Configuration Commands

Generic Commands

description

Syntax
description *description-string*
no description

Context
config>qos>shared-queue
config>qos>network-queue
config>qos>network
config>qos>network>ingress>ipv6-criteria>entry
config>qos>network>ingress>ip-criteria>entry
config>qos>sap-egress
config>qos>sap-ingress
config>qos>sap-ingress>ipv6-criteria>entry
config>qos>sap-ingress>ip-criteria>entry
config>qos>sap-ingress>mac-criteria>entry
config>qos>scheduler-policy
config>qos>scheduler-policy>tier>scheduler

Description
This command creates a text description stored in the configuration file for a configuration context.

The **description** command associates a text string with a configuration context to help identify the context in the configuration file.

The **no** form of this command removes any description string from the context.

Default
No description is associated with the configuration context.

Parameters
description-string — A text string describing the entity. Allowed values are any string up to 80 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, $, spaces, etc.), the entire string must be enclosed within double quotes.

Operational Commands

copy

Syntax	**copy network-queue** *src-name dst-name* [**overwrite**]
Context	config>qos
Description	This command copies or overwrites existing network queue QoS policies to another network queue policy ID.

The **copy** command is a configuration level maintenance tool used to create new policies using existing policies. It also allows bulk modifications to an existing policy with the use of the **overwrite** keyword.

Parameters **network-queue** — Indicates that the source policy ID and the destination policy ID are network-queue policy IDs. Specify the source policy ID that the copy command will attempt to copy from and specify the destination policy ID to which the command will copy a duplicate of the policy.

overwrite — specifies to replace the existing destination policy. Everything in the existing destination policy will be overwritten with the contents of the source policy. If **overwrite** is not specified, a message is generated saying that the destination policy ID exists.

```
SR7>config>qos# copy network-queue nq1 nq2
MINOR: CLI Destination "nq2" exists - use {overwrite}.
SR7>config>qos# copy network-queue nq1 nq2 overwrite
```

Network Queue QoS Policy Commands

network-queue

Syntax   [no] **network-queue** *policy-name*

Context   config>qos

Description   This command creates a context to configure a network queue policy. Network queue policies define the ingress network queuing at the MDA network node level and on the Ethernet port and SONET/SDH path level to define network egress queuing.

Default   default

Parameters   *policy-name —* The name of the network queue policy.

> **Values**   Valid names consist of any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, $, spaces, etc.), the entire string must be enclosed within double quotes.

fc

Syntax   [no] **fc** *fc-name*

Context   config>qos>network-queue

Description   The **fc** context in the network-queue context provides a forwarding class queue context to the contained buffer control and queue rate commands.

The **fc** node contains the PIR, CIR, CBS and MBS commands used to control the buffer pool resources of each forwarding class queue on the ingress and egress pools that are associated with the network-queue policy.

The **no** form of this command restores all PIR, CIR, CBS and MBS parameters for the forwarding class network queue to their default values.

Parameters   *fc-name —* The forwarding class name for which the contained PIR, CIR, CBS and MBS queue attributes apply. An instance of **fc** is allowed for each **fc-name**.

> **Values**   be, l2, af, l1, h2, ef, h1, nc

multicast-queue

Syntax **multicast-queue** *queue-id*
no multicast-queue

Context config>qos>network-queue>fc

Description This command overrides the default multicast forwarding type queue mapping for **fc** *fc-name*. The specified *queue-id* must exist within the policy as a multipoint queue before the mapping can be made. Once the forwarding class mapping is executed, all multicast traffic using this policy is forwarded using the *queue-id*.

The multicast forwarding type includes the **unknown** unicast forwarding type and the **broadcast** forwarding type unless each is explicitly defined to a different multipoint queue. When the unknown and broadcast forwarding types are left as default, they will track the defined queue for the multicast forwarding type.

The **no** form of the command sets the multicast forwarding type *queue-id* back to the default queue for the forwarding class. If the **broadcast** and **unknown** forwarding types were not explicitly defined to a multipoint queue, they will also be set back to the default multipoint queue (queue 11).

Resource Utilization:
When a multipoint queue is created and at least one forwarding class is mapped to the queue using the **multipoint-queue** command, a single ingress multipoint hardware queue is created per instance of the applied network-queue policy using the queue-policy command at the ingress network MDA level. Multipoint queues are not created at egress and the multipoint queues defined in the network-queue policy are ignored when the policy is applied to an egress port.

Parameters *queue-id —* The *queue-id* parameter specified must be an existing, multipoint queue defined in the **config>qos>network-queue>queue** context.

Values Any valid multipoint queue-ID in the policy including 2 through 16.

Default 11

queue

Syntax [**no**] **queue** *queue-id*

Context config>qos>network-queue>fc
config>qos>network-queue
config>qos>network-queue>hsmda

Description This command creates the context to configure forwarding-class to queue mappings.

Explicit definition of an ingress queue's hardware scheduler status is supported. A single ingress queue allows support for multiple forwarding classes. The default behavior automatically chooses the expedited or non-expedited nature of the queue based on the forwarding classes mapped to it. As long as all forwarding

classes mapped to the queue are expedited (`nc`, `ef`, `h1` or `h2`), the queue is treated as an expedited queue by the hardware schedulers. When any non-expedited forwarding classes are mapped to the queue (`be`, `af`, `l1` or `l2`), the queue is treated as best effort (`be`) by the hardware schedulers. The expedited hardware schedulers are used to enforce expedited access to internal switch fabric destinations. The hardware status of the queue must be defined at the time of queue creation within the policy.

The **no** form of this command removes the *queue-id* from the network-queue policy and from any existing MDA or port using the policy. If any forwarding class forwarding types are mapped to the queue, they revert to their default queues. When a queue is removed, any pending accounting information for each MDA or port queue created due to the definition of the queue in the policy is discarded.

Resource Utilization
When the network-queue policy is applied on a ingress MDA, each unicast queue is created multiple times - once for each switch fabric destination currently provisioned. Some IOM types represent one switch fabric destinations while others may represent two. At egress, a single queue is created since the policy is applied at the port level. Queues are only created when at least one forwarding class is mapped to the queue using the queue command within the forwarding class context.

Parameters   *queue-id —* The *queue-id* for the queue, expressed as an integer. The *queue-id* uniquely identifies the queue within the policy. This is a required parameter each time the queue command is executed.

