RBFS HQoS Configuration Guide

Version 21.6.1, 25 June 2021

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

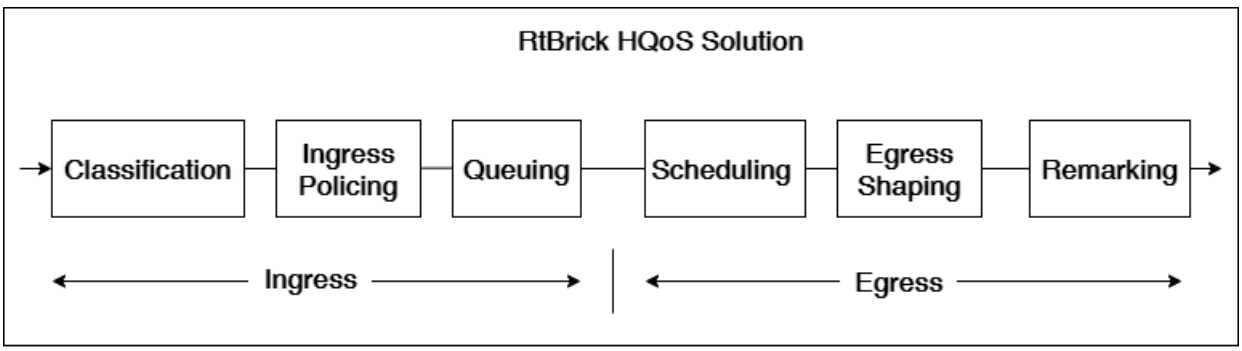
Table of Contents

1. Introduction to Hierarchical Quality of Service (HQoS)

Hierarchical Quality of Service (HQoS) is a technology that allows you to specify Quality of Service (QoS) behavior at multiple policy levels. It provides a high degree of granularity in traffic management. It can ensure that each network service gets the network resources it needs. This is achieved by classifying, policing, shaping, and scheduling the traffic based on service types. For example, in a simple QoS, you can differentiate between services (such as voice and video), but using H-QoS, you can apply QoS policies to different users, VLANs, maybe logical interfaces and so on.

The RtBrick Full Stack (RBFS) uses the following HQoS mechanisms:

- **Classifier:** Classifies each incoming packet as belonging to a specific class, based on packet contents. Supported classifiers are: Behavior Aggregate (BA) and Multifield (MF). In BA classifier, packets are classified according to the CoS field: IEEE 802.1p, IPv4/v6 ToS/TC, or MPLS EXP. In MF classifier, packets are classified using additional fields in IP header: source IPv4/IPv6 prefix, destination IPv4/IPv6 prefix, L4 source port, L4 destination port, and/or IP protocol.
- **Queuing:** Drop unqualified packets in advance using the Weighted Random Early Detection (WRED) technology in the case of congestion to ensure bandwidth for qualified services. This is performed at the egress.
- **Scheduler:** Manage traffic on a device using different algorithms for queue scheduling. Such algorithms include Fair Queuing (FQ), Weighted Round Robin (WRR), and Strict Priority (SP).
- **Policer:** Policer is implemented in the ingress to drop the unwanted traffic. Policer supports Committed Information Rate (CIR), the Committed Burst Size (CBS), Peak Information Rate (PIR), and Peak Burst Size (PBS). Drop behavior is to either mark traffic as green, yellow, or drop.
- **Shaper:** Shaper is implemented in egress to rate-limit the traffic.
- **Remarking:** Remarking allows you to rewrite the outgoing packet's codepoint. Remarking can be performed in the ingress or the egress side of the hardware pipeline.



1.1. MPLS HQoS

The MPLS HQoS has both UNIFORM and PIPE modes. These modes provides the following functionality:

- During MPLS Encapsulation, MPLS Mode is UNIFORM. MSB 3-bits from 8-bits IPv4-ToS or IPv6-TC are copied to the EXP bits of the newly added MPLS header(s).
- During MPLS Decapsulation, MPLS Mode is PIPE. 8-bits IPv4-ToS or IPv6-TC will be retained and hence it provides ToS/TC codepoint transparency.

For the Uniform MPLS mode mapping between IPv4-ToS or IPv6-TC to MPLS-EXP see the table below:

IPv4-TOS / IPv6-TC	EXP	DSCP
0-31	0	0-7
32-63	1	8-15
64-95	2	16-23
96-127	3	24-31
128-159	4	32-39
160-191	5	40-47
192-223	6	48-55
224-255	7	56-63

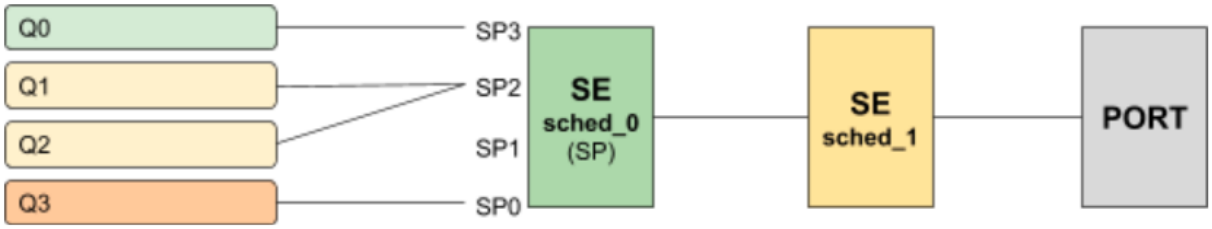
2. HQoS Features

This chapter explains the following topics:

2.1. Priority Propagation

Hierarchical QOS (HQoS) on RBFS is implemented by connecting or chaining queues to scheduler elements (Q —> SE), scheduler elements to each other (SE —> SE) and scheduler elements to ports (SE —> PORT). Each scheduler element can have different child connection points based on types described in section Scheduler.

This means that sched_0 in the example below is not scheduling between the attached queues, but between the different child connection points SP0 to SP3. The scheduler element sched_0 cannot differentiate between Q1 and Q2 in this example because both are connected to SP2.



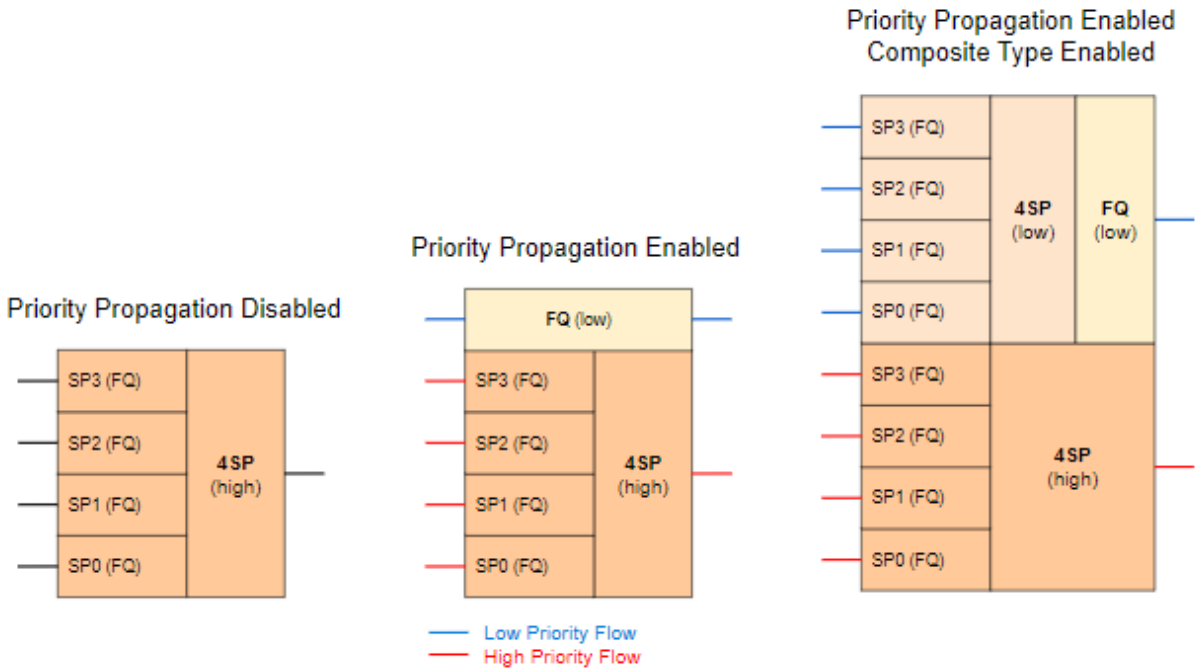
Without priority propagation each scheduler element can have multiple child

connection points but just one parent connection point. Therefore traffic leaving a scheduler element can't be differentiated by the parent scheduling element. The parent scheduler element sched_1 receives the traffic from sched_0 on the selected child connection point. As already mentioned scheduling within a scheduler element happens between child connection points. Second, a scheduler element has only one parent connection point which can be connected to a child connection point of another scheduler element (output of sched_0 input of sched_1). This results into the situation that all traffic from this SE is handled equally regardless of the queue. This may lead into the dropped priority traffic like voice or control traffic in case of congestion in parent elements. For example, if sched_1 has a shaping rate lower than the one of sched_0, it will drop traffic unaware of its original priority.

This problem is addressed with priority propagation which is enabled by default.

With priority propagation the scheduler elements operate in a dual-flow mode with high and low priority flows. The credits generated from the physical interface will be consumed by all attached high priority flows first and only remaining credits will be available for low priority flows. In this mode an implicit FQ element is created for each scheduler element. All queues assigned to low priority flow will be attached to this element.

An additional composite option of the scheduler element allows also the differentiation between multiple low priority queues if required. This composite type is created implicitly and does not need to be configured.

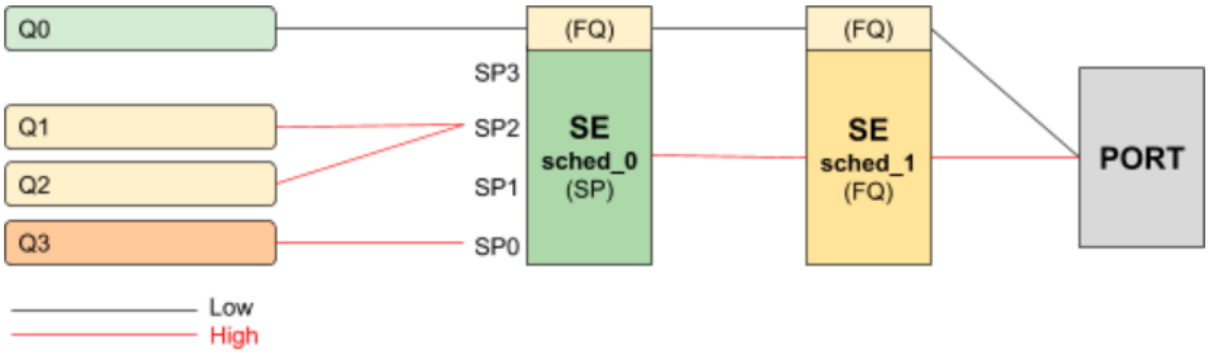


Without priority propagation enabled, each scheduler element consumes only one scheduler resource compared to two elements if enabled. The composite type consumes three scheduler elements.

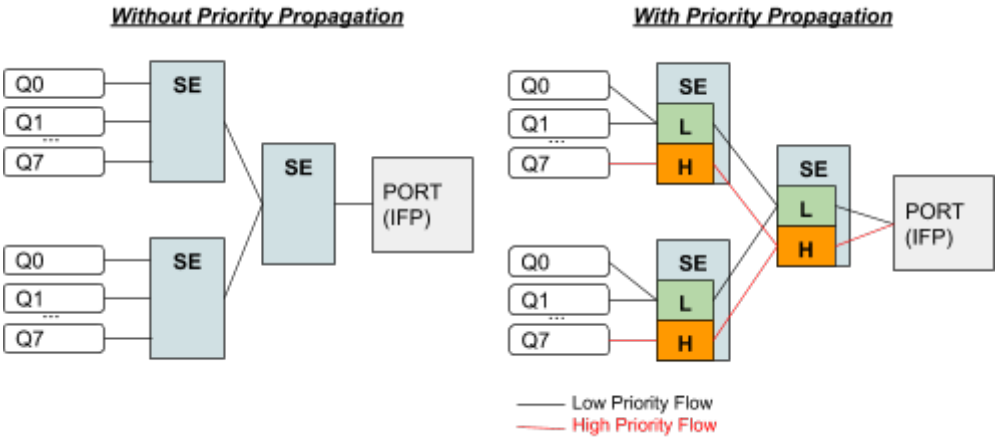
With priority propagation disabled, all traffic is considered as high priority flow.

Now for each queue we can select if connected to high priority or low priority flow where high priority flow is selected per default if not explicitly mentioned.

Assuming the example as before but with priority propagation and Q0 assigned to low priority flow and Q1 - Q3 assigned to high priority flow.



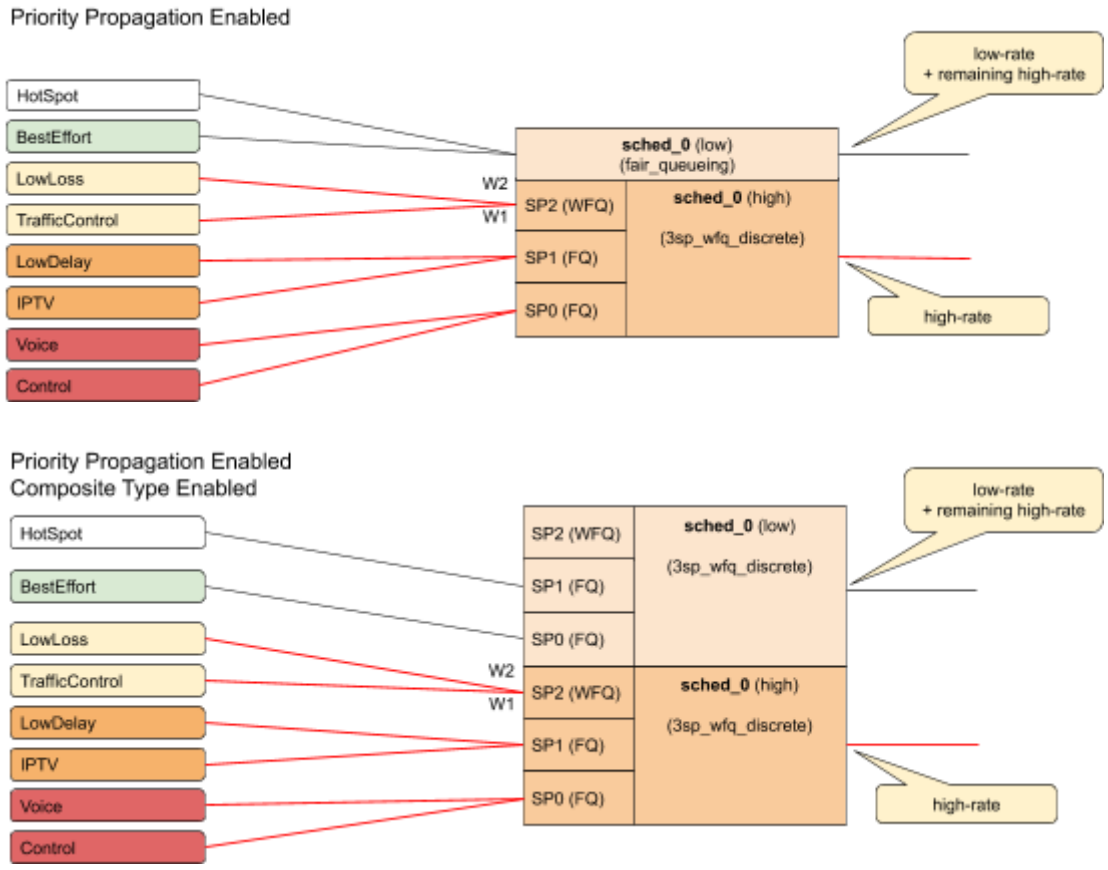
The figure below shows a typical multi level QoS configuration without priority propagation on the left and with priority propagation on the right side.



The credits generated from the physical interface will be consumed by high priority flow first and remaining credits will be available for low priority flow. The high flow traffic at any one element is scheduled based on type and connection point. Between schedulers it depends on how they are connected to the parent scheduling element. Per default all levels there is FQ for low and FQ for high priority flows. The port scheduler is also FQ.

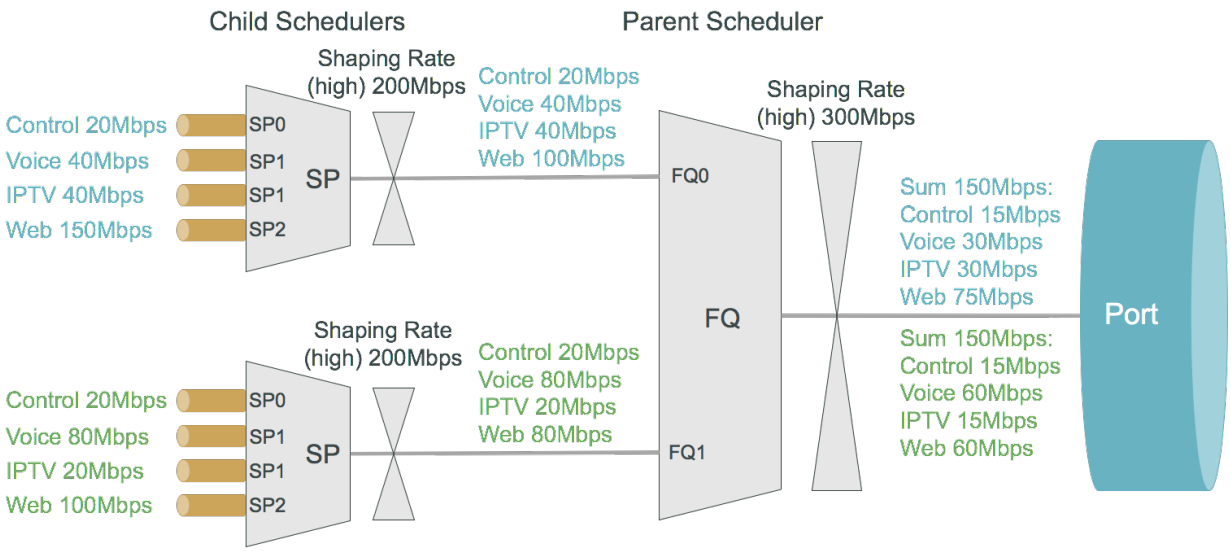
In this mode each shaper supports two different rates for low and high priority where the actual shaper rate is the sum of low and high priority rate. If low priority rate is zero, this flow is only served if high priority flow is not consuming all credits. An example might be a high rate of 9Mbps and low rate of 1m which results in max 10Mbps for low priority flow if high priority flow is not consuming any packets but at least 1m is ensured.

