------------------
Scheduling
-------------------------------------------------------------------------------
PIR (%)        : 100
PIR (bps)      : 101370431000
```

```
Strict Priority: true
Weight         : 1
Scheduler node : Tier 1, Node 1
-------------------------------------------------------------------------------
Queue Depth
-------------------------------------------------------------------------------
Maximum burst (bytes): 268435456
-------------------------------------------------------------------------------
Active WRED Slopes
-------------------------------------------------------------------------------
Slope   Traffic type       Drop       Min-threshol   Max-threshol       Max
                        probability      d(%MBS)        d(%MBS)      probability
1       all             low            0              0              0
2       all             medium         0              0              0
3       all             high           0              0              0
-------------------------------------------------------------------------------
Active ECN Slopes
-------------------------------------------------------------------------------
Slope   Drop probability         Min-               Max-         Max probability
                              threshold(%MBS)    threshold(%MBS)
1       all                   0                  0                  0
-------------------------------------------------------------------------------
Queue Statistics
-------------------------------------------------------------------------------
Tx Packets     : 0
Tx Bytes       : 0
Dropped Packets: 0
Dropped Bytes  : 0
-------------------------------------------------------------------------------
VOQ Id         Fwd-         Fwd-Pkts(L/M/H)        Drop-              Drop-
          Octets(L/M/H)                          Octets(L/M/H)       Pkts(L/M/H)
1        0/0/0              0/0/0              0/0/0              0/0/0
2        0/0/0              0/0/0              0/0/0              0/0/0
3        0/0/0              0/0/0              0/0/0              0/0/0
4        0/0/0              0/0/0              0/0/0              0/0/0
===============================================================================
--{ + candidate shared default }--[  ]--
A:rifa#
```

15.1 Clearing QoS statistics

Procedure

You can reset the queue statistics counters for an interface.

Example: Reset all statistics counters on an interface

The following example resets all statistics counters on an interface:

```
--{ running }--[  ]--
# tools interface ethernet-1/1 statistics queue-statistics clear
```

Example: Reset statistics counters for multicast egress queue

The following example resets statistics counters for a specified egress queue (multicast) on an interface:

```
--{ running }--[  ]--
# tools interface ethernet-1/1 statistics queue-statistics multicast-queue 1 clear
```

15.2 QoS profile resource usage

A QoS profile resource refers to the number of classifier and rewrite policies that are applied to interfaces on a line card. Each classifier or rewrite policy that is applied to an interface on a line card counts as one profile resource used.

For example, if you create classifier policy `dscp1` and apply it to input IPv4 traffic on an interface, and apply the same `dscp1` policy to input IPv6 traffic on a different interface on the same line card, it counts as two classifier profile resources used.

The SR Linux supports up to 15 classifier profile resources and up to 32 rewrite profile resources per line card. You can display the number of QoS profile resources in use for each line card.

15.2.1 Displaying QoS profile resource usage on a 7250 IXR system

Procedure

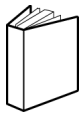
The following example displays the number of used and free classifier and rewrite profile resources for a line card:

Example

```
# info from state platform linecard 1 forwarding-complex 0 qos
    platform {
        linecard 1 {
            forwarding-complex 0 {
                qos {
                    resource classifier-profiles {
                        used 1
                        free 15

                    resource rewrite-profiles {
                        used 1
                        free 31
                    }
                }
            }
        }
    }
```

Customer document and product support



Customer documentation

Customer documentation welcome page

Technical support

Product support portal

Documentation feedback

Customer documentation feedback