   Values   1 — 32

queue-type — The **expedite**, **best-effort** and **auto-expedite** queue types are mutually exclusive to each other. Each defines the method that the system uses to service the queue from a hardware perspective. While parental virtual schedulers can be defined for the queue, they only enforce how the queue interacts for bandwidth with other queues associated with the same scheduler hierarchy. An internal mechanism that provides access rules when the queue is vying for bandwidth with queues in other virtual schedulers is also needed. A keyword must be specified at the time the queue is created in the network-queue policy. If an attempt to change the keyword after the queue is initially defined, an error is generated.

expedite — This keyword ensures that the queue is treated in an expedited manner independent of the forwarding classes mapped to the queue.

best-effort — This keyword ensures that the queue is treated in a non-expedited manner independent of the forwarding classes mapped to the queue.

auto-expedite — This keyword allows the system to auto-define the way the queue is serviced by the hardware. When **auto-expedite** is defined on the queue, the queue is treated in an expedited manner when all forwarding classes mapped to the queue are configured as expedited types `nc`, `ef`, `h1` or `h2`. When a single non-expedited forwarding class is mapped to the queue (`be`, `af`, `l1` and `l2`) the queue automatically falls back to non-expedited status.

   Values   expedite, best-effort, auto-expedite

   Default   auto-expedite

multipoint — This keyword specifies that this *queue-id* is for multipoint forwarded traffic only. This *queue-id* can only be explicitly mapped to the forwarding class multicast, broadcast, or unknown unicast

ingress traffic. If you attempt to map forwarding class unicast traffic to a multipoint queue, an error is generated and no changes are made to the current unicast traffic queue mapping.

A queue must be created as multipoint. The **multipoint** designator cannot be defined after the queue is created. If an attempt is made to modify the command to include the **multipoint** keyword, an error is generated and the command will not execute.

The **multipoint** keyword can be entered in the command line on a pre-existing multipoint queue to edit *queue-id* parameters.

Values   multipoint or not present

Default   Present (the queue is created as non-multipoint)

egress-hsmda

Syntax   **egress-hsmda**

Context   config>qos>network-queue

Description This command enables the context to configure queue definitions for use on HSMDAs.

queue

Syntax   [**no**] **queue** *queue-id*

Context   config>qos>network-queue>egress-hsmda

Description This command is a container for the configuration parameters controlling the behavior of an HSMDA queue. Unlike the standard QoS policy queue command, this command is not used to actually create or dynamically assign the queue to the object which the policy is applied. The queue identified by queue-id always exists whether the command is executed or not. In the case of HSMDA SAPs and subscribers, all eight queues exist at the moment the system allocates an HSMDA queue group to the object.

Best-Effort, Expedited and Auto-Expedite Queue Behavior Based on Queue-ID

With standard service queues, the scheduling behavior relative to other queues is based on two items, the queues Best-Effort or Expedited nature and the dynamic rate of the queue relative to the defined CIR. HSMDA queues are handled differently. The create time auto-expedite and explicit expedite and best-effort qualifiers have been eliminated and instead the scheduling behavior is based solely on the queues identifier. Queues with a queue-id equal to 1 are placed in scheduling class 1. Queues with queue-id 2 are placed in scheduling class 2. And so on up to scheduling class 8. Each scheduling class is either mapped directly to a strict scheduling priority level based on the class ID, or the class may be placed into a weighted scheduling class group providing byte fair weighted round robin scheduling between the members of the group. Two weighted groups are supported and each may contain up to three consecutive scheduling classes. The weighed group assumes its highest member class's inherent strict scheduling level for scheduling purposes. Strict priority level 8 has the highest priority while strict level 1 has the lowest. When grouping of scheduling classes is defined, some of the strict levels will not be in use.

Every HSMDA Queue Supports Profile Mode Implicitly

Unlike standard service queues, the HSMDA queues do not need to be placed into the special mode profile at create time in order to support ingress color aware policing. Each queue may handle in-profile, out-of-profile and profile undefined packets simultaneously. As with standard queues, the explicit profile of a packet is dependant on ingress sub-forwarding class to which the packet is mapped.

The **no** form of the command restores the defined queue-id to its default parameters. All HSMDA queues having the queue-id and associated with the QoS policy are re-initialized to default parameters.

Parameters   *queue-id* — Defines the context of which of the eight egress queues will be entered for editing purposes.

packet-byte-offset

Syntax   **packet-byte-offset** {**add** *add-bytes* | **subtract** *sub-bytes*}
            no packet-byte-offset

Context   config>qos>network-queue>egress-hsmda

Description This command adds or subtracts the specified number of bytes to the accounting function for each packet handled by the HSMDA queue. Normally, the accounting and leaky bucket functions are based on the 14-byte Ethernet DLC header, 4-byte or 8-byte VLAN tag (optional), 20-byte IP header, IP payload and the 4-byte CRC (everything except the preamble and inter-frame gap). As an example, the packet-byte-offset command can be used to add the frame encapsulation overhead (20 bytes) to the queues accounting functions. The accounting functions affected include:

- Offered High Priority / In-Profile Octet Counter
- Offered Low Priority / Out-of-Profile Octet Counter
- Discarded High Priority / In-Profile Octet Counter
- Discarded Low Priority / Out-of-Profile Octet Counter
- Forwarded In-Profile Octet Counter
- Forwarded Out-of-Profile Octet Counter
- Peak Information Rate (PIR) Leaky Bucket Updates
- Committed Information Rate (CIR) Leaky Bucket Updates
- Queue Group Aggregate Rate Limit Leaky Bucket Updates

The secondary shaper leaky bucket, scheduler priority level leaky bucket and the port maximum rate updates are not affected by the configured packet-byte-offset. Each of these accounting functions are frame based and always include the preamble, DLC header, payload and the CRC regardless of the configured byte offset.

The packet-byte-offset command accepts either add or subtract as valid keywords which define whether bytes are being added or removed from each packet traversing the queue. Up to 20 bytes may be added to the packet and up to 43 bytes may be removed from the packet. An example use case for subtracting bytes from each packet is an IP based accounting function. Given a Dot1Q encapsulation, the command packet-byte-offset subtract 14 would remove the DLC header and the Dot1Q header from the size of each packet for accounting functions only. The 14 bytes are not actually removed from the packet, only the accounting size of the packet is affected.