The following example shows a typical access service provider configuration with priority propagation enabled with and without composite type.

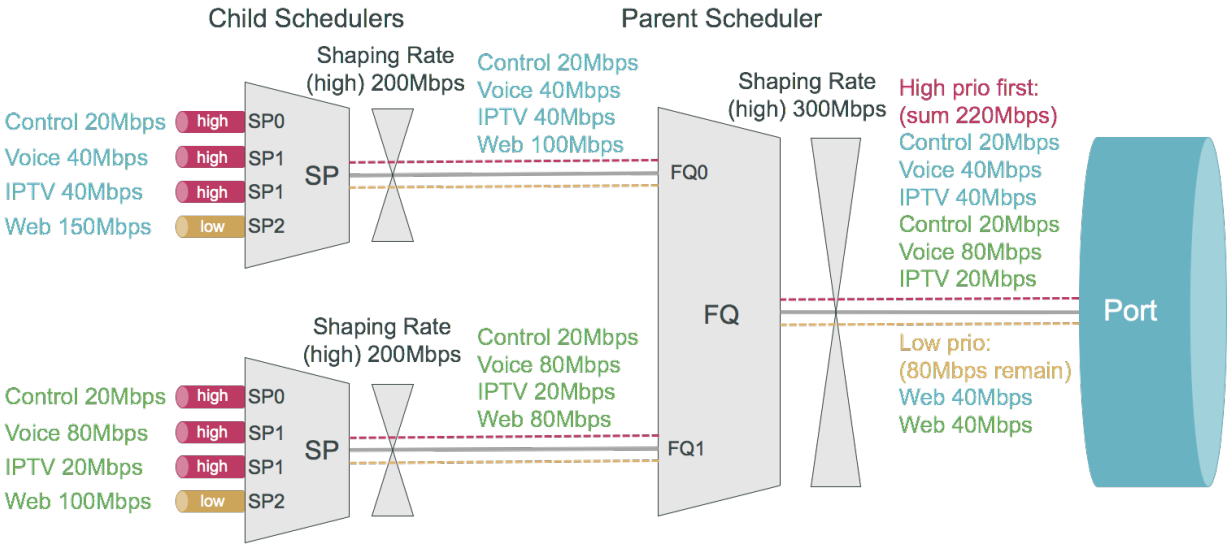


2.1.1. Simple Priority Propagation Scheduling Example

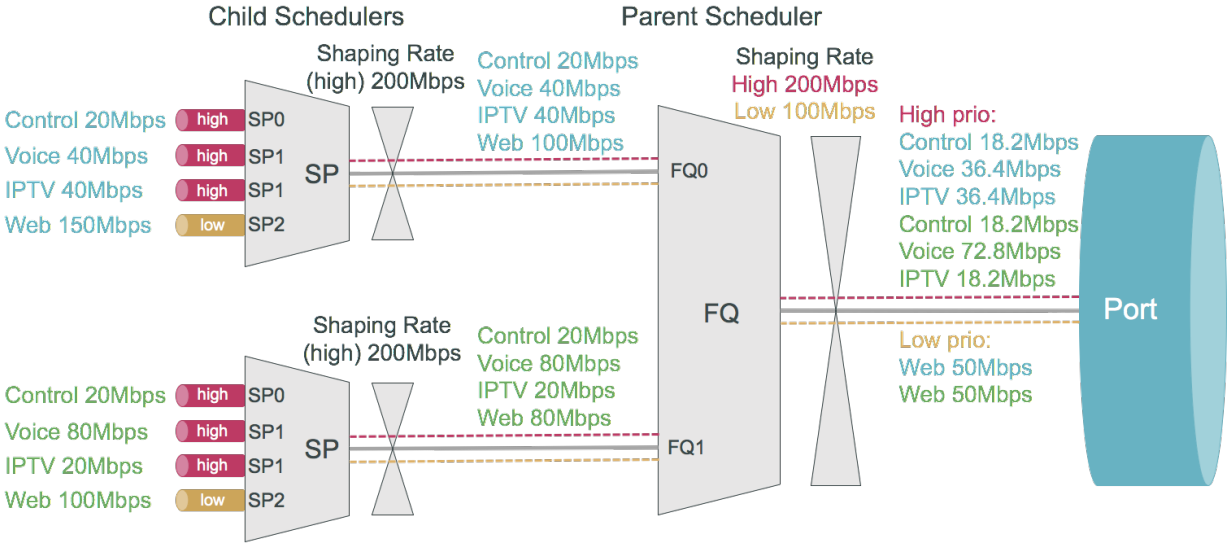
Without priority propagation, the parent scheduler drops traffic equally from all classes as it is unaware of priorities:



With priority propagation, the parent scheduler serves high priority flows first as shown in the figure below:



With priority propagation and dual-flow shaping, the parent scheduler serves high priority flows first up to the high flow shaping rate:



2.2. Behavior Aggregate (BA) Classifier

Classifiers assign the class to which a packet belongs. BA classification is performed on the ingress and maps incoming packet codepoint to a predefined class. BA Classification relies upon markings (that is, codepoint) placed in the headers of incoming packets:

- IEEE 802.1p: Priority - 3 bits
- IPv4: Type of Service byte (ToS) - 8 bits.
- IPv6: Traffic Class (TC) - 8 bits.
- MPLS: Experimental bits (EXP) - 3 bits.

- IEEE 802.1p and IPv4/IPv6 classifiers are applied on either Subscriber IFL or L3 IFL by attaching the classifier to a profile.
- MPLS Exp classifiers are applied either globally or per-instance (to support multiple VPN marking schemes) by attaching the classifier globally or to an instance.

Classifier configuration has the following guidelines and limitations:

- For IPv4: Only ToS based classification is possible. DSCP based classification is not possible.
- For IPv6: Only TC based classification is possible. DSCP based classification is not possible.
- For EXP classification, RBFS uses the uniform mode to copy MSB 3-bits from DSCP to EXP field at the time of MPLS encapsulation at the remote box.
- IPv4/IPv6 Classifiers do not match on labelled traffic. MPLS Classifier is required for the same.

- Default class for Queue or Policer is *class-0*. If for an incoming packet's *codepoint* there is no class mapping configured under a classifier, the packet will be classified as *class-0*.
- RBFS supports 8 **classes**: *class-0* to *class-7*.

2.2.1. Ingress Remarking

Ingress remarking is achieved by configuring the "remark-codepoint" field in the Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC field of the incoming packet at the ingress side with configured remark-codepoint. Note that the ingress remarking is not supported for BA Classifier with MPLS-EXP match-type.

2.3. Multifield (MF) Classifier

Multifield (MF) classifiers assign the class to which a packet belongs based on multiple fields. Unlike the BA classifier where only CoS fields are used for classification, MF classifier additionally uses the following fields:

- **class**: traffic class of the packet (class-0 to class-7) set by prior BA classifier
- **source prefix**: source IPv4 or IPv6 prefix
- **destination prefix**: destination IPv4 or IPv6 prefix
- **protocol**: UDP or TCP
- **source port**: UDP or TCP source port
- **destination port**: UDP or TCP destination port

- **qos markings**: IPv4 TOS or IPv6 TC header value

The actions supported by a multifield classifier are:

- **class**: traffic class to be set (class-0 to class-7)
- **Remark codepoint**: remark codepoint for ingress remarking

RBFS treats all the incoming IPv4-TOS or IPv6-TC qos field value in the incoming packet as untrusted. So a user is required to set action-remark-codepoint in the MF Classifier configuration to mark the QoS bits in the IP header of the outgoing packet. If action-remark-codepoint is not configured in the MF Classifier, default value 0 shall be marked in the packet.

The multifield classifiers can be bound globally (global.qos.global.config) or via QoS profile (global.qos.profile.config). The global multifield classifier applies to all traffic from any instance or interface. The multifield classifier assigned via QoS profile applies only to ingress traffic received on the interface where profile is bound to it.

The multifield classifier is processed after BA classification which allows it to match on selected class from BA classification or to change the assigned class by more granular match conditions. Both classification stages (BA and MF) are optional, they can be combined together or used alone controlled by configuration.

Multifield classifiers can't be bound to MPLS core interfaces. Therefore, the downstream traffic (from core to subscriber) should be classified via global multifield classifier, while upstream traffic (from subscriber to core) can be classified via multifield classifier from QoS profile which is instantiated per subscriber with an implicit match on ingress logical interface (InLIF).

- RBFS supports 8 **classes**: *class-0* to *class-7*.
- Per instance MF classifier for MPLS traffic is not supported in RBFS because of hardware limitations.
- The default class for Queue or Policer is **class-0**. If for an incoming packet, there is no MF classification configured, the packet will be classified as *class-0*.
- Priority 1 is reserved for BA Classifier ACL entries, therefore recommendation is to use Priority starting from 2 for MF Classifier
- If multiple ACL entries are hit in MF having the same priority, the result is unpredictable. So recommendation is to use different priorities for different ACL entries.

2.3.1. Match MPLS traffic

If MF Classifier is to be applied for MPLS traffic (that is, DOWNSTREAM traffic), match mpls traffic has to be configured in the MF ACL. If not configured, traffic may or may-not match the MF ACL entry in the h/w.

2.3.2. Ingress Remarking

Ingress remarking is achieved by configuring the “action remark-codepoint” in the MF Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC field of the incoming packet at the ingress side with configured remark-codepoint.

2.3.3. RADIUS Controlled Dynamic MF Classifier

As described for *RBFS RADIUS Services* document dynamic MF Classifier mapping is supported. The dynamic MF Classifier when configured override the MF Classifier mapped via QoS profile for the corresponding subscriber but not other subscribers.

2.4. Remarking

The packet markers set the codepoint in a packet to a particular value, adding the marked packet to a particular behavior aggregate. When the marker changes the codepoint in a packet, it "remarks" the packet. The codepoint in a packet can be IPv4-ToS, IPv6-TC, MPLS-EXP, or IEEE 802.1p field.

The following remarking options are supported in RBFS:

- IEEE 802.1p : Priority - 3 bits.
- IPv4: Type of Service byte (ToS) - 8 bits.
- IPv6: Traffic Class (TC) - 8 bits.
- MPLS-IPv4: MPLS Experimental bits (EXP) - 3 bits.
- MPLS-IPv6: MPLS Experimental bits (EXP) - 3 bits.

IPv4/v6 and IEEE 802.1p remark-map are applied on an interface - subscriber-ifl or l3ifl using Profile Name.

MPLS-IPv4/v6 remark-map is applied either globally or per-instance (to support multiple VPN marking schemes) using Remark-Map Name.

In RBFS remarking can be performed at the ingress or egress:

- **Ingress remarking** is achieved by configuring the **remark-codepoint** field in the Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC at the ingress side with configured remark-codepoint. The configured remark-codepoint can be modified again at the egress side using remark-map. The ingress remarking

is supported for IPv4, IPv6, and IEEE 802.1p BA classifiers.

- Egress remarking is achieved by configuring the **remark-map**. Remark Map is the mapping of **match-codepoint** and **color** to **remark-codepoint**. Egress remarking helps to remark the IPv4-ToS / IPv6-TC field in the IP header, or to write the EXP field in the MPLS label(s), or to write the IEEE 802.1p field in the VLAN header.

Here *Color* is used to set different *remark-codepoint* for same *match-codepoint* based on color marked by the Policer (i.e. *green* or *yellow*). Color is a mandatory field in remark-map. To set the same *remark-codepoint* for a *match-codepoint* irrespective of color, we have to set color as "*all*".

IPv4-ToS, IPv6-TC, or MPLS-EXP remarking:

- If the *remark-codepoint* is not configured in the BA Classifier or there is no hit in MF Classifier, match-codepoint in the remark-map is the ToS/TC value of the incoming IP packet.
- If the *remark-codepoint* is configured in the BA Classifier and there is no hit in the MF Classifier, match-codepoint is the same value as the remark-codepoint in the BA Classifier
- Irrespective of the *remark-codepoint* configured in the BA Classifier, if there is a hit in the MF Classifier the *match-codepoint* is the same value as the action remark-codepoint (0 if no action *remark-codepoint* configured) in the MF Classifier.

IEEE 802.1p VLAN remarking:

- In the current version of RBFS, match-codepoint in the VLAN remark-map is the Class derived at the ingress using MF or BA Classifier

Class-to-IP based Remark Map for L2TP UPSTREAM traffic is mapped globally. For more information, see the L2TP QoS section.

- In tunnel termination cases (i.e. Downstream traffic from core to Subscriber) the *remark-codepoint* in the MPLS BA Classifier is of no use. Therefore the *match_codepoint* in remark-map at the egress shall be the ToS/TC value of the incoming IP packet.
- In IP tunnel encapsulation cases (i.e. L2TP Upstream traffic from Subscriber to core) the remark-codepoint in the IPv4-TOS BA Classifier is of no use. Therefore the match_codepoint in class-to-ip remark-map at the egress shall be the Class derived from ingress BA Classifier.
- If no MPLS remarking is configured for the Upstream traffic, EXP bits in the MPLS header are derived from IP header TOS/TC bits using the Uniform MPLS mode.
- For VLAN: Only class to IEEE 802.1p remarking is supported.
- For IPv4: Only ToS based remarking is possible. DSCP based remarking is not possible.
- For IPv6: Only TC based remarking is possible. DSCP based remarking is not possible.

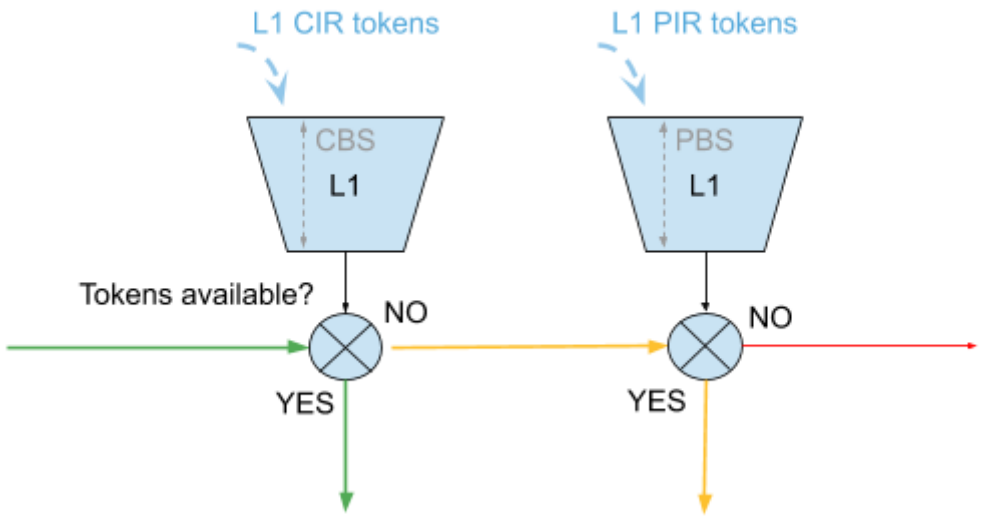
IPv4/v6 and IEEE 802.1p remark-map is applied on an interface - subscriber-ifl or l3ifl using Profile Name.

MPLS-IPv4/v6 remark-map is applied either globally or per-instance (to support multiple VPN marking schemes) using Remark-Map Name.

2.5. Policer

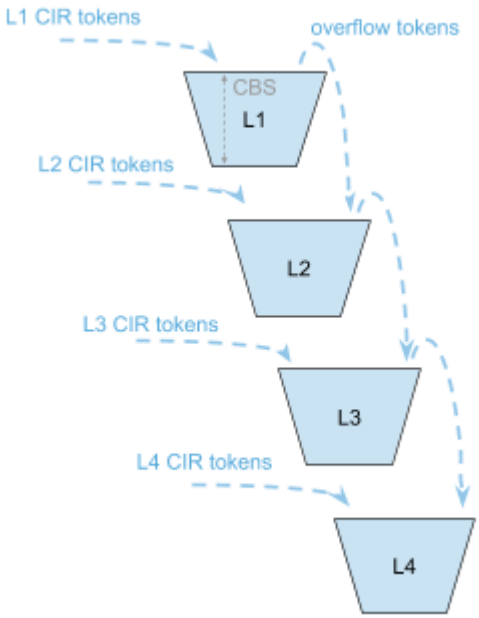
Policer defines the rate at which certain applications can access the hardware resource. So as to rate-limit the traffic from an application, policer hard-drops the unwanted packets in the ingress side.

In RBFS, policers support "**two-rate, three-color**" type in a 4-levels cascaded mode. This means that each policer level has two rates (CIR and PIR) and three colors (green, yellow and red) with two token buckets as shown below.