As inferred above, the variable accounting size offered by the packet-byte-offset command is targeted at the queue and queue group level. When the queue group represents the last-mile bandwidth constraints for a subscriber, the offset allows the HSMDA queue group to provide an accurate accounting to prevent overrun and underrun conditions for the subscriber. The accounting size of the packet is ignored by the secondary shapers, the scheduling priority level shapers and the scheduler maximum rate. The actual on-the-wire frame size is used for these functions to allow an accurate representation of the behavior of the subscribers packets on an Ethernet aggregation network.

The packet-byte-offset value may be overridden for the HSMDA queue at the network queue level.

The **no** form of the command removes any accounting size changes to packets handled by the queue. The command does not affect overrides that may exist on an associated with the queue.

Parameters   **add** *add-bytes* **—** Indicates that the byte value should be added to the packet for queue and queue group level accounting functions. Either the **add** or **subtract** keyword must be specified. The corresponding byte value must be specified when executing the packet-byte-offset command. The **add** keyword is mutually exclusive with the **subtract** keyword.

   Values   1 — 31

   subtract *sub-bytes* **—** Indicates that the byte value should be subtracted from the packet for queue and queue group level accounting functions. The **subtract** keyword is mutually exclusive with the **add** keyword. Either the **add** or **subtract** keyword must be specified. The corresponding byte value must be specified when executing the packet-byte-offset command. Note that the minimum resulting packet size used by the system is 64 bytes with an HS-MDA.

   Values   1 — 64

wrr-policy

   Syntax   **wrr-policy** *hsmda-wrr-policy-name*
   no wrr-policy

   Context   config>qos>network-queue>egress-hsmda

Description   This command associates an existing HSMDA weighted-round-robin (WRR) scheduling loop policy to the HSMDA queue.

Parameters   *hsmda-wrr-policy-name —* Specifies the existing HSMDA WRR policy name to associate to the queue.

slope-policy

   Syntax   **slope-policy** *hsmda-slope-policy-name*
   no slope-policy

   Context   config>qos>network-queue>egress-hsmda>queue

Description   This command associates an existing HSMDA slope policy to the QoS policy HSMDA queue. The specified hsmda-slope-policy-name must exist for the command to succeed. If the policy name does not exist, the command has no effect on the existing slope policy association. Once a slope policy is associated with a

QoS policy queue or override, the slope policy cannot be removed from the system. Any edits to an associated slope policy are immediately applied to the queues using the slope policy.

Within the ingress and egress QoS policies, packets are classified as high priority or low-priority. For color aware policies, packets are also potentially classified as in-profile, out-of-profile or profile-undefined. Based on these classifications, packets are mapped to the RED slopes in the following manner:

Ingress Slope Mapping

- In-Profile — High Slope (priority ignored)
- Profile-Undefined, High Priority — High Slope
- Out-of-Profile Low Slope (priority ignored)
- Profile-Undefined, Low Priority — Low Slope

Egress Slope Mapping

- In-Profile from ingress — High Slope
- Out-of-Profile from ingress — Low Slope

The specified policy contains a value that defines the queue's MBS value (queue-mbs). This is the maximum depth of the queue specified in bytes where all packets start to discard. The high and low priority RED slopes provide congestion control mechanisms that react to the current depth of the queue and start a random discard that increases in probability as the queue depth increases. The start point and end point for each discard probability slope is defined as follows:

- Start-Utilization — This is defined as percentage of MBS and specifies where the discard probability for the slope begins to rise above 0%. (A corresponding Start-Probability parameter is not needed as the start probability is always 0%.
- Maximum-Utilization — This is also defined as a percentage of MBS and specifies where (based on MBS utilized) the discard probability rises to 100%. This is the first portion of the knee coordinates and is meaningless without the Maximum-Probability parameter.
- Maximum-Probability — This is defined as a percentage of discard probability and in conjunction with maximum-utilization completes the knee coordinate where the discard probability deviates from the slope and rises to 100%.

Up to 1024 HSMDA slope policies may be configured on a system.

The system maintains a slope policy named **hsmda-default** which acts as a default policy when an explicit slope policy has not been defined for an HSMDA queue. The default policy may be edited, but it cannot be deleted. If a no slope-policy hsmda-default command is executed, the default slope policy returns to the factory default settings. The factory default settings are as follows:

High Slope:

- Start-Utilization 100%
- Max-Utilization 100%
- Max-Probability 100%
- Shutdown

Low Slope:

- Start-Utilization 90%

- Max-Utilization 90%

- Max-Probability 1

- No Shutdown

Time-Average-Factor: 0

The **no** form of the command restores the association between the queue and the HSMDA default slope policy. The command has no immediate effect for queues that have a local override defined for the slope policy.

Parameters *hsmda-slope-policy-name —* Specifies an existing slope policy within the system. If a slope policy with the specified name does not exist, the slope-policy command will fail without modifying the slope behavior on the queue. Once a slope policy is associated with an HSMDA queue, the policy cannot be deleted.

> **Default** hsmda-default

adaptation-rule

Syntax **adaptation-rule** [**pir** {**max** | **min** | **closest**}]
no adaptation-rule

Context config>qos>network-queue>egress-hsmda>queue

Description This command defines the method used by the system to derive the operational PIR settings when the HSMDA queue is provisioned in hardware. For the PIR parameters individually, the system attempts to find the best operational rate depending on the defined constraint.

The **no** form of the command removes any explicitly defined constraints used to derive the operational PIR created by the application of the policy. When a specific **adaptation-rule** is removed, the default constraints for **pir** apply.

Parameters *adaptation-rule —* Specifies the adaptation rule to be used while computing the operational PIR value.

> **Values** **pir** — Defines the constraints enforced when adapting the PIR rate defined. The **pir** parameter requires a qualifier that defines the constraint used when deriving the operational PIR for the HSMDA queue. When the **pir** command is not specified, the default applies.
>
> **max** — The **max** (maximum) option is mutually exclusive with the **min** and **closest** options.
>
> **min** — The **min** (minimum) option is mutually exclusive with the **max** and **closest** options.