This means that traffic below CIR is marked green. Traffic above CIR but below PIR is yellow and above PIR is red. Traffic marked red will be dropped. Traffic marked yellow can be demoted by changing ToS, TC, or EXP using remark-map.

In 4 level cascade mode, unused tokens can be passed from higher priority levels to lower priorities where level 1 has highest and level 4 has the lowest priority as shown in the figure below.



Therefore a lower level configured with CIR 0 can still serve traffic if higher priority levels are not consuming all available tokens.

The available tokens per level are calculated by remaining CIR credits from upper levels and additional credits based on configured CIR per level. Per default the resulting tokens are not limited. The optional max CIR rate attribute allows to limit the sum of tokens from CIR and upper levels. Let's assume level 1 and 2 are both configured with a CIR of 2m. Without max CIR (default behaviour) level 2 can reach

up to 4m (level 1 CIR plus level 2 CIR). This can be limited by max CIR (for example, 3m). Obviously max CIR is not relevant for level 1.

Example

	CIR	RX	TX	CIR	RX	TX
L1	4m	1m	1m	4m	1m	1m
L2	6m	20m	9m	6m / max CIR 8	20m	8m
L3	0m	20m	0m	0m	20m	1m
L4	0m	20m	0m	0m	20m	0m
SUM	10m	61m	10m	10m	61m	10m

- Here m indicates mbps (Megabits per second)

In the columns 2 through 4 of the preceding example table, L1 consumes only 1m of the available 4m and passes the remaining 3m to L2 which adds additional 6m based on their own configured CIR resulting in 9m.

In the columns 5 through 7 of the preceding example table, L1 consumes only 1m of the available 4m and passes the remaining 3m to L2 which adds additional 6m based on their own configured CIR resulting in 9m. But because of the CIR limit set to 8m, only 8m of 9m can be used at this level. The remaining 1m is now passed to L3 which does not add additional CIR based credits. In both examples L4 would be able to reach up to 10m if upper levels are not consuming credits.

RADIUS Controlled Dynamic Policer

The RBFS RADIUS services support dynamic policer rate updates. The dynamic policer rate when configured affects only the QoS instance of the corresponding subscriber but not other subscribers.

2.5.1. Class-Policer-Map

Since RBFS supports up to 8 classes but only 4 policer levels, there is a need to map multiple classes to the same policer level. A *class-policer-map* defines such mappings. Using class-policer-map configuration, one can map any class to any supported policer level (that includes mapping multiple or all classes to the same level). Similar to policer, a class-policer-map is attached to a profile.

If class to level mapping is not configured, no policing will be applied to the traffic for that class.

2.6. Queueing

Queuing helps to drop the unwanted traffic in advance at the ingress side in case of congestion. This is to ensure bandwidth for qualified services.

RBFS supports the following queueing techniques:

- Tail Drop (TD): This is a conventional congestion avoidance technique. When the network is congested, drop subsequent packets from the queue.
- Weighted Random Early Detection (WRED): This technique requires configuring “Minimum Threshold”, “Maximum Threshold” and “Drop Probability”, which define the start and end range where packets may get discarded. When the average queue size is below the min threshold, no packets will be discarded. The drop_probability parameter can be used to specify the drop probability at the max threshold. When the average queue size is between the min and max threshold, the drop probability increases linearly from zero percent (at the min threshold) to drop_probability percent (at the max threshold). When the average queue size is greater than the max threshold, all packets are discarded.
  - When the average queue size is less than the “Minimum Threshold”, no packets will be discarded.
  - When the average queue size is greater than the “Maximum Threshold”, all packets are discarded.
  - When the average queue size is between “Minimum Threshold” and “Maximum Threshold”, the drop probability increases linearly from zero percent (at the min threshold) to drop probability (at the max threshold).

- Default queue within a queue group is the one mapped to *class-0*. If classification is not configured for an incoming packet’s codepoint, the packet will be classified as *class-0*. Thus will be mapped to queue mapped to *class-0* in *Class-Queue-Map*. For more information, see Class-Queue-Map.
- Maximum supported Queue size depends upon DRAM/OCB memory. Since OCB is external memory, hardware does not limit the size that can be configured per Queues.

2.6.1. RADIUS Controlled Dynamic Queue

As described for *RBFS RADIUS Services* document dynamic Queue buffer size updates are supported. The dynamic Queue buffer values when configured affect only the QoS instance of the corresponding subscriber but not other subscribers.

2.6.2. Class-Queue-Map

A class-queue-map defines the mapping of classes and queues. Class Queue Map is attached to a profile.

- You cannot map two classes to the same queue. The class to queue mapping is 1:1.
- If a queue group is created with four queues, only class-0 to class-3 can be mapped to the queues in class-queue-map, that is, class-4 to class-7 cannot be used.

2.6.3. Queue-Group

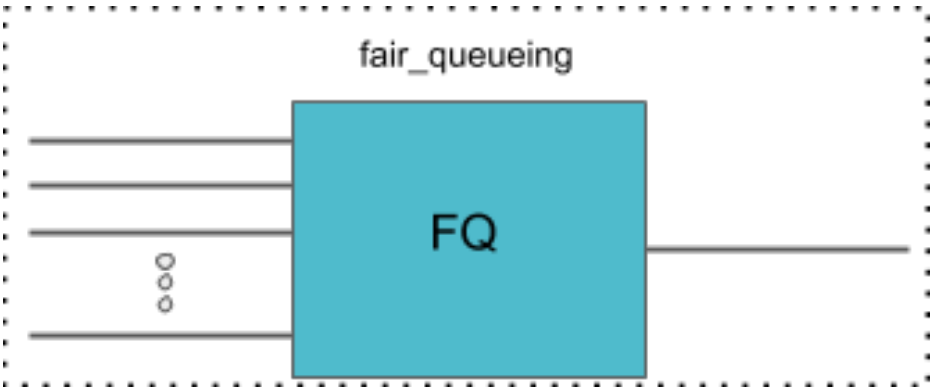
A Queue Group defines the Queue bundle. A Queue Group contains bundle of either 4 or 8 queues.

2.7. Scheduler

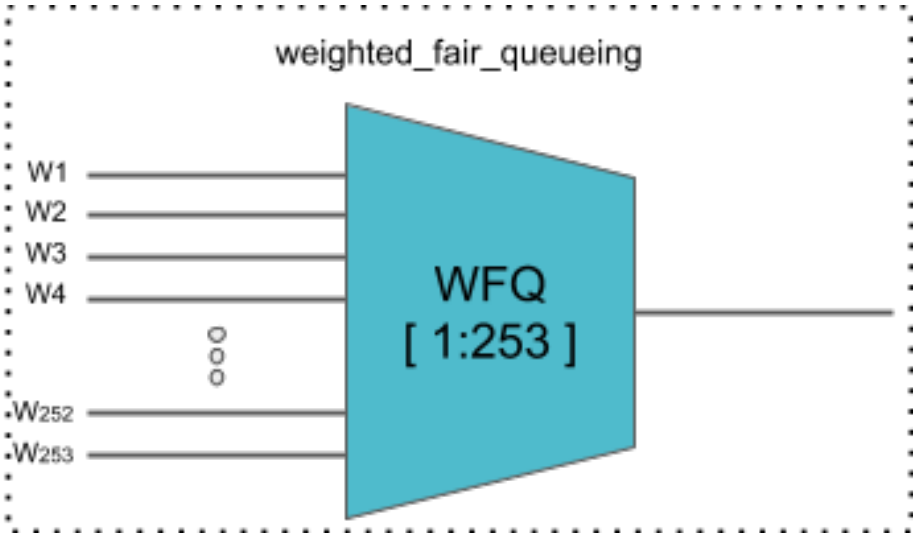
A scheduler configuration defines scheduler parameters such as type and shaping rate. The shaping rate defined for a scheduler applies to queue(s) associated with it.

The following scheduler types are supported:

- **Fair Queueing (FQ)**: Uses round-robin approach to select the next packet to service. This method ensures that all the flows are serviced equally. Configure scheduler type as *fair_queueing* to create FQ scheduler.



- **Weighted Fair Queueing (WFQ)**: Uses round-robin approach but with no guarantee of flow being serviced equally (like in FQ). The rotation of the next packet to service is based on the weight that is assigned to each flow. Configure scheduler type as *weighted_fair_queueing* to create WFQ scheduler.
  - Supported weight: 1 to 253



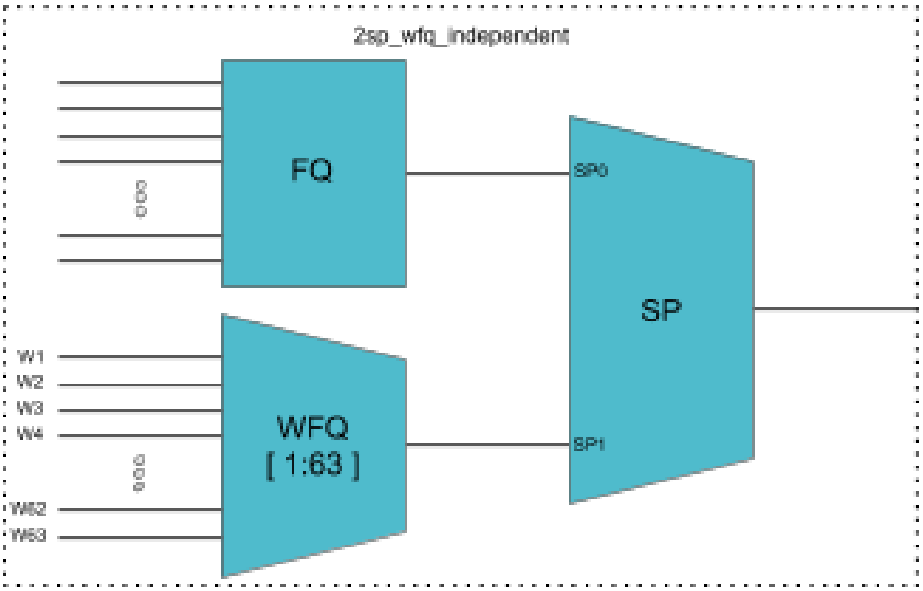
In any WFQ scheduler the lower the weight, the higher the bandwidth portion is awarded.

- **Strict Priority (SP)**: Uses priority based approach to service the flow. SP schedulers are supported in “hybrid” mode only. Hybrid mode combines FQ-WFQ schedulers using strict priority.

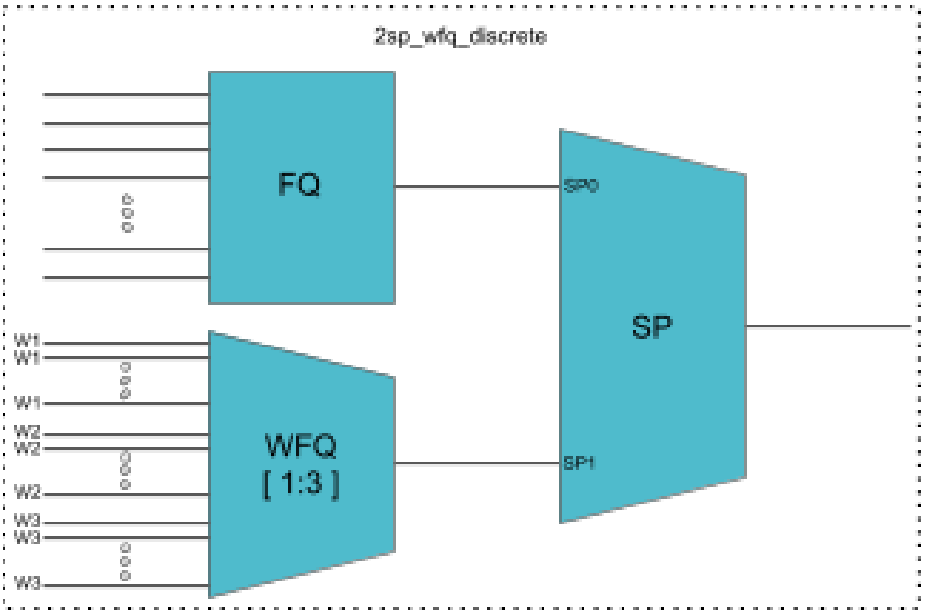
The priority order for SP is: **strict_priority_0 > strict_priority_1 > strict_priority_2 > strict_priority_3** (where **strict_priority_0** being highest priority and **strict_priority_3** being lowest)

The following SP scheduler types are supported:

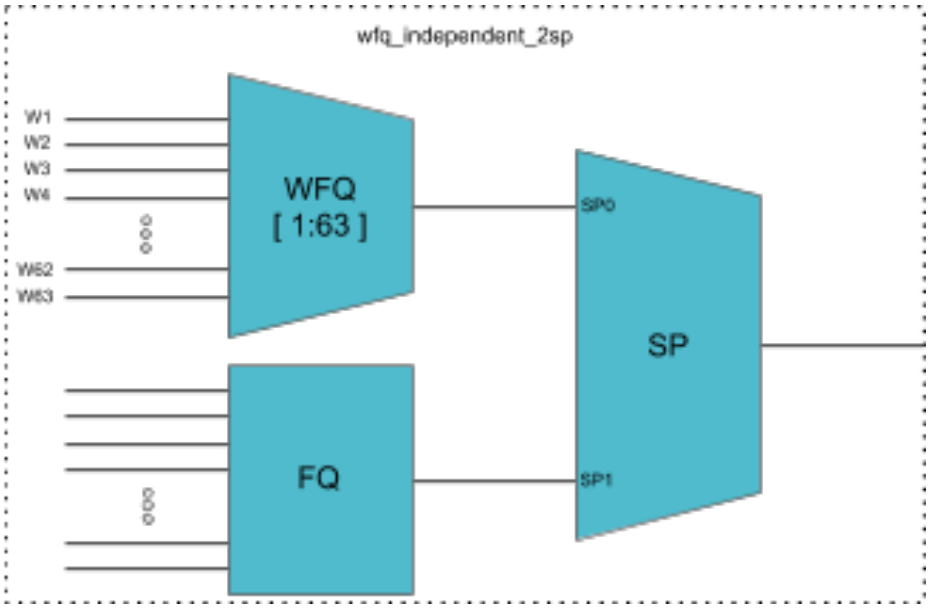
- **2 Strict Priority (2SP)**: Uses SP between 1-FQ and 1-WFQ. There are following types of 2SP hybrid schedulers:
  - type **“2sp_wfq_independent”**
    - Supported weight: 1 to 63



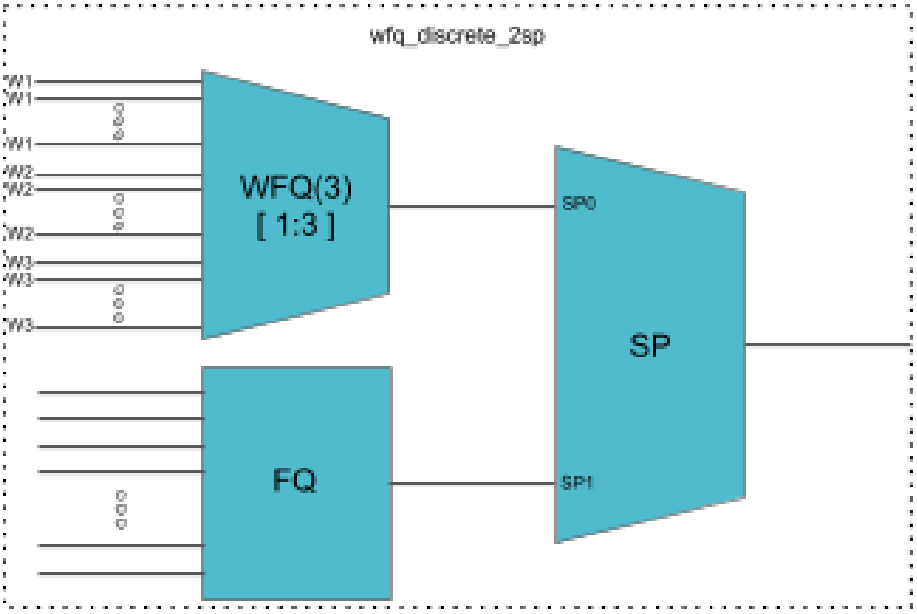
- type **“2sp_wfq_discrete”**
  - Supported weight: { 1, 2, 3 }



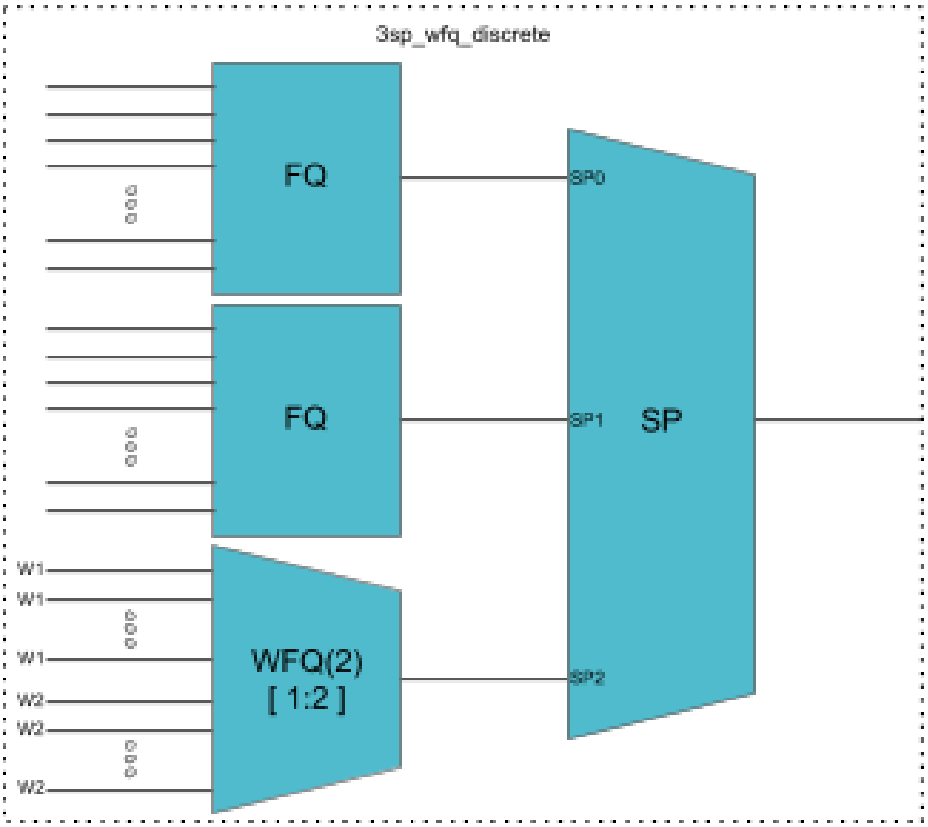
- type **“wfq_independent_2sp”**
  - Supported weight: 1 to 63



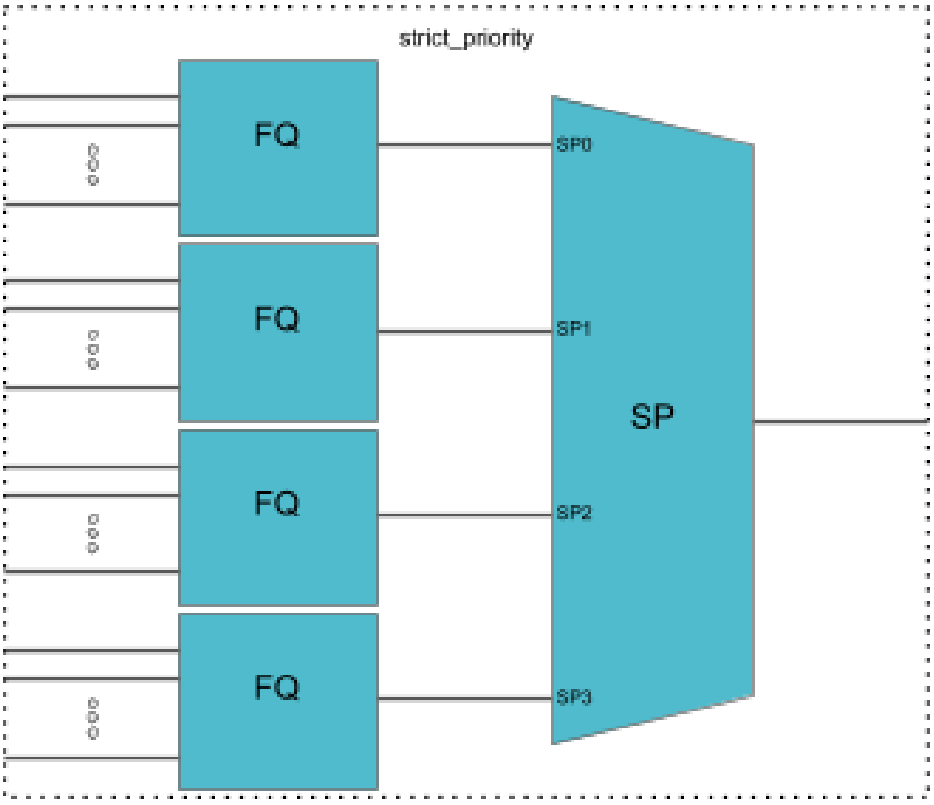
- type **“wfq_discrete_2sp”**
  - Supported weight: { 1, 2, 3 }



- **3 Strict Priority (3SP)**: maps 2-FQs and 1-WFQ
  - type: **“3sp_wfq_discrete“**
    - Supported weight: { 1, 2 }



- **4 Strict Priority (4SP)**: maps 4-FQs using SP
  - type **“strict_priority"**



2.8. Scheduler-Map

Scheduler Map defines the set of relationships between parents and children in egress scheduling hierarchy. A child in a Scheduler Map configuration can be either Queue or Scheduler. Whereas a parent in a Scheduler Map configuration can be either Port or Scheduler.

Connection Point and Weight

Child-queue or child-scheduler in a scheduler map configuration is connected to the parent-scheduler at "**connection point (CP)**". Connection point configuration also has "**weight**" associated with it if the parent has a WFQ scheduler corresponding to that connection point. Valid connection point value for a child to connect to parent **WFQ/FQ** scheduler is **no_priority** and to connect to parent **SP/Hybrid** scheduler is between **strict_priority_0** to **strict_priority_3** (based on number of Strict Priority points in parent scheduler).

Connection Types

There are five connection types in a scheduler map entry:

- queue_to_port
- queue_to_scheduler
- scheduler_to_scheduler
- scheduler_to_port

- For the **queue_to_port** connection type, the scheduler has no role.

2.9. Shaper

Shaper is used to rate-limit the traffic at the egress. In RBFS, shapers can be attached to both Queue and Scheduler.

A shaper configuration defines the shaping rate in kilo-bits-per-seconds (kbps).

Setting the shaping rate to 0 (zero) sets the rate to unlimited. Hence it is recommended to configure at-least 1 kbps so that shaping takes place.

RADIUS Controlled Dynamic Shapers

RBFS RADIUS services support dynamic shaper updates. The dynamic shaper when configured affects only the QoS instance of the corresponding subscriber but not other subscribers.

2.10. Profiles

A profile configuration defines the QoS profile that is attached to either a Subscriber interface or an L3 interface.

Profile maps the following QoS constructs to a Subscriber or an L3 interface:

- Classifier
- Multifield (MF) Classifier
- Class Policer Map
- Policer
- Class Queue Map
- Scheduler Map
- Remark Map

2.11. Header Compensation

2.11.1. Queue Compensation

The rate at which the packets are dequeued from a queue dependents on the credit received by that queue. The source of the credit received by a queue is the egress port to which the queue is mapped. When a packet is dequeued, the credit balance is decreased by the packet size. But, the packet size that is used must be adjusted to model the packet size at the egress, rather than its actual size at the ingress queue. Thus the header compensation is used to adjust for the differences in header size between ingress-queue and egress-port. RBFS supports static header compensation configuration per queue (in bytes).

2.11.2. Port Compensation

Similar to queue header compensation where header compensation is performed at the per-queue level, RBFS supports the following header compensation at the per-port level:

- **Ingress Header Compensation**: Inline with the header compensation option that we have per-queue, RBFS supports static header compensation configuration at the ingress to be used by the policing. Header compensation changes the effective size of the packet to compensate for change in header size (such as the CRC removal) when considering the packet for policing. Unlike queue, RBFS ingress header compensation configuration is per ingress port (in bytes).
- **Egress Header Compensation**: Inline with the header compensation option that we have per-queue or per-port at the ingress, RBFS supports static header compensation configuration at the egress. The egress header compensation

configuration is per egress port (in bytes).

The supported range for header compensation is -64 to +64 bytes.

2.12. L2TP QoS

The Layer 2 Tunneling Protocol (L2TP) QoS for upstream is similar to any other locally terminated subscriber. The QoS Profile is mapped dynamically via RADIUS for the L2TP subscribers.

The L2TP QoS for Downstream requires IPv4-TOS based BA Classifier which is mapped to L2TP Tunnel. The same can be achieved by attaching *l2tp-classifier-name* in *global QoS* configuration.

```
forwarding-options {
    class-of-service {
        global {
            l2tp-classifier-name l2tp-ip;
        }
    }
}
```

For Downstream Queueing, there is no change. Queueing is applied using QoS Profile similar to locally terminated Subscribers.

The following feaures are supported for L2TP QoS.

- Upstream
  - BA Classifier : IEEE 802.1p
  - Policing
    - Policer statistics
- Downstream
  - BA Classifier : IPv4-TOS
  - Queueing/Scheduling/Shaping
    - Queue statistics
  - Remark-Map : IEEE 802.1p (Class to VLAN priority remarking)

2.12.1. Guidelines

- To avoid control traffic policing/shaping, the assumption is that the IEEE 802.1p bits in Upstream or IPv4-TOS bits in Downstream will be different for control and data traffic, control traffic is expected to have the highest precedence.
  - Upstream classification is based on IEEE 802.1p bits.

- Downstream classification is based on IPv4-TOS bits of outer IP header.

2.13. Multi-level H-QoS : Level-1 to Level-5

The following HQoS levels are required to build internet access services like FTTH, FTTC, or FTTB:

Level-1 (IFP)

Physical Interface Shaper.

Level-2 (PON TREE)

Each PON tree is a TDM based shared medium with typically ~2.5 GBit/s (GPON) shared by up to 32 consumers (ONT or DPU).

Level-3 (DPU)

In case of FTTB there is a single DPU with multiple consumers via G.Fast DSL connected which requires an additional hierarchy. This level is not needed for FTTH or FTTC.

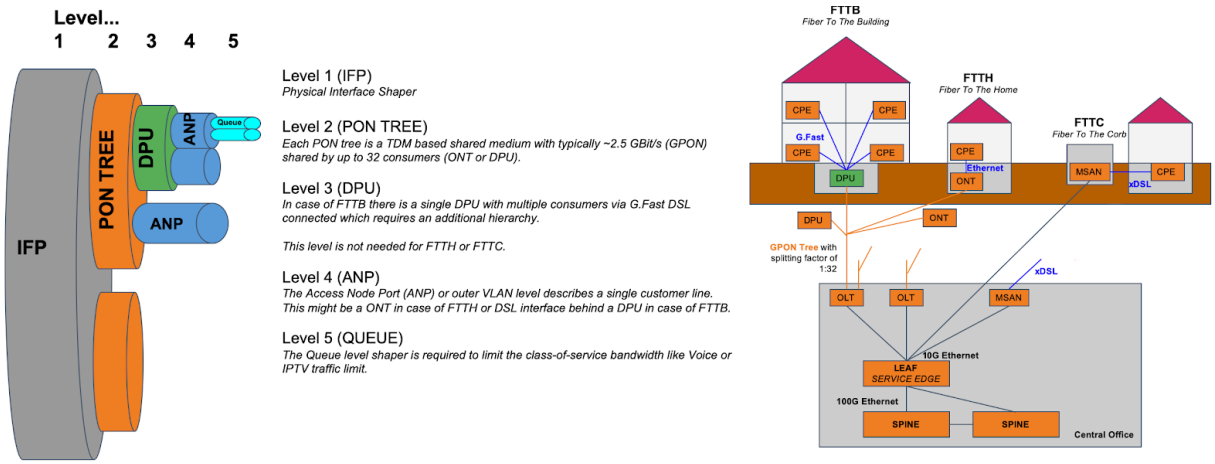
Level-4 (ANP or Session)

The Access Node Port (ANP) or outer VLAN level describes a single customer line. This might be an ONT in case of FTTH or DSL interface behind a DPU in case of FTTB. This level can be also represented on PPPoE sessions as long as just one session is permitted per VLAN.

Level-5 (QUEUE)

The Queue level shaper is required to limit the class-of-service bandwidth like Voice or IPTV traffic limit.

The figure below shows the diagram along with QoS representing Level-1 to Level-5 Hierarchical scheduling.



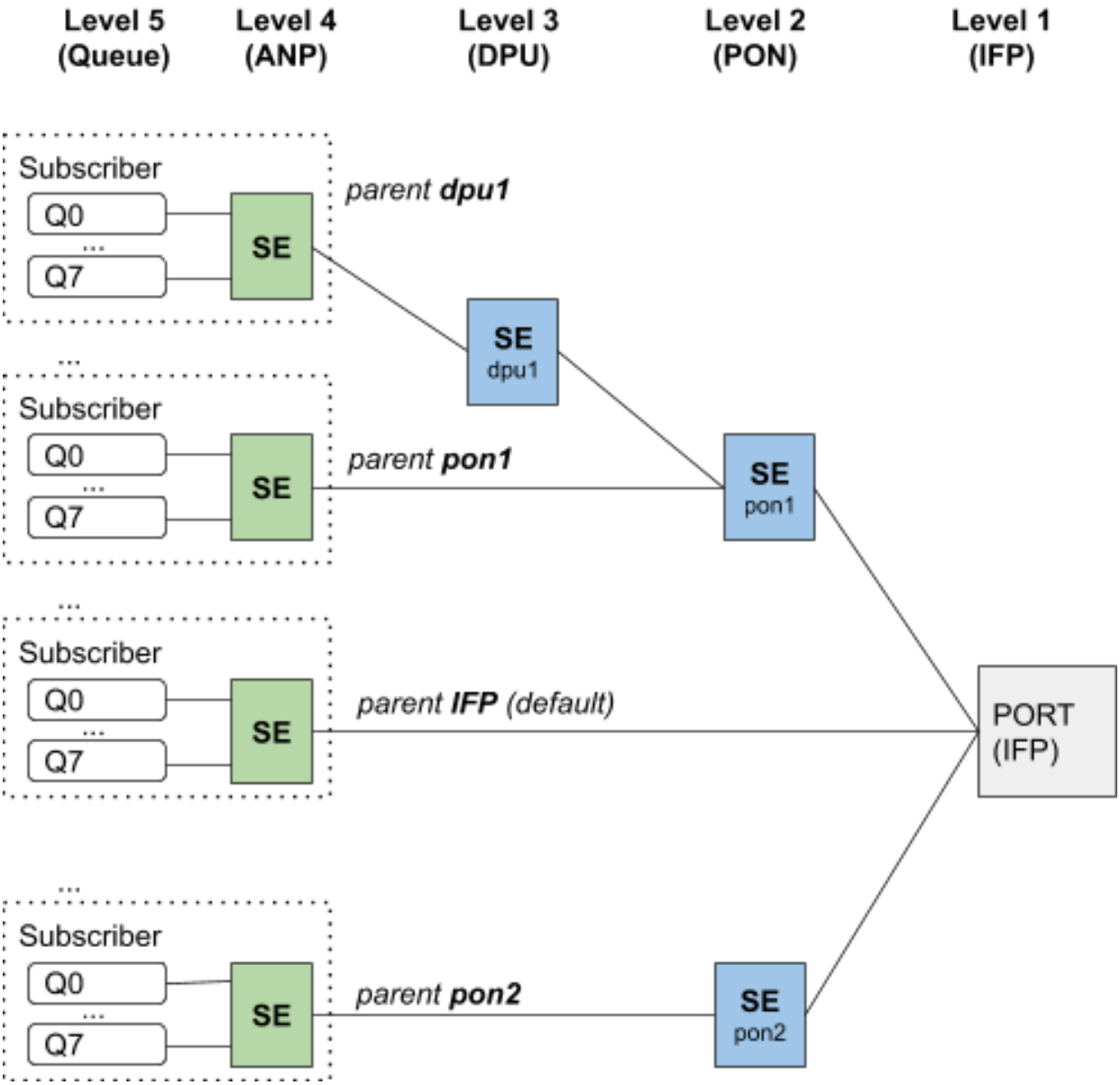
The levels 4 and 5 are configured per logical interface (i.e. subscriber-ifl or l3-ifl).

Separate scheduler-map representing levels 1 to 3 connectivity shall be statically configured and mapped to corresponding physical interface (IFP).

Child scheduler in a subscriber scheduler-map is connected to parent scheduler in physical interface scheduler-map using the following way:

- Dynamically via RADIUS in case of dynamic subscribers like PPPoE sessions (Subscriber-IFL).

The figure below shows the same details as the preceding figure before with the different levels but from the DPU-PON-IFP scheduler-map point of view.



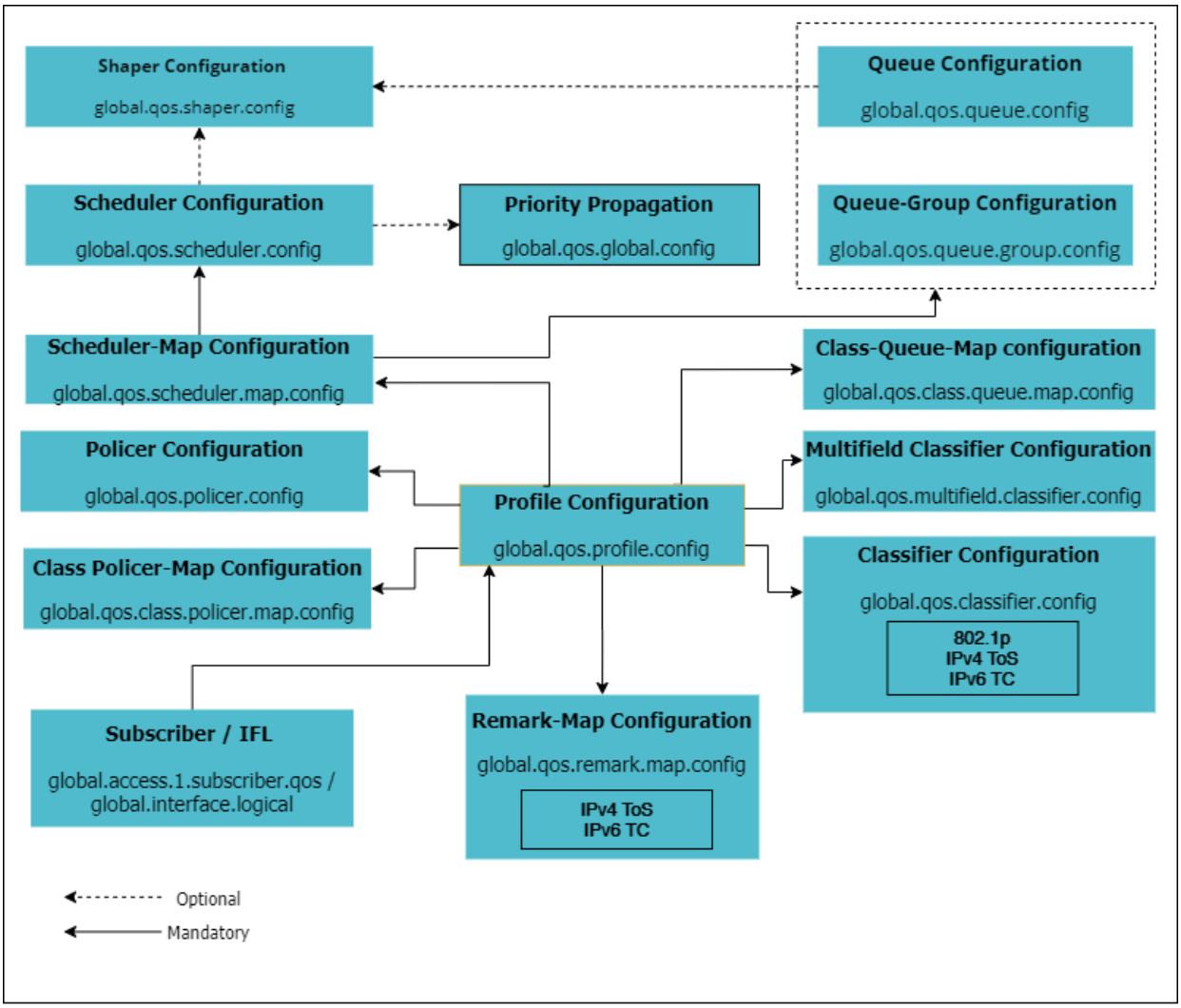
3. Configuring HQoS

To configure HQoS, perform the following steps which include creating a QoS profile and enabling QoS on a PPP Subscriber-Interface or L3-Interface.