> **Default** **closest** — The **closest** parameter is mutually exclusive with the **min** and **max** parameter. When **closest** is defined, the operational PIR for the HSMDA queue will be the rate closest to the rate specified using the **rate** command

mbs

Syntax	**mbs** {*size* [**byte | kilobytes**] | **default**} **no mbs**
Context	config>qos>network-queue>egress-hsmda>queue
Description	The Maximum Burst Size (**mbs**) command specifies the relative amount of the buffer pool space for the maximum buffers for a specific HSMDA queue.
	The MBS value is used to by a queue to determine whether it has exhausted its total allowed buffers while enqueuing packets. Once the queue has exceeded its maximum amount of buffers, all packets are discarded until the queue transmits a packet. A queue that has not exceeded its MBS size is not guaranteed that a buffer will be available when needed or that the packet's RED slope will not force the discard of the packet. Setting proper CBS parameters and controlling CBS oversubscription is one major safeguard to queue starvation (when a queue does not receive its fair share of buffers). Another is properly setting the RED slope parameters for the needs of the network queues.
	The **no** form of the command returns the MBS size for the queue to the default for the forwarding class.
Parameters	*size* — Specifies the size of the MBS for the HSMDA queue.

> **Values**   0 — 2688000 (for bytes)
>
> **Values**   0 — 2625 (for kilobytes)

        bytes — Identifies the size value in terms of bytes.

        kilobytes — Identifies the size value in terms of kilobytes.

        default — Specifies the size of the MBS to be the default value configured on the system.

percent-rate

Syntax	**percent-rate** *percentage* **no percent-rate**
Context	config>qos>network-queue>egress-hsmda>queue
Description	This command specifies the PIR shaping rate for the HSMDA queue.
	The **no** form of the command returns the PIR size for the queue to the default value.
Parameters	*percentage* — Specifies the PIR percentage rate for the HSMDA queue.

> **Values**   0.10 — 100.00

wrr-weight

Syntax	**wrr-weight** *value* **no wrr-weight**
Context	config>qos>network-queue>egress-hsmda>queue
Description	This command assigns the weight value to the HSMDA queue.
	The **no** form of the command returns the weight value for the queue to the default value.
Parameters	*percentage* — Specifies the weight for the HSMDA queue.
	Values 1— 32

Network Queue QoS Policy Queue Commands

queue

Syntax   **queue** *queue-id* [**multipoint**] [*queue-type*] [*queue-mode*] **pool** *pool-name*
           no queue *queue-id*

Context   config>qos>network-queue

Description   This command enables the context to configure a QoS network-queue policy queue.

Explicit definition of an ingress queue's hardware scheduler status is supported. A single ingress queue allows support for multiple forwarding classes. The default behavior automatically chooses the expedited or non-expedited nature of the queue based on the forwarding classes mapped to it. As long as all forwarding classes mapped to the queue are expedited (`nc`, `ef`, `h1` or `h2`), the queue is treated as an expedited queue by the hardware schedulers. When any non-expedited forwarding classes are mapped to the queue (`be`, `af`, `l1` or `l2`), the queue is treated as best effort (`be`) by the hardware schedulers. The expedited hardware schedulers are used to enforce expedited access to internal switch fabric destinations. The hardware status of the queue must be defined at the time of queue creation within the policy.

The **queue** command allows the creation of multipoint queues. Only multipoint queues can receive ingress packets that need flooding to multiple destinations. By separating the unicast for multipoint traffic at service ingress and handling the traffic on separate multipoint queues, special handling of the multipoint traffic is possible. Each queue acts as an accounting and (optionally) shaping device offering precise control over potentially expensive multicast, broadcast and unknown unicast traffic. Only the back-end support of multipoint traffic (between the forwarding class and the queue based on forwarding type) needs to be defined. The individual classification rules used to place traffic into forwarding classes are not affected. Queues must be defined as multipoint at the time of creation within the policy.

The multipoint queues are for multipoint traffic.

The **no** form of this command removes the *queue-id* from the network-queue policy and from any existing SAPs using the policy. If any forwarding class forwarding types are mapped to the queue, they revert to their default queues. When a queue is removed, any pending accounting information for each SAP queue created due to the definition of the queue in the policy is discarded.

If the specified pool-name does not exist on the MDA, the queue will be treated as 'pool orphaned' and will be mapped to the appropriate default pool. Once the pool comes into existence on the MDA or port, the queue will be mapped to the new pool.

Once the queue is created within the policy, the pool command may be used to either remove the queue from the pool, or specify a new pool name association for the queue. The pool command

does not appear in save or show command output. Instead, the current pool name for the queue will appear (or not appear) on the queue command output using the pool keyword.

Parameters

queue-id — The *queue-id* for the queue, expressed as an integer. The *queue-id* uniquely identifies the queue within the policy. This is a required parameter each time the queue command is executed.

Values 1 — 32

queue-type — The **expedite**, **best-effort** and **auto-expedite** queue types are mutually exclusive to each other. Each defines the method that the system uses to service the queue from a hardware perspective. While parental virtual schedulers can be defined for the queue, they only enforce how the queue interacts for bandwidth with other queues associated with the same scheduler hierarchy. An internal mechanism that provides access rules when the queue is vying for bandwidth with queues in other virtual schedulers is also needed. A keyword must be specified at the time the queue is created in the network-queue policy. If an attempt is made to change the keyword after the queue is initially defined, an error is generated.

expedite — This keyword ensures that the queue is treated in an expedited manner independent of the forwarding classes mapped to the queue.

best-effort — This keyword ensures that the queue is treated in a non-expedited manner independent of the forwarding classes mapped to the queue.

auto-expedite — This keyword allows the system to auto-define the way the queue is serviced by the hardware. When **auto-expedite** is defined on the queue, the queue is treated in an expedited manner when all forwarding classes mapped to the queue are configured as expedited types nc, ef, h1 or h2. When a single non-expedited forwarding class is mapped to the queue (be, af, l1 and l2) the queue automatically falls back to non-expedited status.

Values expedite, best-effort, auto-expedite

Default auto-expedite

multipoint — This keyword specifies that this *queue-id* is for multipoint forwarded traffic only. This *queue-id* can only be explicitly mapped to the forwarding class multicast, broadcast, or unknown unicast ingress traffic. If you attempt to map forwarding class unicast traffic to a multipoint queue, an error is generated and no changes are made to the current unicast traffic queue mapping.

A queue must be created as multipoint. The **multipoint** designator cannot be defined after the queue is created. If an attempt is made to modify the command to include the **multipoint** keyword, an error is generated and the command will not execute.

The **multipoint** keyword can be entered in the command line on a pre-existing multipoint queue to edit *queue-id* parameters.

Values multipoint or not present

Default Not present (the queue is created as non-multipoint)

queue-mode — Specifies the mode in which the queue is operating. This attribute is associated with the queue at the time of creation and cannot be modified thereafter.

Values **profile-mode**: When the queue is operating in the profile mode (or, the color aware mode), the queue tries to provide the appropriate bandwidth to the packets with different

profiles. The profiles are assigned according to the configuration of the forwarding class or the sub-forwarding class.

priority-mode: The queue is capable of handling traffic differently with two distinct priorities. These priorities are assigned by the stages preceding the queueing framework in the system. In priority mode, the queue does not have the functionality to support the profiled traffic and in such cases the queue will have a degraded performance. However, the converse is not valid and a queue in profile mode should be capable of supporting the different priorities of traffic.

Default   **priority-mode**

pool-name — The specified pool-name identifies a named pool where the policy will be applied. Each queue created within the system is tied to a physical port. When the policy is applied and the queue is created, the system will scan the named pools associated with the port to find the specified pool name. If the pool is not found on the port, the system will then look at named pools defined at the ports MDA level. If the pool name is not found on either the port or MDA, the queue will be marked as 'pool-orphaned' and will be mapped to the appropriate default pool. If the pool comes into existence, the queue will be moved from the default pool to the new named pool and the 'pool-orphaned' state will be cleared. The specified name must be an ASCII name string up to 16 characters long.

Values   Any valid ASCII name string

Default   None

The queue's pool association may only be removed by either re-executing the queue command without the pool keyword or by executing the no pool command within the queue's CLI context. When the pool name is removed, the queue will be placed on the appropriate default pool.

adaptation-rule

Syntax   **adaptation-rule** [**pir** *adaptation-rule*] [**cir** *adaptation-rule*]
no adaptation-rule

Context   config>qos>network-queue>queue

Description   This command defines the method used by the system to derive the operational CIR and PIR settings when the queue is provisioned in hardware. For the CIR and PIR parameters individually, the system attempts to find the best operational rate depending on the defined constraint.

The **no** form of the command removes any explicitly defined constraints used to derive the operational CIR and PIR created by the application of the policy. When a specific **adaptation-rule** is removed, the default constraints for **pir** and **cir** apply.

Default   adaptation-rule pir closest cir closest

Parameters   *adaptation-rule —* Specifies the adaptation rule to be used while computing the operational CIR or PIR value.

Values   **pir** — Defines the constraints enforced when adapting the PIR rate defined within the **queue** *queue-id* **rate** command. The **pir** parameter requires a qualifier that defines the

constraint used when deriving the operational PIR for the queue. When the **pir** command is not specified, the default applies.

cir — Defines the constraints enforced when adapting the CIR rate defined within the **queue** queue-id **rate** command. The **cir** parameter requires a qualifier that defines the constraint used when deriving the operational CIR for the queue. When the **cir** parameter is not specified, the default constraint applies.

max — The **max** (maximum) option is mutually exclusive with the **min** and **closest** options. When **max** is defined, the operational PIR for the queue will be equal to or less than the administrative rate specified using the **rate** command.

min — The **min** (minimum) option is mutually exclusive with the **max** and **closest** options. When **min** is defined, the operational PIR for the queue will be equal to or greater than the administrative rate specified using the **rate** command.

closest — The **closest** parameter is mutually exclusive with the **min** and **max** parameter. When **closest** is defined, the operational PIR for the queue will be the rate closest to the rate specified using the **rate** command.

avg-frame-overhead

Syntax   **avg-frame-overhead** *percent*
              no avg-frame-overhead

Context   config>qos>network-queue>queue

Description This command configures the average frame overhead to define the average percentage that the offered load to a queue will expand during the frame encapsulation process before sending traffic on-the-wire. While the avg-frame-overhead value may be defined on any queue, it is only used by the system for queues that egress a SONET or SDH port or channel. Queues operating on egress Ethernet ports automatically calculate the frame encapsulation overhead based on a 20 byte per packet rule (8 bytes for preamble and 12 bytes for Inter-Frame Gap).

When calculating the frame encapsulation overhead for port scheduling purposes, the system determines the following values:

- Offered-Load — The offered-load of a queue is calculated by starting with the queue depth in octets, adding the received octets at the queue and subtracting queue discard octets. The result is the number of octets the queue has available to transmit. This is the packet-based offered-load.

- Frame-encapsulation overhead — Using the avg-frame-overhead parameter, the frame-encapsulation overhead is simply the queue's current offered-load (how much has been received by the queue) multiplied by the avg-frame-overhead. If a queue had an offered load of 10,000 octets and the avg-frame-overhead equals 10%, the frame-encapsulation overhead would be 10,000 x 0.1 or 1,000 octets.

For egress Ethernet queues, the frame-encapsulation overhead is calculated by multiplying the number of offered-packets for the queue by 20 bytes. If a queue was offered 50 packets then the frame-encapsulation overhead would be 50 x 20 or 1,000 octets.

- Frame-based offered-load — The frame-based offered-load is calculated by adding the offered-load to the frame-encapsulation overhead. If the offered-load is 10,000 octets and the encapsulation overhead was 1,000 octets, the frame-based offered-load would equal 11,000 octets.

- Packet to frame factor — The packet-to-frame factor is calculated by dividing the frame-encapsulation overhead by the queue's offered-load (packet based). If the frame-encapsulation overhead is 1,000 octets and the offered-load is 10,000 octets then the packet to frame factor would be 1,000 / 10,000 or 0.1. When in use, the avg-frame-overhead will be the same as the packet to frame factor making this calculation unnecessary.

- Frame-based CIR — The frame-based CIR is calculated by multiplying the packet to frame factor with the queue's-configured CIR and then adding that result to that CIR. If the queue CIR is set at 500 octets and the packet to frame factor equals 0.1, the frame-based CIR would be 500 x 1.1 or 550 octets.

- Frame-based within-cir offered-load — The frame-based within-cir offered-load is the portion of the frame-based offered-load considered to be within the frame-based CIR. The frame-based within-cir offered-load is the lesser of the frame-based offered-load and the frame-based CIR. If the frame-based offered-load equaled 11000 octets and the frame-based CIR equaled 550 octets, the frame-based within-cir offered-load would be limited to 550 octets. If the frame-based offered-load equaled 450 octets and the frame-based CIR equaled 550 octets, the frame-based within-cir offered-load would equal 450 octets (or the entire frame-based offered-load).