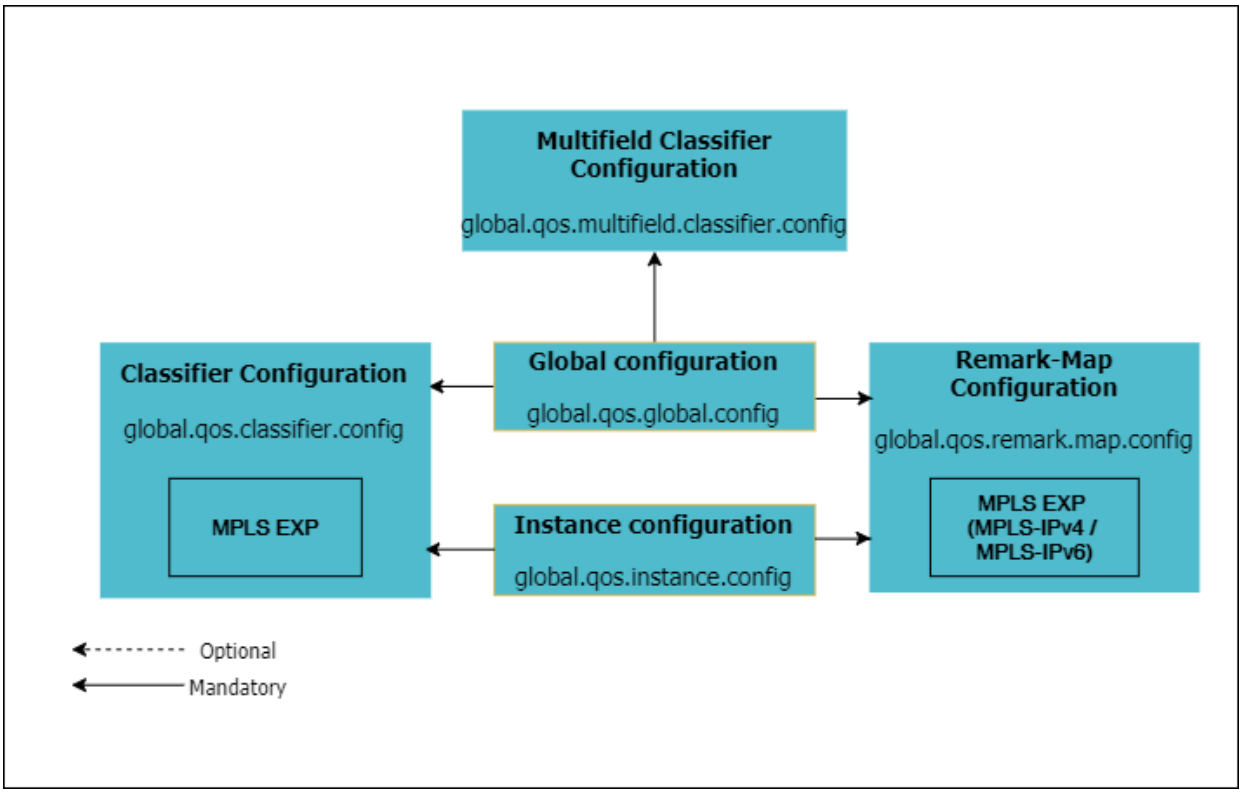
1. Create Behavioral Aggregate (BA) and/or Multifield (MF) classifier to classify the network traffic at the ingress.
2. Create a policer to police the classified traffic at the ingress.
3. Create necessary class-to-policer-map to map the classes to policer-levels (mandatory for policing).
4. Create queue-groups and configure the queue numbers (4/8) in the queue group.
5. Create necessary queues with proper size to queue the classified traffic at the egress.
6. Create necessary class-to-queue-map to map the classes to queues (mandatory for queuing).
7. Specify scheduler(s) with type as required.
8. (Optional) Attach a shaper to queue(s) and/or scheduler(s).
9. Specify a scheduler map to define set of relationships between parent (scheduler or port) and child (queue/queue-group or scheduler) at the egress.
10. (optional) Create Remark-Map for QoS field remarking of the outgoing packet.
11. Define a QoS profile with classifier, multifield-classifier, class-policer-map, policer, class-queue-map, scheduler-map, and remark-map based on user requirements.
12. Specify another scheduler map to represent level-3 to level-5 hierarchy in multi-level HQoS and map it to physical interface.
13. Map the MPLS EXP classifier either to an instance or configure it as global entity.
14. (optional) Map the MPLS-IPv4/IPv6 remark-map either to an instance or configure it as global entity.
15. For downstream MPLS traffic, map the Multifield (MF) classifier as global entity.

Priority propagation is enabled by default. To disable the Priority Propagation, we recommend doing this at the beginning and not during an active session.

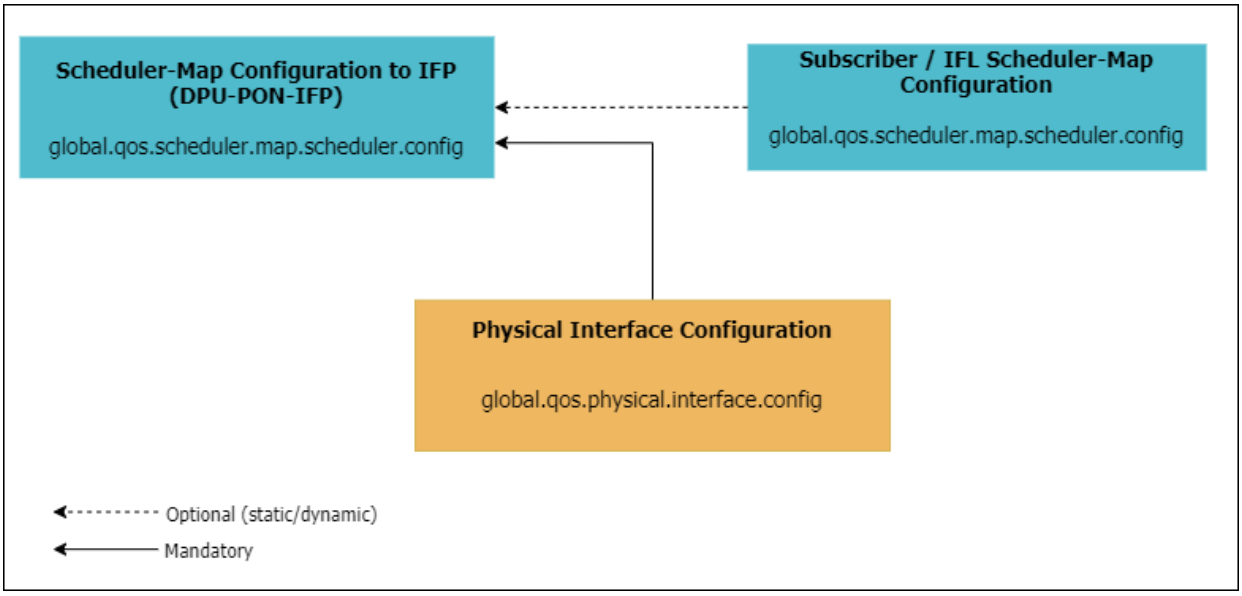
The figure below shows the dependencies between the various HQoS configuration elements.



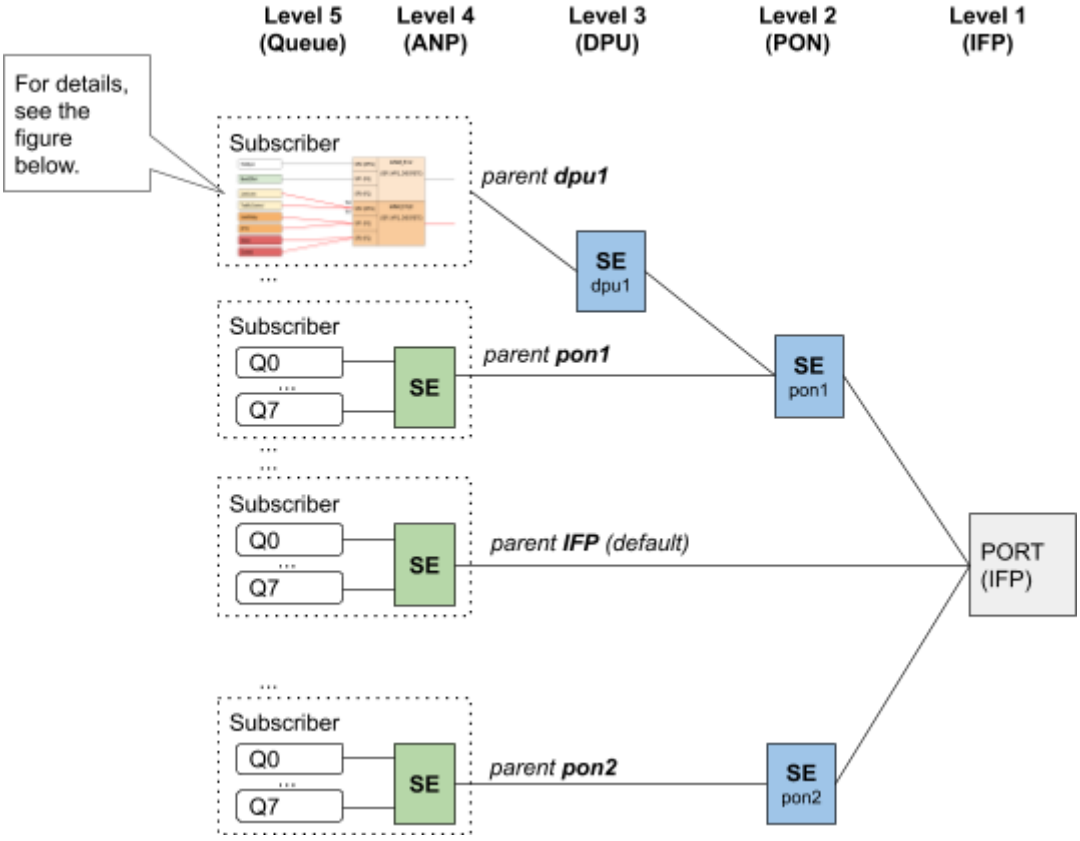
The figure below shows the dependencies for per instance or global classifier and remark-map configurations.

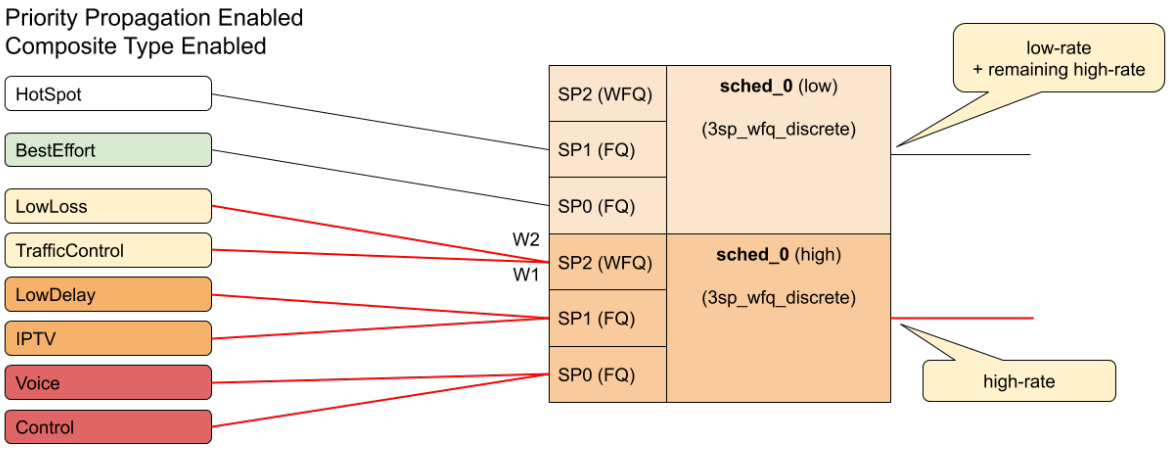


The figure below shows the additional dependencies for Multi-level HQoS.



The figures below show the scheduling hierarchy example.





The following sections provide the commands and examples for configuring HQoS.

- Behavior Aggregate (BA) Classifier Configuration
- Multifield Classifier Configuration
- Remark-Map Configuration
- Policer Configuration
- Class Policer-Map Configuration
- Queue Configuration
- Class Queue-Map Configuration

- Queue-Group Configuration
- Scheduler Configuration
- Scheduler-Map Configuration
- Shaper Configuration
- Profiles Configuration
- Interface Configuration

3.1. Behavior Aggregate (BA) Classifier Configuration

Syntax

set forwarding-options class-of-service classifier <classifier-name> **match-type** <match-type> **codepoint** <codepoint> [**class** <class> | **remark-codepoint** <remark-codepoint>]

Command arguments

<classifier-name>	Specifies the classifier user-defined name
<match-type>	Specifies the type of traffic to classify, that is, ipv4-tos, ipv6-tc, ieee-802.1, exp
<codepoint>	Specifies the code-point value based on match-type
<class>	Specifies the traffic class as as class-0, class-1, class-2, class-3, class-4, class-5, class-6, and class-7
<remark-codepoint>	Specifies the remark-codepoint that used for remarking

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service classifier
residential-ip-classifier match-type ipv6-tc codepoint 192 remark-codepoint
224
root@rtbrick: cfg> commit
```

3.1.1. BA Classifier to Profile Mapping

IPv4, IPv6 and IEEE 802.1p BA classifiers are applied on a subscriber-ifl or l3ifl using the Profile Name.

Syntax

set forwarding-options class-of-service profile <profile-name> **classifier-name** <classifer-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
lac_4queues_4classes classifier-name TC_voice1
root@rtbrick: cfg> commit
```

MPLS Exp classifier is applied either globally or per-instance (to support multiple VPN marking schemes) using Classifier Name.

3.1.2. BA Classifier to Global Mapping

The MPLS classifiers can be applied globally using global configuration.

set forwarding-options class-of-service global classifier-name <classifier-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service global classifier-
name TC_voice1
root@rtbrick: cfg> commit
```

3.1.3. BA Classifier to Instance Mapping

The MPLS classifiers can be applied on an instance using the instance configuration.

set forwarding-options class-of-service instance <instance-name> **classifier-name** <classifier-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service instance ip2vrf
classifier-name TC_voice1
root@rtbrick: cfg> commit
```

3.2. Multifield Classifier Configuration

Syntax

set forwarding-options class-of-service multifield-classifier acl [l3v4 | l3v6] rule <rule-name> **ordinal** <ordinal-value>

In Release 20.10.2, explicit use of ordinal keyword is mandatory in Multifield Classifier configuration.