As a special case, when a queue or associated intermediate scheduler is configured with a CIR-weight equal to 0, the system automatically sets the queue's frame-based within-cir offered-load to 0, preventing it from receiving bandwidth during the port scheduler's within-cir pass.

- Frame-based PIR — The frame-based PIR is calculated by multiplying the packet to frame-factor with the queue's-configured PIR and then adding the result to that PIR. If the queue PIR is set to 7500 octets and the packet to frame-factor equals 0.1, the frame-based PIR would be 7,500 x 1.1 or 8,250 octets.

- Frame-based within-pir offered-load — The frame-based within-pir offered-load is the portion of the frame-based offered-load considered to be within the frame-based PIR. The frame-based within-pir offered-load is the lesser of the frame-based offered-load and the frame-based PIR. If the frame-based offered-load equaled 11,000 octets and the frame-based PIR equaled 8250 octets, the frame-based within-pir offered-load would be limited to 8,250 octets. If the frame-based offered-load equaled 7,000 octets and the frame-based PIR equaled 8,250 octets, the frame-based within-pir offered load would equal 7,000 octets.

Port Scheduler Operation Using Frame Transformed Rates — The port scheduler uses the frame based rates to figure the maximum rates that each queue may receive during the within-cir and

above-cir bandwidth allocation passes. During the within-cir pass, a queue may receive up to its frame based within-cir offered-load. The maximum it may receive during the above-cir pass is the difference between the frame based within-pir offered load and the amount of actual bandwidth allocated during the within-cir pass.

SAP and Subscriber SLA-Profile Average Frame Overhead Override — The average frame overhead parameter on a sap-egress may be overridden at an individual egress queue basis. On each SAP and within the sla-profile policy used by subscribers an avg-frame-overhead command may be defined under the queue-override context for each queue. When overridden, the queue instance will use its local value for the average frame overhead instead of the sap-egress-defined overhead.

The **no** form of this command restores the average frame overhead parameter for the queue to the default value of 0 percent. When set to 0, the system uses the packet based queue statistics for calculating port scheduler priority bandwidth allocation. If the no avg-frame-overhead command is executed in a queue-override queue id context, the avg-frame-overhead setting for the queue within the sap-egress QoS policy takes effect.

Default **0**

Parameters *percent —* This parameter sets the average amount of packet-to-frame encapsulation overhead expected for the queue. This value is not used by the system for egress Ethernet queues.

 Values 0.00 — 100.00

cbs

Syntax **cbs** *percent*
no cbs

Context config>qos>network-queue>queue

Description The Committed Burst Size (**cbs**) command specifies the relative amount of reserved buffers for a specific ingress network MDA forwarding class queue or egress network port forwarding class queue. The value is entered as a percentage.

The CBS for a queue is used to determine whether it has exhausted its reserved buffers while enqueuing packets. Once the queue has exceeded the amount of buffers considered in reserve for this queue, it must contend with other queues for the available shared buffer space within the buffer pool. Access to this shared pool space is controlled through Random Early Detection (RED) slope application.

Two RED slopes are maintained in each buffer pool. A high priority slope is used by in-profile packets. A low priority slope is used by out-of-profile packets. All Network-Control and Management packets are considered in-profile. Assured packets are handled by their in-profile and out-of-profile markings. All Best-Effort packets are considered out-of-profile. Premium queues should be configured such that the CBS percent is sufficient to prevent shared buffering of

packets. This is generally taken care of by the CIR scheduling of Premium queues and the overall small amount of traffic on the class. Premium queues in a properly designed system will drain before all others, limiting their buffer utilization.

The RED slopes will detect congestion conditions and work to discard packets and slow down random TCP session flows through the queue. The RED slope definitions can be defined, modified or disabled through the network-queue policy assigned to the MDA for the network ingress buffer pool or assigned to the network port for network egress buffer pools.

The resultant CBS size can be larger than the MBS. This will result in a portion of the CBS for the queue to be unused and should be avoided.

The **no** form of this command returns the CBS size for the queue to the default for the forwarding class.

Special Cases   **Forwarding Class Queue on Egress Network Port or Channel —** For network egress, each forwarding class is supported by an egress queue on a per network port basis. These forwarding class-based queues are automatically created once a port or channel is placed in the network mode. The configuration parameters for each queue come from the applied egress network-queue policy on the network port or channel.

The **cbs** value is used to calculate the queue's CBS size based on the total amount of buffer space allocated for the buffer pool on the egress network port or channel. This buffer pool size will dynamically fluctuate based on the port or channel's egress pool size setting.

The total reserved buffers based on the total percentages can exceed 100 percent. This might not be desirable and should be avoided as a rule of thumb. If the total percentage equals or exceeds 100 percent of the buffer pool size, no buffers will be available in the shared portion of the pool. Any queue exceeding its CBS size will experience a hard drop on all packets until it drains below this threshold.

Forwarding Class Queue on Ingress MDA — For network ingress, each forwarding class is supported by an ingress queue per MDA. These forwarding class queues are automatically created once a single port or channel is placed in the network mode on the MDA and are removed once all network ports or channels are removed from the MDA (defined as access). The configuration parameters for each queue come from the applied ingress policy under the network context of the MDA.

The **cbs** value is used to calculate the queue's CBS size based on the total amount buffer space allocated for the network ingress buffer pool on the MDA. This buffer pool will dynamically fluctuate based on the sum of all ingress pool sizes for all network ports and channels on the MDA.

The total reserved buffers based on the total percentages can exceed 100 percent. This might not be desirable and should be avoided as a rule of thumb. If the total percentage equals or exceeds 100 percent of the buffer pool size, no buffers will be available in the shared portion of the pool.

Any queue exceeding its CBS size will experience a hard drop on all packets until it drains below this threshold.

Default The **cbs** forwarding class defaults are listed in the table below:

Table 24: cbs forwarding class defaults

Forwarding Class	Fowarding Class Label	Default CBS
Network-Control	nc	3
High-1	h1	3
Expedited	ef	1
High-2	h2	1
Low-1	l1	3
Assured	af	1
Low-2	l2	3
Best-Effort	be	1

Parameters *percent* — The percent of buffers reserved from the total buffer pool space, expressed as a decimal integer. If 10 MB is the total buffers in the buffer pool, a value of 10 would reserve 1MB (10%) of buffer space for the forwarding class queue. The value 0 specifies that no reserved buffers are required by the queue (a minimal reserved size can be applied for scheduling purposes).

Values 0 — 100

high-prio-only

Syntax **high-prio-only** *percent*
no high-prio-only

Context config>qos>network-queue>queue

Description The **high-prio-only** command allows the reservation of queue buffers for use exclusively by high priority packets as a default condition for access buffer queues for this network queue policy.

The difference between the MBS size for the queue and the high priority reserve defines the threshold where low priority traffic will be discarded. The result is used on the queue to define a threshold where low priority packets are discarded, leaving the rest of the default MBS size for high priority packets only. If the current MBS for the queue is 10MBytes, a value of 5 will result in a high priority reserve on the queue of 500KBytes. A value of 0 specifies that none of the MBS of

the queue will be reserved for high priority traffic. This does not affect RED slope operation for packets attempting to be queued.

Modifying the current MBS for the queue through the **mbs** command will cause the default **high-prio-only** function to be recalculated and applied to the queue. The **high-prio-only** command as defined for the specific queue can be used to override the default **high-prio-only** setting as defined in the network queue policy. This prevents the **high-prio-only** command for the network queue policy from having an affect on the queue.

The **no** form of this command restores the default value.

Default The **high-prio-only** forwarding class defaults are listed in the table below.

Table 25: High-prio-only forwarding class defaults

Forwarding Class	Fowarding Class Label	Default high-prio-only
Network-Control	nc	10
High-1	h1	10
Expedited	ef	10
High-2	h2	10
Low-1	l1	10
Assured	af	10
Low-2	l2	10
Best-Effort	be	10

Parameters *percent* — The amount of queue buffer space, expressed as a decimal percentage of the MBS.

    Values   0 — 100, default

mbs

Syntax **mbs** *percent*
no mbs

Context config>qos>network-queue>queue

Description The Maximum Burst Size (**mbs**) command specifies the relative amount of the buffer pool space for the maximum buffers for a specific ingress network MDA forwarding class queue or egress network port forwarding class queue. The value is entered as a percentage.

The MBS value is used to by a queue to determine whether it has exhausted its total allowed buffers while enqueuing packets. Once the queue has exceeded its maximum amount of buffers, all packets are discarded until the queue transmits a packet. A queue that has not exceeded its MBS size is not guaranteed that a buffer will be available when needed or that the packet's RED slope will not force the discard of the packet. Setting proper CBS parameters and controlling CBS oversubscription is one major safeguard to queue starvation (when a queue does not receive its fair share of buffers). Another is properly setting the RED slope parameters for the needs of the network queues.

The MBS size can sometimes be smaller than the CBS. This will result in a portion of the CBS for the queue to be unused and should be avoided.

The **no** form of the command returns the MBS size for the queue to the default for the forwarding class.

Special Cases **Forwarding Class Queue on Egress Network Port or Channel —** For network egress, each forwarding class is supported by an egress queue on a per network port basis. These forwarding class-based queues are automatically created once a port or channel is placed in the network mode. The configuration parameters for each queue come from the applied egress policy on the network port or channel.

The **mbs** value is used to calculate the queue's MBS size based on the total amount buffer space allocated for the buffer pool on the egress network port or channel. This buffer pool size will dynamically fluctuate based on the port or channels egress pool size setting.

The total MBS settings for all network egress queues on the port or channel based on the total percentages can exceed 100 percent. Some oversubscription can be desirable to allow exceptionally busy forwarding classes more access to buffer space. The proper use of CBS settings will ensure that oversubscribing MBS settings will not starve other queues of buffers when needed.

Forwarding Class Queue on Ingress MDA — For network ingress, each forwarding class is supported by an ingress queue per MDA. These forwarding class queues are automatically created once a single port or channel is placed in the network mode on the MDA and are removed once all network ports or channels are removed from the MDA (defined as access). The configuration parameters for each queue come from the applied ingress policy under the network context of the MDA.

The **mbs** value is used to calculate the queue's MBS size based on the total amount buffer space allocated for the network ingress buffer pool on the MDA. This buffer pool will dynamically fluctuate based on the sum of all ingress pool sizes for all network ports and channels on the MDA.

The total MBS settings for all network egress queues on the port or channel based on the total percentages can exceed 100 percent. Some oversubscription can be desirable to allow exceptionally busy forwarding classes more access to buffer space. The proper use of CBS settings

will ensure that oversubscribing MBS settings will not starve other queues of buffers when needed.

Parameters *percent —* The percent of buffers from the total buffer pool space for the maximum amount of buffers, expressed as a decimal integer. If 10 MB is the total buffers in the buffer pool, a value of 10 would limit the maximum queue size to 1MB (10%) of buffer space for the forwarding class queue. If the total size is increased to 20MB, the existing value of 10 would automatically increase the maximum size of the queue to 2MB.

> **Values**   0 — 100

pool

Syntax **pool** *pool-name* [**create**]
no pool *pool-name*

Context config>qos>network-queue>queue

Description This command is utilized once the queue is created within the policy. The pool command may be used to either remove the queue from the pool, or specify a new pool name association for the queue. The pool command does not appear in save or show command output. Instead, the current pool name for the queue will appear (or not appear) on the queue command output using the pool keyword.

The **no** form of the command removes a named pool association for the queue. When the pool name is removed, the queue will be placed on the appropriate default pool.

Parameters *pool-name —* The specified *pool-name* identifies a named pool where the policy will be applied. Each queue created within the system is tied to a physical port. When the policy is applied and the queue is created, the system will scan the named pools associated with the port to find the specified pool name. If the pool is not found on the port, the system will then look at named pools defined at the ports MDA level. If the pool name is not found on either the port or MDA, the queue will be marked as 'pool-orphaned' and will be mapped to the appropriate default pool. If the pool comes into existence, the queue will be moved from the default pool to the new named pool and the 'pool-orphaned' state will be cleared. The specified name must be an ASCII name string up to 32 characters long.

Default None

port-parent

Syntax **port-parent** [**weight** *weight*] [**level** *level*] [**cir-weight** *cir-weight*] [**cir-level** *cir-level*]
no port-parent

Context config>qos>network-queue>queue

Description This command specifies whether this queue feeds off a port-level scheduler. For the network-queue policy context, only the port-parent command is supported. When a port scheduler exists on

the port, network queues without a port-parent association will be treated as an orphan queue on the port scheduler and treated according to the current orphan behavior on the port scheduler. If the port-parent command is defined for a network queue on a port without a port scheduler defined, the network queue will operate as if a parent association does not exist. Once a port scheduler policy is associated with the egress port, the port-parent command will come into effect.