<rule-name>	Specifies the multifield classifier rule name
<ordinal-value>	Specifies the ordinal value that is used for traffic policy rule referencing

3.2.1. IPv4 Match Configuration

Syntax

set forwarding-options class-of-service multifield-classifier acl l3v4 rule <rule-name> **ordinal** <ordinal-value> **match** [**source-ipv4-prefix** <source-ipv4-prefix> | **destination-ipv4-prefix** <destination-ipv4-prefix> | **ip-protocol** <ip-protocol> | **source-l4-port** <source-l4-port> | **destination-l4-port** <destination-l4-port> | **ipv4-tos** <ipv4-tos> | **ipv4-dscp** <ipv4-dscp> | **forward-class** <forward-class> | **mpls-traffic** <mpls-traffic>]

Command arguments

<rule-name>	Specifies the multifield classifier rule name
<ordinal-value>	Specifies the ordinal that is used for traffic policy rule referencing
<source-ipv4-prefix>	Specifies the source IPv4 prefix address
<destination-ipv4-prefix>	Specifies the destination IPv4 prefix address
<ip-protocol>	Specifies the IP protocol such as UDP or TCP
<source-l4-port>	Specifies the Layer 4 source port number

<destination-l4-port>	Specifies the Layer 4 destination port number
<ipv4-tos	Specifies the IPv4 ToS value
<ipv4-dscp	Specifies the IPv4 dscp value
<forward-class>	Specifies the forward class name
<mpls-traffic>	Species the MPLS traffic

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match destination-ipv4-prefix
3.3.3.3/24
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match direction ingress
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match forward-class class-0
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match ipv4-tos 100
root@rtbrick: cfg> set  forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 100 match ipv4-dscp 55
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match source-ipv4-prefix 2.2.2.2/23
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match destination-l4-port 2000
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match ip-protocol TCP
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match source-l4-port 3000
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule v4 ordinal 99 match mpls-traffic true
root@rtbrick: cfg> commit
```

3.2.2. IPv6 Match Configuration

Syntax

set forwarding-options class-of-service multifield-classifier acl l3v6 rule <rule-name> **ordinal** <ordinal-value> **match** [**source-ipv6-prefix** <source-ipv6-prefix> | **destination-ipv6-prefix** <destination-ipv6-prefix> | **ip-protocol** <ip-protocol> | **source-l4-port** <source-l4-port> | **destination-l4-port** <destination-l4-port> | **ipv6-tc** <ipv6-tc> | **forward-class** <forward-class> | **mpls-traffic** <mpls-traffic>]

Command arguments

<rule-name>	Specifies the multifield classifier rule name

<ordinal-value>	Specifies the ordinal that is used for traffic policy rule referencing
<source-ipv6-prefix>	Specifies the source IPv6 prefix address
<destination-ipv6-prefix>	Specifies the destination IPv6 prefix address
<ip-protocol>	Specifies the IP protocol such as UDP or TCP
<source-l4-port>	Specifies the Layer 4 source port number
<destination-l4-port>	Specifies the Layer 4 destination port number
<ipv6-tc>	Specifies the IPv6 traffic class value
<forward-class>	Specifies the forward class name
<mpls-traffic>	Specifies the MPLS traffic

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match destination-ipv6-prefix
2002::3/64
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match direction ingress
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match forward-class class-1
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match ipv6-tc 200
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match source-ipv6-prefix 3003::32/128
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match destination-l4-port 20000
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match ip-protocol UDP
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match source-l4-port 30000
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v6 rule v6 ordinal 200 match mpls-traffic true
root@rtbrick: cfg> commit
```

3.2.3. IPv4/IPv6 Priority Configuration

Syntax

set forwarding-options class-of-service multifield-classifier acl [**l3v4** |**l3v6**] <rule-name> **ordinal** <ordinal-value> **priority** <priority>

Command arguments

<rule-name>	Specifies the multifield classifier rule name
<ordinal-value>	Specifies the ordinal that is used for traffic policy rule referencing
<priority>	Specify the priority value. Range: 0 - 65535.

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule rtb_mfc ordinal 100 priority 250
root@rtbrick: cfg> commit
```

3.2.4. IPv4/IPv6 Action Configuration

Syntax

set forwarding-options class-of-service multifield-classifier acl [**l3v4** |**l3v6**] **rulename** <rule-name> **ordinal** <ordinal-value> **action** [**forward-class** <class> | **remark-codepoint** <remark-codepoint>]

Command arguments

<rule-name>	Specifies the rule name
<ordinal-value>	Specifies the ordinal that is used for traffic policy rule referencing
<class>	class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7
<remark-codepoint>	Specifies the remark-map codepoint

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service multifield-
classifier acl l3v4 rule rtb_mfc ordinal 100 action remark-codepoint 100
root@rtbrick: cfg> commit
```

3.2.5. MF Classifier to Profile Mapping

Syntax

set forwarding-options class-of-service profile <profile-name> **multifield-classifier-name** <multifield-classifer-name>

Command arguments

<profile-name>	Specifies the profile name
<multifield-classifer-name>	Specifies the multifield classifier name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
rtbrick_residential_profile multifield-classifier-name mf100
root@rtbrick: cfg> commit
```

3.2.6. MF Classifier to Global Mapping

Syntax

set forwarding-options class-of-service global multifield-classifier-name <multifield-classifier-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service global multifield-
classifier-name mf100
root@rtbrick: cfg> commit
```

3.3. Remark-Map Configuration

Syntax

set forwarding-options class-of-service remark-map <remark-map-name> **remark-type** <remark-type> **match-codepoint** <match-codepoint> **color** <color> **remark-codepoint** <remark-codepoint>

Command arguments

<remark-map-name>	Specifies the remaking map name

<remark-type>	Specifies the remarking type - ipv4-tos, ipv6-tc, mpls-ipv4, mpls-ipv6, ieee-802.1
<match-codepoint>	Specifies the match code point
<color>	Indicates the colour - all, green, yellow. Colour is used to set different remark-codepoint for same match-codepoint based on color marked by the Policer.
<remark-codepoint>	Specifies the remarking codepoint

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service remark-map remark-
exp remark-type ipv6-tc match-codepoint 100 color all remark-codepoint 224
root@rtbrick: cfg> commit
```

3.3.1. Remark-map to Profile Mapping

Syntax

set forwarding-options class-of-service profile <profile-name> **remark-map-name** <remark-map-name>

Command arguments

<profile-name>	Specifies the profile name
<remark-map-name>	Specifies the remark map name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
rtbrick_residential_profile remark-map-name remark-exp
root@rtbrick: cfg> commit
```

3.3.2. Global Profile Mapping

set forwarding-options class-of-service global remark-map-name <remark-map-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service global remark-map-
name remark-exp
root@rtbrick: cfg> commit
```

3.3.3. Remark-map to Instance Mapping

set forwarding-options class-of-service instance <instance-name> **remark-map-name** <remark-map-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service instance ip2vrf
remark-map-name remark-exp
root@rtbrick: cfg> commit
```

3.4. Policer Configuration

Syntax

set forwarding-options class-of-service policer <policer-name>

Command Arguments

<policer-name>	Specifies the policer name.
<levels>	Specifies levels in the Policer. Levels will be from 1 to 4.
<type>	Specifies the policer type.
<flag>	Set flags.
level1-rates <cir>	Set committed information rate (CIR) in kilobits per second (kbps) for level-1. The same is applicable for level-2 to level-4.
level1-rates <pir>	Set peak information rate (PIR) in kilobits per second (kbps) for level-1. The same is applicable for level-2 to level-4.
level1-rates <cbs>	Set Committed burst size (CBS) in kilobits for level-1. The same is applicable for level-2 to level-4.
level1-rates <pbs>	Set peak burst size (PBS) in kilobits for level-1. The same is applicable for level-2 to level-4.
level1-rates <max-cir>	Set the maximum for the level-1 committed information rate (CIR) in kilobits per second (kbps). The same is applicable for level-2 to level-4.

level1-rates <max-pir>	Set the maximum for the level-1 peak information rate (PIR) in kilobits per second (kbps). The same is applicable for level-2 to level-4.

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential
root@rtbrick: cfg> commit
```

Levels configuration

set forwarding-options class-of-service policer <policer-name> **levels** <levels>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential levels 4
root@rtbrick: cfg> commit
```

Type configuration

set forwarding-options class-of-service policer <policer-name> **type** <type>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential type two-rate-three-color
root@rtbrick: cfg> commit
```

Flag configuration

set forwarding-options class-of-service policer <policer-name> **flags** <flags>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential flags color-blind
root@rtbrick: cfg> commit
```

Level rates configuration

set forwarding-options class-of-service policer <policer-name> **level1-rates** [**cir** <cir>]

set forwarding-options class-of-service policer <policer-name> **level1-rates** [**pir** <pir>]

set forwarding-options class-of-service policer <policer-name> **level1-rates** [**cbs** <cbs>]

set forwarding-options class-of-service policer <policer-name> **level1-rates** [**pbs** <pbs>]

set forwarding-options class-of-service policer <policer-name> **level1-rates** [**max-cir** <max-cir>]

set forwarding-options class-of-service policer <policer-name> **level1-rates** [**max-pir** <max-pir]

The same is applicable for level-2 to level-4.

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level1-rates cir 8000
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level1-rates cbs 800
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level1-rates pir 8000
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level1-rates pbs 800
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level3-rates cir 0
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level3-rates cbs 800
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level3-rates pir 0
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level3-rates pbs 800
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level4-rates cir 0
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level4-rates cbs 800
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level4-rates pir 0
root@rtbrick: cfg> set forwarding-options class-of-service policer policer-
residential level4-rates pbs 800
root@rtbrick: cfg> commit
```

Example

```
set forwarding-options class-of-service policer policer-residential level1-
rates max-cir 4500
set forwarding-options class-of-service policer policer-residential level1-
rates max-pir 5500
root@rtbrick: cfg> commit
```

3.5. Class Policer-Map Configuration

Syntax

set forwarding-options class-of-service class-policer-map <class-policer-map-name> **class** <class> **policer-level** <policer-level>

Command arguments

<class-policer-map-name>	Specifies the class policer map name,
<class>	Specifies the class such as class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7

<policer-level>	level-1, level-2, level-3, level-4

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service class-policer-map
policer-map-l2tp class class-0 policer-level level-4
root@rtbrick: cfg> set forwarding-options class-of-service class-policer-map
policer-map-l2tp class class-1 policer-level level-3
root@rtbrick: cfg> set forwarding-options class-of-service class-policer-map
policer-map-l2tp class class-2 policer-level level-2
root@rtbrick: cfg> set forwarding-options class-of-service class-policer-map
policer-map-l2tp class class-3 policer-level level-1
root@rtbrick: cfg> set forwarding-options class-of-service class-policer-map
policer-map-residential class class-0 policer-level level-3
set forwarding-options class-of-service class-policer-map policer-map-
residential class class-1 policer-level level-4
root@rtbrick: cfg> set forwarding-options class-of-service class-policer-map
policer-map-residential class class-2 policer-level level-2
root@rtbrick: cfg> set forwarding-options class-of-service class-policer-map
policer-map-residential class class-3 policer-level level-1
root@rtbrick: cfg> commit
```

3.6. Queue Configuration

Syntax

set forwarding-options class-of-service queue <queue-name>

Command arguments

<queue-name>	Specifies the user-defined queue name	
<queue-size>	Specifies the size of the queue in bytes	
<shaper-name>	(Optional) Specifies the shaper that is associated with the queue	
WRED	<minimum-threshold>	Specifies the minimum average queue size to apply WRED in bytes
	<maximum-threshold>	Specifies the maximum average queue size to apply WRED in bytes
	<drop-prob>	WRED drop probability applied at the maximum threshold

<bytes>	Specifies the header compensation value
decrement <true |false>	The header compensation value is -ve (if true) or +ve (if false).

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service queue BE_L
root@rtbrick: cfg> commit
```

Queue Size

set forwarding-options class-of-service queue <queue-name> **queue-size** <queue-size>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service queue BE_L queue-
size 375000
root@rtbrick: cfg> commit
```

Queue WRED Profile

set forwarding-options class-of-service queue <queue-name> **wred minimum-threshold** <minimum-threshold>

set forwarding-options class-of-service queue <queue-name> **wred maximum-threshold** <maximum-threshold>

set forwarding-options class-of-service queue <queue-name> **wred drop-probability** <drop-probability>

Example

```
root@rtbrick: cfg> set forwarding-options queue BE_L wred minimum-threshold
15000
root@rtbrick: cfg> set forwarding-options queue BE_L wred maximum-threshold
18000
root@rtbrick: cfg> set forwarding-options queue BE_L wred drop-probability 70
root@rtbrick: cfg> commit
```

Queue Shaper

set forwarding-options class-of-service queue <queue-name> **shaper-name** <shaper-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service queue IO_L shaper-
name shaper_IO
root@rtbrick: cfg> commit
```

Queue Header Compensation

set forwarding-options class-of-service queue <queue-name> **header-compensation bytes** <bytes>

set forwarding-options class-of-service queue <queue_name> **header-compensation decrement** <*true* | **false**>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service queue IO_L header-
compensation bytes 30
root@rtbrick: cfg> set forwarding-options class-of-service queue IO_L header-
compensation decrement true
root@rtbrick: cfg> commit
```

3.7. Class Queue-Map Configuration

3.7.1. Class to Queue mapping

Syntax

set forwarding-options class-of-service class-queue-map <class-queue-map-name> **class** <class> **queue-name** <queue-name>

Command arguments

<class-queue-map-name>	Specifies the class queue map name
<class>	Specifies the class such as class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7

<queue-name>	Specifies the queue name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service class-queue-map
lac_4queues_L class class-0 queue-name BE_L
root@rtbrick: cfg> commit
```

3.8. Queue-Group Configuration

Queue group size: 4 or 8

Syntax

set forwarding-options class-of-service queue-group <queue-group-name> **queue-numbers** <queue-numbers>

Command arguments

<queue-group-name>	User-defined name for the queue-group
<queue-numbers>	Specifies the number of queues in a Queue Group

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
lac_4queues_L queue-numbers 4
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
lac_4queues_M queue-numbers 4
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
lac_4queues_S queue-numbers 4
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
pta_4queues_L queue-numbers 4
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
pta_4queues_M queue-numbers 4
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
pta_4queues_S queue-numbers 4
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
pta_8queues_L queue-numbers 8
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
pta_8queues_M queue-numbers 8
root@rtbrick: cfg> set forwarding-options class-of-service queue-group
pta_8queues_S queue-numbers 8
root@rtbrick: cfg> commit
```

3.9. Scheduler Configuration

Syntax

set forwarding-options class-of-service scheduler <scheduler-name>

Command arguments

<scheduler-name>	User-defined Scheduler Name
<shaper-name>	(Optional) User-defined Shaper Name
<type>	Specifies the Scheduler Type 2sp_wfq_discrete 3sp_wfq_discrete strict_priority wfq_discrete_2sp 2sp_wfq_independent fair_queueing weighted_fair_queueing wfq_independent_2sp
composite <true | false>	(Optional) keyword to specify the scheduler as composite type

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service scheduler
lac_4queues
root@rtbrick: cfg> commit
```

Scheduler Type

set forwarding-options class-of-service scheduler <scheduler-name> <type> [**composite** <true | false>]

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service scheduler
lac_4queues type strict_priority
root@rtbrick: cfg> set forwarding-options class-of-service scheduler
lac_4queues composite true
root@rtbrick: cfg> commit
```

Scheduler Shaper

set forwarding-options class-of-service scheduler <scheduler-name> **shaper-name** <shaper-name>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service scheduler
lac_4queues shaper-name shaper_session
root@rtbrick: cfg> commit
```

3.10. Scheduler-Map Configuration

Syntax

set forwarding-options class-of-service scheduler-map <scheduler-map-name> [**group-name** <group-name> **queue-name** <name>] [**scheduler-name** <scheduler-name>]

Command arguments

<scheduler-map-name>	User-defined Scheduler-Map Name
<scheduler-name>	User-defined Scheduler Name
<group-name>	User-defined Queue-Group Name
<name>	User-defined Queue-Name
<parent-scheduler-name>	Name of the parent scheduler
<connection-type>	Specifies the type of port connection, that is, queue_to_port or scheduler_to_port
<connection-point>	Specifies the type of connection point, such as no_priority, strict_priority_0, strict_priority_1, strict_priority_2, strict_priority

3.10.1. Queue to Port

Syntax

set forwarding-options class-of-service scheduler-map <scheduler-map-name> **queue-group-name** <queue-group-name> **queue-name** <name> **port-connection** <port-connection>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service scheduler-map
lac_4queues_S queue-group-name lac_4queues_S queue-name BE_S port-connection
queue_to_port
root@rtbrick: cfg> commit
```

3.10.2. Scheduler to Port

Syntax

set forwarding-options class-of-service scheduler-map <scheduler-map-name> **scheduler-name** <scheduler-name> **port-connection** <port-connection>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service scheduler-map
lac_4queues_S scheduler-name olt-pon1 port-connection scheduler_to_port
root@rtbrick: cfg> commit
```

3.10.3. Scheduler to Scheduler (same Scheduler-Map)

Syntax

set forwarding-options class-of-service scheduler-map <scheduler-map-name> **scheduler-name** <scheduler-name>

set forwarding-options class-of-service scheduler-map <scheduler-map-name> **scheduler-name** <scheduler-name> **connection-point** <connection-point>

set forwarding-options class-of-service scheduler-map <scheduler-map-name> **scheduler-name** <scheduler-name> **parent-scheduler-name** <parent-scheduler-name>

set forwarding-options class-of-service scheduler-map <scheduler-map-name> **scheduler-name** <scheduler-name> [**weight** <weight>]

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service scheduler-map
pta_4queues_comp_off_M  scheduler-name olt-pon1
root@rtbrick: cfg> set forwarding-options class-of-service scheduler-map
pta_4queues_comp_off_M scheduler-name olt-pon1 connection-point
strict_priority_1
root@rtbrick: cfg> set forwarding-options class-of-service scheduler-map
pta_4queues_comp_off_M scheduler-name olt-pon1 lac_4queues
root@rtbrick: cfg> set forwarding-options class-of-service scheduler-map
pta_4queues_comp_off_M scheduler-name olt-pon1 weight 1
root@rtbrick: cfg> commit
```

3.10.4. Queue to scheduler

Syntax

set forwarding-options class-of-service queue-group-name <queue-group-name> **queue-name** <name> **parent-scheduler** <parent-scheduler-name>

set forwarding-options class-of-service queue-group-name <queue-group-name> **queue-name** <name> **parent-flow** <parent-flow>

set forwarding-options class-of-service queue-group-name <queue-group-name> **queue-name** <name> **connection-point** <connection-point>

set forwarding-options class-of-service queue-group-name <queue-group-name> **queue-name** <name> [**weight** <weight>]

The parent-flow configuration is optional.

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service queue-group-name
queue_group_residential queue-name IPTV parent-scheduler rtbrick_sched_0
root@rtbrick: cfg> set forwarding-options class-of-service queue-group-name
queue_group_residential queue-name IPTV parent-flow low-flow
root@rtbrick: cfg> set forwarding-options class-of-service queue-group-name
queue_group_residential queue-name IPTV connection-point strict_priority_0
root@rtbrick: cfg> set forwarding-options class-of-service queue-group-name
queue_group_residential queue-name IPTV weight 1
root@rtbrick: cfg> commit
```

3.11. Shaper Configuration

Syntax

set forwarding-options class-of-service shaper <shaper-name>

Command Arguments

<shaper-name>	User-defined shaper name
<shaping-rate-high>	High flow shaping rate in kilobits per second
<shaping-rate-low>	Low flow shaping rate in kilobits per second

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service shaper
session_shaper
root@rtbrick: cfg> commit
```

High Flow Shaping Rate

To configure only high-flow shaping rate, enter the following command:

set forwarding-options class-of-service shaper <shaper-name> **shaping-rate-high** <shaping-rate-high>

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service shaper shaper_LD
shaping-rate-high 2488000
root@rtbrick: cfg> commit
```

Low Flow Shaping Rate

To configure only low-flow shaping rate, enter the following command:

set forwarding-options class-of-service shaper <shaper-name> **shaping-rate-low** <shaping-rate-low>

If priority propagation is not enabled, high-flow shaping value will be considered for shaper.

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service shaper shaper_LD
shaping-rate-low 0
root@rtbrick: cfg> commit
```

3.12. Priority Propagation

Syntax

set forwarding-options class-of-service global priority-propagation [**enable** | **disable**]

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service global priority-
propagation enable
root@rtbrick: cfg> commit
```

3.13. Profiles Configuration

Syntax

set forwarding-options class-of-service profile <profile-name>

Command arguments

<profile-name>	User-defined QoS Profile name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
lac_4queues_4classes
root@rtbrick: cfg> commit
```

BA Classifier

set forwarding-options class-of-service profile <profile-name> **classifier-name** <classifier_name>

Command arguments

<profile-name>	Profile name
<classifier-name>	Classifier name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
lac_4queues_4classes classifier-name residential-pbit-classifier
root@rtbrick: cfg> commit
```

Multifield Classifier

set forwarding-options class-of-service profile <profile-name> **multifield-classifier-name** <multifield-classifer-name>

Command arguments

<profile-name>	Profile name
<multifield-classifier-name>	MF Classifier name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profiles profile-
name rtbrick_residential_profile multifield-classifier-name mf100
root@rtbrick: cfg> commit
```

Class Policer-Map

set forwarding-options class-of-service profile <profile-name> **class-policer-map-name** <class-policer-map-name>

Command arguments

<profile-name>	Profile name
<class-policer-map-name>	User-defined class to policer level map name map name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
pta_8queues_comp_on_S class-policer-map-name policer-map-residential
root@rtbrick: cfg> commit
```

Policer

set forwarding-options class-of-service profile <profile-name> **policer-name** <policer-name>

Command arguments

<profile-name>	Profile name
<policer-name>	User-defined Policer name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
pta_8queues_comp_on_S policer-name policer-residential
root@rtbrick: cfg> commit
```

Class Queue-Map

set forwarding-options class-of-service profile <profile-name> **class-queue-map-name** <class-queue-map-name>

Command arguments

<profile-name>	Profile name
<class-queue-map-name>	User-defined class to queue map name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
pta_8queues_comp_on_L class-queue-map-name pta_8queues_L
root@rtbrick: cfg> commit
```

Scheduler-Map

set forwarding-options class-of-service profile <profile-name> **scheduler-map-name** <scheduler-map-name>

Command arguments

profile-name	Profile name
<scheduler-map-name>	User-defined Scheduler map name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
pta_8queues_comp_on scheduler-map-name pta_8queues_comp_on_M
root@rtbrick: cfg> commit
```

Remark-Map

set forwarding-options class-of-service profile <profile-name> **remark-map-name** <remark-map-name>

Command arguments

<remark-map-name>	Remarking map name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service profile
pta_8queues_comp_on remark-map-name remark-exp
root@rtbrick: cfg> commit
```

3.14. Interface Configuration

3.14.1. Logical Interface QoS Profile

QOS Profile can be mapped to an L3 interface (that is, logical interface).

Syntax

set interface <ifp-name> **unit** <unit-id> **class-of-service** <class-of-service>

Command arguments

<ifp-Name>	Logical Interface Name
<unit-id>	Configure unit which identifies sub-interface under physical interface
<class-of-service>	Class of service

Example

```
root@rtbrick: cfg> set interface ifl-0/0/1 unit 100 class-of-service profile1
root@rtbrick: cfg> commit
```

3.14.2. Physical Interface Scheduler Map

Syntax

set forwarding-options class-of-service interface <name> **scheduler-map-name** <scheduler-map-name>

Command arguments

<name>	Physical Interface Name
<scheduler-map-name>	Scheduler map name

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service interface ifp-
0/0/3 scheduler-map-name  pta_8queues_comp_on
root@rtbrick: cfg> commit
```

3.14.3. Physical Interface Shaper

Syntax

set forwarding-options class-of-service interface <name> **shaping-rate** <shaping-rate>

Command arguments

<name>	Physical Interface Name
<shaping-rate>	Shaping Rate

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service interface ifp-
0/0/3 shaping-rate 10000
root@rtbrick: cfg> commit
```

3.14.4. Physical Interface Header Compensation

3.14.4.1. Ingress Header Compensation

Syntax

set forwarding-options class-of-service interface <name> **ingress-header-compensation bytes** <bytes>

set forwarding-options class-of-service interface <name> **ingress-header-compensation decrement** <true | false>

Command arguments

<name>	Physical Interface Name
<bytes>	Ingress header compensation bytes
decrement <true | false>	Specifies whether the header compensation to be decremented

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service interface ifp-
0/0/19 ingress-header-compensation bytes 40
root@rtbrick: cfg> set forwarding-options class-of-service interface ifp-
0/0/19 ingress-header-compensation decrement true
root@rtbrick: cfg> commit
```

3.14.4.2. Egress Header Compensation

Syntax

set forwarding-options class-of-service interface <name> **egress-header-compensation bytes** <bytes>

set forwarding-options class-of-service interface <name> **egress-header-compensation decrement decrement** <true | false>

Command arguments

<name>	Physical Interface Name
<bytes>	Egress header compensation bytes
decrement <true | false>	Specifies whether the header compensation to be decremented

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service interface ifp-
0/0/19 egress-header-compensation bytes 30
root@rtbrick: cfg> set forwarding-options class-of-service interface ifp-
0/0/19 egress-header-compensation decrement true
root@rtbrick: cfg> commit
```

3.14.5. L2TP Tunnel Classifier

Syntax

set forwarding-options class-of-service global l2tp-classifier-name <l2tp-classifier-name>

Command arguments

<l2tp-classifier-name>	Name of the L2TP classifier

Example

```
root@rtbrick: cfg> set forwarding-options class-of-service global l2tp-
classifier-name BA_L2TP_HEADER_CLASSIFIER_ONE
root@rtbrick: cfg> commit
```

4. HQoS Show Running-Configuration

To display the running configuration, use the **show running-configuration** command.

Syntax

show running-configuration

Example

```
supervisor@rtbrick: cfg> show config forwarding-options class-of-service
{
  "rtbrick-config:class-of-service": {
    "classifier": [
      {
        "classifier-name": "BA_L2TP_UPSTREAM_CLASSIFIER_ONE",
        "match-type": [
          {
            "match-type": "ieee-802.1",
            "codepoint": [
              {
                "codepoint": 1,
                "class": "class-1"
              },
              {
                "codepoint": 2,
                "class": "class-2"
              },
              {
                "codepoint": 3,
                "class": "class-3"
              },
              {
                "codepoint": 4,
                "class": "class-4"
              }
            ]
          }
        ]
      },
      {
        "classifier-name": "EXP_CLASSIFIER",
        "match-type": [
          {
            "match-type": "exp",
            "codepoint": [
              {
                "codepoint": 0,
                "class": "class-0"
              },
```

```
            {
              "codepoint": 1,
              "class": "class-1"
            },
            {
              "codepoint": 2,
              "class": "class-2"
            },
            {
              "codepoint": 3,
              "class": "class-3"
            }
          ]
        }
      ]
    }
  ],
  "class-policer-map": [
    {
      "class-policer-map-name": "LAC_CLASS_POLICER_MAP",
      "class": [
        {
          "class": "class-0",
          "policer-level": "level-1"
        },
        {
          "class": "class-1",
          "policer-level": "level-1"
        },
        {
          "class": "class-2",
          "policer-level": "level-2"
        },
        {
          "class": "class-3",
          "policer-level": "level-2"
        },
        {
          "class": "class-4",
          "policer-level": "level-3"
        },
        {
          "class": "class-5",
          "policer-level": "level-3"
        },
        {
          "class": "class-6",
          "policer-level": "level-4"
        },
        {
          "class": "class-7",
          "policer-level": "level-4"
        }
      ]
    },
    {
      "class-policer-map-name": "PTA_CLASS_POLICER_MAP",
      "class": [
```

```
        {
          "class": "class-0",
          "policer-level": "level-1"
        },
        {
          "class": "class-1",
          "policer-level": "level-1"
        },
        {
          "class": "class-2",
          "policer-level": "level-2"
        },
        {
          "class": "class-3",
          "policer-level": "level-2"
        },
        {
          "class": "class-4",
          "policer-level": "level-3"
        },
        {
          "class": "class-5",
          "policer-level": "level-3"
        },
        {
          "class": "class-6",
          "policer-level": "level-4"
        },
        {
          "class": "class-7",
          "policer-level": "level-4"
        }
      ]
    }
  ],
  "class-queue-map": [
    {
      "class-queue-map-name": "CLASS_QUEUE_MAP_LAC_RESIDENTIAL",
      "class": [
        {
          "class-type": "class-0",
          "queue-name": "LAC_IPTV"
        },
        {
          "class-type": "class-1",
          "queue-name": "LAC_VOIP"
        },
        {
          "class-type": "class-2",
          "queue-name": "LAC_CONTROL"
        },
        {
          "class-type": "class-3",
          "queue-name": "LAC_LOWLOSS"
        }
      ]
    },
    {
```

```
        "class-queue-map-name": "CLASS_QUEUE_MAP_PTA_RESIDENTIAL",
        "class": [
          {
            "class-type": "class-0",
            "queue-name": "PTA_IPTV"
          },
          {
            "class-type": "class-1",
            "queue-name": "PTA_VOIP"
          },
          {
            "class-type": "class-2",
            "queue-name": "PTA_CONTROL"
          },
          {
            "class-type": "class-3",
            "queue-name": "PTA_LOWLOSS"
          }
        ]
      }
    ],
    "global": {
      "classifier-name": "EXP_CLASSIFIER"
    },
    "interface": [
      {
        "name": "ifp-0/0/3",
        "scheduler-map-name": "DPU_GPON_THREE"
      }
    ],
    "policer": [
      {
        "policer-name": "LAC_SUBSCRIBERS_POLICER",
        "flags": "color-blind",
        "level1-rates": {
          "cir": 9000,
          "cbs": 1000,
          "pir": 9200,
          "pbs": 1000
        },
        "level2-rates": {
          "cir": 1000,
          "cbs": 1000,
          "pir": 1200,
          "pbs": 1000
        },
        "level3-rates": {
          "cir": 3100,
          "cbs": 1000,
          "pir": 3500,
          "pbs": 1000
        },
        "level4-rates": {
          "cir": 7100,
          "cbs": 1000,
          "pir": 7500,
          "pbs": 1000
        },
```

```
        "levels": 4,
        "type": "two-rate-three-color"
      },
      {
        "policer-name": "PTA_SUBSCRIBERS_POLICER",
        "flags": "color-blind",
        "level1-rates": {
          "cir": 9000,
          "cbs": 1000,
          "pir": 9200,
          "pbs": 1000
        },
        "level2-rates": {
          "cir": 1000,
          "cbs": 1000,
          "pir": 1200,
          "pbs": 1000
        },
        "level3-rates": {
          "cir": 3100,
          "cbs": 1000,
          "pir": 3500,
          "pbs": 1000
        },
        "level4-rates": {
          "cir": 7100,
          "cbs": 1000,
          "pir": 7500,
          "pbs": 1000
        },
        "levels": 4,
        "type": "two-rate-three-color"
      }
    ],
    "profile": [
      {
        "profile-name": "LAC_RESIDENTIAL_PROFILE_RTBRICK",
        "classifier-name": "BA_L2TP_UPSTREAM_CLASSIFIER_ONE",
        "class-queue-map-name": "CLASS_QUEUE_MAP_LAC_RESIDENTIAL",
        "class-policer-map-name": "LAC_CLASS_POLICER_MAP",
        "policer-name": "LAC_SUBSCRIBERS_POLICER",
        "scheduler-map-name": "LAC_SCHEDULER_MAP_RTBRICK_RESIDENTIAL"
      },
      {
        "profile-name": "PTA_RESIDENTIAL_PROFILE_RTBRICK",
        "class-queue-map-name": "CLASS_QUEUE_MAP_PTA_RESIDENTIAL",
        "remark-map-name": "REMARK_DOWN",
        "scheduler-map-name": "PTA_SCHEDULER_MAP_RTBRICK_RESIDENTIAL",
        "class-policer-map-name": "PTA_CLASS_POLICER_MAP",
        "policer-name": "PTA_SUBSCRIBERS_POLICER",
      }
    ],
    "queue": [
      {
        "queue-name": "LAC_CONTROL",
        "queue-size": 250000
      },
      {
```

```
        "queue-name": "LAC_IPTV",
        "queue-size": 250000
      },
      {
        "queue-name": "LAC_LOWLOSS",
        "queue-size": 250000
      },
      {
        "queue-name": "LAC_VOIP",
        "queue-size": 250000
      },
      {
        "queue-name": "PTA_CONTROL",
        "queue-size": 250000
      },
      {
        "queue-name": "PTA_IPTV",
        "queue-size": 250000
      },
      {
        "queue-name": "PTA_LOWLOSS",
        "queue-size": 250000
      },
      {
        "queue-name": "PTA_VOIP",
        "queue-size": 250000
      }
    ],
    "queue-group": [
      {
        "queue-group-name": "LAC_QUEUE_GROUP_ONE",
        "queue-numbers": 4
      },
      {
        "queue-group-name": "PTA_QUEUE_GROUP_ONE",
        "queue-numbers": 4
      }
    ],
    "remark-map": [
      {
        "remark-map-name": "REMARK_DOWN",
        "remark-type": [
          {
            "remark-type": "ieee-802.1",
            "match-codepoint": [
              {
                "match-codepoint": 2,
                "color": [
                  {
                    "color": "all",
                    "remark-codepoint": 4
                  }
                ]
              },
              {
                "match-codepoint": 64
              }
            ]
```

```
        }
      ]
    }
  ],
  "scheduler": [
    {
      "scheduler-name": "DPU_FOUR_SCHEDULER",
      "type": "fair_queueing"
    },
    {
      "scheduler-name": "DPU_ONE_SCHEDULER",
      "type": "fair_queueing"
    },
    {
      "scheduler-name": "GPON_ONE_SCHEDULER",
      "type": "fair_queueing"
    },
    {
      "scheduler-name": "GPON_THREE_SCHEDULER",
      "type": "fair_queueing"
    },
    {
      "scheduler-name": "LAC_ONE_SCHEDULER_RTBRICK",
      "shaper-name": "SESSION_SHAPER",
      "type": "3sp_wfq_discrete",
      "composite": "true"
    },
    {
      "scheduler-name": "PTA_ONE_SCHEDULER_RTBRICK",
      "shaper-name": "SESSION_SHAPER",
      "type": "3sp_wfq_discrete",
      "composite": "true"
    }
  ],
  "scheduler-map": [
    {
      "scheduler-map-name": "DPU_GPON_ONE",
      "scheduler-name": [
        {
          "name": "DPU_ONE_SCHEDULER",
          "parent-scheduler-name": "GPON_ONE_SCHEDULER",
          "connection-point": "no_priority"
        },
        {
          "name": "GPON_ONE_SCHEDULER"
        }
      ]
    },
    {
      "scheduler-map-name": "DPU_GPON_THREE",
      "scheduler-name": [
        {
          "name": "DPU_FOUR_SCHEDULER",
          "parent-scheduler-name": "GPON_THREE_SCHEDULER",
          "connection-point": "no_priority"
        },
        {
          "name": "GPON_THREE_SCHEDULER"
```

```
        }
      ]
    },
    {
      "scheduler-map-name": "LAC_SCHEDULER_MAP_RTBRICK_RESIDENTIAL",
      "queue-group-name": [
        {
          "group-name": "LAC_QUEUE_GROUP_ONE",
          "queue-name": [
            {
              "name": "LAC_CONTROL",
              "parent-flow": "high-flow",
              "parent-scheduler-name": "PTA_ONE_SCHEDULER_RTBRICK",
              "connection-point": "strict_priority_2",
              "weight": 2
            },
            {
              "name": "LAC_IPTV",
              "parent-flow": "high-flow",
              "parent-scheduler-name": "LAC_ONE_SCHEDULER_RTBRICK",
              "connection-point": "strict_priority_1"
            },
            {
              "name": "LAC_LOWLOSS",
              "parent-flow": "low-flow",
              "parent-scheduler-name": "LAC_ONE_SCHEDULER_RTBRICK",
              "connection-point": "strict_priority_2",
              "weight": 1
            },
            {
              "name": "LAC_VOIP",
              "parent-flow": "high-flow",
              "parent-scheduler-name": "LAC_ONE_SCHEDULER_RTBRICK",
              "connection-point": "strict_priority_0"
            }
          ]
        }
      ],
      "scheduler-name": [
        {
          "name": "LAC_ONE_SCHEDULER_RTBRICK"
        }
      ]
    },
    {
      "scheduler-map-name": "PTA_SCHEDULER_MAP_RTBRICK_RESIDENTIAL",
      "queue-group-name": [
        {
          "group-name": "PTA_QUEUE_GROUP_ONE",
          "queue-name": [
            {
              "name": "PTA_CONTROL",
              "parent-flow": "high-flow",
              "parent-scheduler-name": "PTA_ONE_SCHEDULER_RTBRICK",
              "connection-point": "strict_priority_2",
              "weight": 2
            },
            {
```