When a network-queue policy is associated with an MDA or CMA for ingress queue definition, the port-parent association of the queues are ignored.

The **no** form of this command removes a port scheduler parent association for the queue or scheduler. If a port scheduler is defined on the port which the queue or scheduler instance exists, the queue or scheduler will become orphaned.

Default   no port-parent

Parameters   **weight** *weight* — Defines the weight the queue or scheduler will use at the above-cir port priority level (defined by the level parameter).

> **Values**   0 — 100
>
> **Default**   1

level *level* — Defines the port priority the queue or scheduler will use to receive bandwidth for its above-cir offered-load.

> **Values**   1 — 8 (8 is the highest priority)
>
> **Default**   1

cir-weight *cir-weight* — Defines the weight the queue or scheduler will use at the within-cir port priority level (defined by the cir-level parameter). The weight is specified as an integer value from 0 to 100 with 100 being the highest weight. When the cir-weight parameter is set to a value of 0, the queue or scheduler does not receive bandwidth during the port scheduler's within-cir pass and the cir-level parameter is ignored. If the cir-weight parameter is 1 or greater, the cir-level parameter comes into play.

> **Values**   0 — 100
>
> **Default**   1

cir-level *cir-level* — Defines the port priority the queue or scheduler will use to receive bandwidth for its within-cir offered-load. If the cir-weight parameter is set to a value of 0 (the default value), the queue or scheduler does not receive bandwidth during the port scheduler's within-cir pass and the cir-level parameter is ignored. If the cir-weight parameter is 1 or greater, the cir-level parameter comes into play.

> **Values**   0 — 8 (8 is the highest priority)
>
> **Default**   0

rate

Syntax   **rate** *percent* [**cir** *percent*]
        no rate

Context   config>qos>network-queue>queue

Description   This command defines the administrative Peak Information Rate (PIR) and the administrative Committed Information Rate (CIR) parameters for the queue. The PIR defines the percentage that the queue can transmit packets through the switch fabric (for SAP ingress queues) or out an egress interface (for SAP egress queues). Defining a PIR does not necessarily guarantee that the queue can transmit at the intended rate. The actual rate sustained by the queue can be limited by oversubscription factors or available egress bandwidth.

The CIR defines the percentage at which the system prioritizes the queue over other queues competing for the same bandwidth. For SAP ingress, the CIR also defines the rate that packets are considered in-profile by the system. In-profile packets are preferentially queued by the system at egress and at subsequent next hop nodes where the packet can traverse. To be properly handled as in- or out-of-profile throughout the network, the packets must be marked accordingly for profiling at each hop.

The CIR can be used by the queue's parent commands *cir-level* and *cir-weight* parameters to define the amount of bandwidth considered to be committed for the child queue during bandwidth allocation by the parent scheduler.

The **rate** command can be executed at anytime, altering the PIR and CIR rates for all queues created through the association of the SAP ingress or SAP egress QoS policy with the *queue-id*.

The **no** form of the command returns all queues created with the *queue-id* by association with the QoS policy to the default PIR and CIR parameters (100, 0).

Parameters   **cir** *percent* — Defines the percentage of the guaranteed rate allowed for the queue. When the **rate** command is executed, a valid PIR setting must be explicitly defined. When the **rate** command has not been executed, the default PIR of **100** is assumed. Fractional values are not allowed and must be given as a positive integer.

The actual PIR rate is dependent on the queue's **adaptation-rule** parameters and the actual hardware where the queue is provisioned.

        Values   0 — 100

        Default   100

    cir *percent* — Defines the percentage of the maximum rate allowed for the queue. When the **rate** command is executed, a CIR setting is optional. When the **rate** command has not been executed or the **cir** parameter is not explicitly specified, the default CIR (0) is assumed. Fractional values are not allowed and must be given as a positive integer.

        Values   0 — 100

        Default   0

Show Commands

network-queue

Syntax	**network-queue** [*network-queue-policy-name*] [**detail**]
Description	This command displays network queue policy information.
Context	show>qos
Parameters	*network-queue-policy-name* — The name of the network queue policy.

        Values Valid names consist of any string up to 32 characters long composed of printable, 7-bit ASCII characters. If the string contains special characters (#, $, spaces, etc.), the entire string must be enclosed within double quotes.

    detail — Includes each queue's rates and adaptation-rule and & cbs details. It also shows FC to queue mapping details.

Table 26: Network Queue Labels and Descriptions

Label	Description
Policy	The policy name that uniquely identifies the policy.
Description	A text string that helps identify the policy's context in the configuration file.
Port-Id	Displays the physical port identifier where the network queue policy is applied.
Queue	Displays the queue ID.
CIR	Displays the committed information rate.
PIR	Displays the peak information rate.
CBS	Displays the committed burst size.
MBS	Displays the maximum burst size.
HiPrio	Displays the high priority value.
FC	Displays FC to queue mapping.
UCastQ	Displays the specific unicast queue to be used for packets in the forwarding class.

```
A:ALA-12# show qos network-queue nq1
===============================================================================
```

```
QoS Network Queue Policy
===============================================================================
Network Queue Policy (nq1)
-------------------------------------------------------------------------------
Policy        : nq1
Description   : (Not Specified)
-------------------------------------------------------------------------------
Associations
-------------------------------------------------------------------------------
Port-id : 1/1/1
===============================================================================
A:ALA-12>show>qos#

A:ALA-12>show>qos# network-queue nq1 detail
===============================================================================
QoS Network Queue Policy
===============================================================================
Network Queue Policy (nq1)
-------------------------------------------------------------------------------
Policy        : nq1
Description   : (Not Specified)
-------------------------------------------------------------------------------
Queue CIR       PIR        CBS      MBS      HiPrio
-------------------------------------------------------------------------------
1     0         100        1        50       10
2     25        100        5        50       10
3     25        100        20       50       10
4     25        100        5        25       10
5     100       100        20       50       10
6     100       100        20       50       10
7     10        100        5        25       10
8     10        100        5        25       10
-------------------------------------------------------------------------------
FC              UCastQ
-------------------------------------------------------------------------------
be              1
l2              2
af              3
l1              4
h2              5
ef              6
h1              7
nc              8


-------------------------------------------------------------------------------
Associations
-------------------------------------------------------------------------------
Port-id : 1/1/1
===============================================================================
A:ALA-12>show>qos#
```