```
                "name": "PTA_IPTV",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "PTA_ONE_SCHEDULER_RTBRICK",
                "connection-point": "strict_priority_1"
              },
              {
                "name": "PTA_LOWLOSS",
                "parent-flow": "low-flow",
                "parent-scheduler-name": "PTA_ONE_SCHEDULER_RTBRICK",
                "connection-point": "strict_priority_2",
                "weight": 1
              },
              {
                "name": "PTA_VOIP",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "PTA_ONE_SCHEDULER_RTBRICK",
                "connection-point": "strict_priority_0"
              }
            ]
          }
        ],
        "scheduler-name": [
          {
            "name": "PTA_ONE_SCHEDULER_RTBRICK"
          }
        ]
      }
    ],
    "shaper": [
      {
        "shaper-name": "SESSION_SHAPER",
        "shaping-rate-high": 10000,
        "shaping-rate-low": 10200
      }
    ]
  }
}
```

5. HQoS Show Commands

5.1. show qos

```
supervisor@rtbrick: op> show qos
 classifier             interface              policer                profile
queue
 remark-map             scheduler              scheduler-map          shaper
subscriber
```

5.2. show qos classifier

```
supervisor@rtbrick: op> show qos classifier
Classifier: residential-ip-classifier
Active: False
  Match Type   Codepoint    Class      Remark Codepoint    Color
  ipv4-tos     0            class-0    -                   -
  ipv4-tos     32           class-1    -                   -
  ipv4-tos     64           class-2    -                   -
  ipv4-tos     96           class-3    -                   -
  ipv4-tos     128          class-4    -                   -
  ipv4-tos     160          class-5    -                   -
  ipv4-tos     192          class-6    -                   -
  ipv4-tos     224          class-7    -                   -
  ipv6-tc      0            class-0    -                   -
  ipv6-tc      32           class-1    -                   -
  ipv6-tc      64           class-2    -                   -
  ipv6-tc      96           class-3    -                   -
  ipv6-tc      128          class-4    -                   -
  ipv6-tc      160          class-5    -                   -
  ipv6-tc      192          class-6    -                   -
  ipv6-tc      224          class-7    -                   -
Classifier: residential-pbit-classifier
Active: True
  Match Type   Codepoint    Class      Remark Codepoint    Color
  ieee-802.1   0            class-0    -                   -
  ieee-802.1   1            class-1    -                   -
  ieee-802.1   2            class-2    -                   -
  ieee-802.1   3            class-3    -                   -
  ieee-802.1   4            class-4    -                   -
  ieee-802.1   5            class-5    -                   -
  ieee-802.1   6            class-6    -                   -
  ieee-802.1   7            class-7    -                   -
```

5.3. show qos classifier <classifier-name>

```
supervisor@rtbrick: op> show qos classifier residential-pbit-classifier
Classifier: residential-pbit-classifier
Active: True
  Match Type   Codepoint    Class      Remark Codepoint    Color
  ieee-802.1   0            class-0    -                   -
  ieee-802.1   1            class-1    -                   -
  ieee-802.1   2            class-2    -                   -
  ieee-802.1   3            class-3    -                   -
  ieee-802.1   4            class-4    -                   -
  ieee-802.1   5            class-5    -                   -
  ieee-802.1   6            class-6    -                   -
  ieee-802.1   7            class-7    -                   -
supervisor@rtbrick: op>
>>>>>>>> qos interface output
supervisor@rtbrick: op> show qos interface
Interface           Profile
ifl-0/0/10/100      pta_8queues_comp_on_S
ifl-0/0/10/200      pta_8queues_comp_on_S
ifl-0/0/10/300      pta_8queues_comp_on_S
```

5.4. show qos interface <interface-name>

```
supervisor@rtbrick: op> show qos interface ifl-0/0/10/200
Interface           Profile
ifl-0/0/10/200      pta_8queues_comp_on_S
supervisor@rtbrick: op>
```

5.5. show qos policer

```
supervisor@rtbrick: op> show qos policer
Policer: _DEFAULT_POLICER_50_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max
CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          33000          33000          -
-
  2        -              -              -              -              -
-
  3        -              -              -              -              -
-
  4        -              -              -              -              -
-
Policer: policer-residential
Active: True, Type: two-rate-three-color, Levels: 4, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max
CIR(Kbps)  Max PIR(Kbps)
  1        8000           8000           800            800            -
-
  2        -              -              -              -              -
-
  3        -              -              800            800            -
-
  4        -              -              800            800            -
-
```

5.6. show qos policer <policer-name>

```
supervisor@rtbrick: op> show qos policer policer-residential
Policer: policer-residential
Active: True, Type: two-rate-three-color, Levels: 4, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max
CIR(Kbps)  Max PIR(Kbps)
  1        8000           8000           800            800            -
-
  2        -              -              -              -              -
-
  3        -              -              800            800            -
-
  4        -              -              800            800            -
-
supervisor@rtbrick: op>
>>>>>>>>>> qos profile output
supervisor@rtbrick: op> show qos profile
Profile: lac_4queues_4classes
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: lac_4queues_M
   Class queue map: lac_4queues_M
   Remark map: -
   Class policer map: policer-map-l2tp
   Mulifield classifier: -
Profile: lac_4queues_4classes_L
   Classifier: residential-pbit-classifier
```

```
   Policer: policer-residential
   Scheduler map: lac_4queues_L
   Class queue map: lac_4queues_L
   Remark map: -
   Class policer map: policer-map-l2tp
   Mulifield classifier: -
Profile: lac_4queues_4classes_S
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: lac_4queues_S
   Class queue map: lac_4queues_S
   Remark map: -
   Class policer map: policer-map-l2tp
   Mulifield classifier: -
Profile: pta_4queues_comp_off
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: pta_4queues_comp_off_M
   Class queue map: pta_4queues_M
   Remark map: -
   Class policer map: policer-map-residential
   Mulifield classifier: -
Profile: pta_4queues_comp_off_L
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: pta_4queues_comp_off_L
   Class queue map: pta_4queues_L
   Remark map: -
   Class policer map: policer-map-residential
   Mulifield classifier: -
Profile: pta_4queues_comp_off_S
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: pta_4queues_comp_off_S
   Class queue map: pta_4queues_S
   Remark map: -
   Class policer map: policer-map-residential
   Mulifield classifier: -
Profile: pta_4queues_comp_on
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: pta_4queues_comp_on_M
   Class queue map: pta_4queues_M
   Remark map: -
   Class policer map: policer-map-residential
   Mulifield classifier: -
```

5.7. show qos profile <profile>

```
supervisor@rtbrick: op> show qos profile lac_4queues_4classes
Profile: lac_4queues_4classes
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: lac_4queues_M
   Class queue map: lac_4queues_M
   Remark map: -
   Class policer map: policer-map-l2tp
   Mulifield classifier: -
supervisor@rtbrick: op>
```

5.8. show qos queue

```
supervisor@rtbrick: op> show qos queue
Applied queues:
  Interface          Queue              Queue Size         Min Thres
Max Thres          Drop Prob          Shaper
  ifl-0/0/10/100     BE_S               240000             -
-                  -                  -
  ifl-0/0/10/100     LD_S               200000             -
-                  -                  shaper_LD
  ifl-0/0/10/100     LL_S               200000             -
-                  -                  shaper_LL
  ifl-0/0/10/100     VO_S               50000              -
-                  -                  shaper_VO
  ifl-0/0/10/200     BE_S               240000             -
-                  -                  -
  ifl-0/0/10/200     LD_S               200000             -
-                  -                  shaper_LD
  ifl-0/0/10/200     LL_S               200000             -
-                  -                  shaper_LL
  ifl-0/0/10/200     VO_S               50000              -
-                  -                  shaper_VO
  ifl-0/0/10/300     BE_S               240000             -
-                  -                  -
  ifl-0/0/10/300     LD_S               200000             -
-                  -                  shaper_LD
  ifl-0/0/10/300     LL_S               200000             -
-                  -                  shaper_LL
  ifl-0/0/10/300     VO_S               50000              -
-                  -                  shaper_VO
Configured queues:
  Queue                Queue Size           Min Thres            Max Thres
Drop Prob            Shaper
  BE_L                 375000               -                    -
-                    -
  BE_M                 375000               -                    -
-                    -
  BE_S                 240000               -                    -
-                    -
  CO_L                 312500               -                    -
-                    -
  CO_M                 156250               -                    -
-                    -
```

```
  CO_S                  50000               -                   -
-                     -
  IO_L                  312500              -                   -
-                     shaper_IO
  IO_M                  156250              -                   -
-                     shaper_IO
  IO_S                  50000               -                   -
-                     shaper_IO
  LD_L                  1250000             -                   -
-                     shaper_LD
  LD_M                  625000              -                   -
-                     shaper_LD
  LD_S                  200000              -                   -
-                     shaper_LD
  LL_L                  1250000             -                   -
-                     shaper_LL
  LL_M                  625000              -                   -
-                     shaper_LL
  LL_S                  200000              -                   -
-                     shaper_LL
  VO_L                  312500              -                   -
-                     shaper_VO
  VO_M                  156250              -                   -
-                     shaper_VO
  VO_S                  50000               -                   -
-                     shaper_VO
  free_6_L              375000              -                   -
-                     -
  free_6_M              375000              -                   -
-                     -
  free_6_S              240000              -                   -
-                     -
  free_7_L              375000              -                   -
-                     -
  free_7_M              375000              -                   -
-                     -
  free_7_S              240000              -                   -
-                     -
supervisor@rtbrick: op>
```

5.9. show qos queue <interface-name>

```
supervisor@rtbrick: op> show qos queue ifl-0/0/10/100
Applied queues:
  Interface          Queue              Queue Size         Min Thres
Max Thres          Drop Prob          Shaper
  ifl-0/0/10/100     BE_S               240000             -
-                  -                  -
  ifl-0/0/10/100     LD_S               200000             -
-                  -                  shaper_LD
  ifl-0/0/10/100     LL_S               200000             -
-                  -                  shaper_LL
  ifl-0/0/10/100     VO_S               50000              -
-                  -                  shaper_VO
supervisor@rtbrick: op>
```

5.10. show qos scheduler

```
supervisor@rtbrick: op> show qos scheduler
Scheduler              Type                   Shaper
Composite            Active
fff                    strict_priority        -                      False
False
fffd                   strict_priority        -                      False
False
lac_4queues            strict_priority        -                      True
False
olt-pon1               fair_queueing          -                      False
False
olt-pon10              fair_queueing          -                      False
False
olt-pon11              fair_queueing          -                      False
False
olt-pon12              fair_queueing          -                      False
False
olt-pon13              fair_queueing          -                      False
False
olt-pon14              fair_queueing          -                      False
False
olt-pon15              fair_queueing          -                      False
False
olt-pon16              fair_queueing          -                      False
False
olt-pon17              fair_queueing          -                      False
False
olt-pon18              fair_queueing          -                      False
False
olt-pon19              fair_queueing          -                      False
False
olt-pon2               fair_queueing          -                      False
False
olt-pon20              fair_queueing          -                      False
False
olt-pon21              fair_queueing          -                      False
False
olt-pon22              fair_queueing          -                      False
```

```
False
olt-pon23               fair_queueing          -                       False
False
olt-pon24               fair_queueing          -                       False
False
olt-pon25               fair_queueing          -                       False
False
olt-pon26               fair_queueing          -                       False
False
olt-pon27               fair_queueing          -                       False
False
olt-pon28               fair_queueing          -                       False
False
olt-pon29               fair_queueing          -                       False
False
olt-pon3                fair_queueing          -                       False
False
olt-pon30               fair_queueing          -                       False
False
olt-pon31               fair_queueing          -                       False
False
olt-pon32               fair_queueing          -                       False
False
olt-pon4                fair_queueing          -                       False
False
olt-pon5                fair_queueing          -                       False
False
olt-pon6                fair_queueing          -                       False
False
olt-pon7                fair_queueing          -                       False
False
olt-pon8                fair_queueing          -                       False
False
olt-pon9                fair_queueing          -                       False
False
pta_4queues_comp_off    strict_priority        -                       True
False
pta_4queues_comp_on     strict_priority        -                       True
False
pta_8queues_comp_off    strict_priority        -                       True
False
pta_8queues_comp_on     strict_priority        -                       True
False
supervisor@rtbrick: op>
```

5.11. show qos scheduler <scheduler-name>

```
supervisor@rtbrick: op> show qos scheduler  lac_4queues
Scheduler               Type                    Shaper
Composite            Active
lac_4queues             strict_priority        -                       True
False
```

5.12. show qos scheduler-map

```
supervisor@rtbrick: op> show qos scheduler-map
Scheduler-Map: lac_4queues_S
   Scheduler: fff                    Scheduler: strict_priority
     Queue: LD_S                     strict_priority_1
     Scheduler: pta_4queues_comp_off  Scheduler: strict_priority
       Queue: LL_S                     strict_priority_1
       Queue: VO_S                     strict_priority_0
   Scheduler: fffd                   Scheduler: strict_priority
     Queue: BE_S                     strict_priority_0
Scheduler-Map: schedmap-olt
   Scheduler: olt-pon1               Scheduler: fair_queueing
   Scheduler: olt-pon2               Scheduler: fair_queueing
   Scheduler: olt-pon3               Scheduler: fair_queueing
   Scheduler: olt-pon4               Scheduler: fair_queueing
   Scheduler: olt-pon5               Scheduler: fair_queueing
   Scheduler: olt-pon6               Scheduler: fair_queueing
   Scheduler: olt-pon7               Scheduler: fair_queueing
   Scheduler: olt-pon8               Scheduler: fair_queueing
   Scheduler: olt-pon9               Scheduler: fair_queueing
   Scheduler: olt-pon10              Scheduler: fair_queueing
   Scheduler: olt-pon11              Scheduler: fair_queueing
   Scheduler: olt-pon12              Scheduler: fair_queueing
   Scheduler: olt-pon13              Scheduler: fair_queueing
   Scheduler: olt-pon14              Scheduler: fair_queueing
   Scheduler: olt-pon15              Scheduler: fair_queueing
   Scheduler: olt-pon16              Scheduler: fair_queueing
   Scheduler: olt-pon17              Scheduler: fair_queueing
   Scheduler: olt-pon18              Scheduler: fair_queueing
   Scheduler: olt-pon19              Scheduler: fair_queueing
   Scheduler: olt-pon20              Scheduler: fair_queueing
   Scheduler: olt-pon21              Scheduler: fair_queueing
   Scheduler: olt-pon22              Scheduler: fair_queueing
   Scheduler: olt-pon23              Scheduler: fair_queueing
   Scheduler: olt-pon24              Scheduler: fair_queueing
   Scheduler: olt-pon25              Scheduler: fair_queueing
   Scheduler: olt-pon26              Scheduler: fair_queueing
   Scheduler: olt-pon27              Scheduler: fair_queueing
   Scheduler: olt-pon28              Scheduler: fair_queueing
   Scheduler: olt-pon29              Scheduler: fair_queueing
   Scheduler: olt-pon30              Scheduler: fair_queueing
   Scheduler: olt-pon31              Scheduler: fair_queueing
   Scheduler: olt-pon32              Scheduler: fair_queueing
   Scheduler: olt-pon33              Scheduler: False
Scheduler-Map: lac_4queues_L
   Scheduler: lac_4queues            Scheduler: strict_priority
     Queue: BE_L                     strict_priority_1
     Queue: LD_L                     strict_priority_1
     Queue: LL_L                     strict_priority_2
     Queue: VO_L                     strict_priority_0
```

5.13. show qos scheduler-map <scheduler-map>

```
supervisor@rtbrick: op> show qos scheduler-map lac_4queues_S
Scheduler-Map: lac_4queues_S
   Scheduler: fff                    Scheduler: strict_priority
     Queue: LD_S                     strict_priority_1
     Scheduler: pta_4queues_comp_off  Scheduler: strict_priority
       Queue: LL_S                     strict_priority_1
       Queue: VO_S                     strict_priority_0
   Scheduler: fffd                   Scheduler: strict_priority
     Queue: BE_S                     strict_priority_0
```

5.14. show qos shaper

```
supervisor@rtbrick: op> show qos shaper
Shaper              High Rate(Kbps)     Low Rate(Kbps)      High Burst(Kb)
Low Burst(Kb)       Active
pon-shaper          2488000             -                   -
-                   True
shaper_IO           -                   1000000             -
-                   True
shaper_LD           1000000             -                   -
-                   True
shaper_LL           1000000             -                   -
-                   True
shaper_VO           1000000             -                   -
-                   True
shaper_session      1000000             100                 -
-                   True
```

5.15. show qos shaper <shaper-name>

```
supervisor@rtbrick: op> show qos shaper shaper_session
Shaper              High Rate(Kbps)     Low Rate(Kbps)      High Burst(Kb)
Low Burst(Kb)       Active
shaper_session      1000000             100                 -
-                   False
```