

Configure Queuing and Scheduling

- Queuing and Scheduling, on page 1
- Prerequisites for Queuing and Scheduling, on page 5
- Guidelines and Limitations for Queuing and Scheduling, on page 5
- Configure Queuing and Scheduling, on page 14
- Configure Congestion Management, on page 22
- Apply a Queuing Policy on a System, on page 27
- Verify the Queuing and Scheduling Configuration, on page 27
- Control the QoS Shared Buffer, on page 28
- Manage Dynamic Buffer Sharing, on page 28
- Monitor the QoS Packet Buffer, on page 29
- Configuration Examples for Queuing and Scheduling , on page 30

Queuing and Scheduling

The queuing and scheduling processes provides a robust framework for managing network traffic, ensuring that data flows smoothly and efficiently across the network. Traffic queuing, traffic scheduling, traffic shaping, congestion avoidance, and congestion management services are used to achieve this as mentioned in the following sections.

Traffic Queuing

Traffic queuing involves ordering packets for both input and output data. Devices can support multiple queues to control packet sequencing in different traffic classes. This is crucial for managing how data flows through a network, ensuring that packets are processed in an orderly manner.

Traffic Scheduling

Traffic scheduling is the methodical output of packets at a desired frequency to accomplish a consistent flow of traffic. You can apply traffic scheduling to different traffic classes to weight the traffic by priority.

Modifying Class Maps



Note The provided system-defined queuing class maps cannot be modified.

- **Default Behavior**: By default, all network traffic is grouped into a single category called **qos-group 0**. This means that without any specific configuration, all traffic is treated the same way.
- **System-Defined Classes**: These are predefined categories that manage how different types of traffic are handled. They cannot be changed directly.
- **Policy Handling**:
  - You can create a type queuing policy which allows you to configure particular que group. For more information see Configure Type Queuing Policies, on page 16.
  - When you assign traffic to a different qos-group using a **Type QoS policy**, you might need to adjust these system-defined policies further to meet specific needs, such as reallocating bandwidth.

For information about configuring policy maps and class maps, see the Using Modular QoS CLI chapter.

Traffic Shaping

Traffic shaping is a technique that is used to control the flow of traffic leaving an interface to ensure it matches the speed of the remote target interface and adheres to contracted policies. This process helps eliminate bottlenecks caused by data-rate mismatches by regulating and smoothing packet flow. Key aspects include:

- **Maximum Traffic Rate**: Imposes a limit on the traffic rate for each port's egress queue, buffering packets that exceed this threshold to minimize packet loss.
- **Comparison to Traffic Policing**: Traffic shaping buffers packets instead of dropping them, thereby improving TCP traffic behavior.
- **Bandwidth Control**: Allows control over available bandwidth, ensuring traffic conforms to the shaper rates and avoids excess egress traffic for the particular target interface.
- **Queue Length Thresholds**: Configured using Weighted Random Early Detection (WRED) to manage queue lengths effectively.

Congestion Avoidance

You can use the following methods to proactively avoid traffic congestion on the device:

- Apply WRED to TCP or non-TCP traffic.
- Apply tail drop to TCP or non-TCP traffic.

Congestion Management

Congestion Management uses the following methods to maintain network performance by preventing congestion when queues exceed their thresholds.

- Explicit Congestion Notification
- Approximate Fair Drop
- Weighted Random Early Detection

For information about configuring congestion management, see the Configuring WRED on Egress Queues section.

Explicit Congestion Notification

Explicit Congestion Notification (ECN) is an extension to WRED that marks packets instead of dropping them when the average queue length exceeds a specific threshold value. This helps in signaling congestion to routers and end hosts, prompting them to slow down packet transmission.

Approximate Fair Drop

Approximate Fair Drop (AFD) is an Active Queue Management (AQM) algorithm that acts on long lived large flows (elephant flows) in case of congestion, and does not impact short flows (mice flows).

When congestion occurs, the AFD algorithm maintains the queue occupancy at the configured queue desired value by probabilistically dropping packets from the large flows and not impacting short flows.

ECN can be enabled with AFD on a particular class of traffic to mark the congestion state instead of dropping the packets.



Note The AFD algorithm is applicable only on the flows that are qualified as elephant flows. Mice flows are protected and are not subject to AFD dropping.

AFD User Profiles

Three user profiles are provided with AFD:

- Mesh (Aggressive)

  AFD and ETRAP timers are set to be aggressive, so that the queue depth does not grow much and is kept close to the queue-desired value.

- Burst (Default)

  AFD and ETRAP timers are neither aggressive nor conservative, so that the queue depth could be observed to be hovering near the queue-desired value.

- Ultra-burst (Conservative)

  AFD and ETRAP timers are set to be conservative, so that more bursts are absorbed and fluctuations for queue depth can be observed around the queue-desired value.

These profiles set the ETrap and AFD timers to pre-configured values for different traffic profiles such as, very bursty or not-so bursty traffic. For more configuration flexibility, the ETrap period set by the profile can be overridden by configuring the ETrap age-period with the **hardware qos etrap** command. However, the AFD timer cannot be changed.

The following is an example of configuring the ETrap age-period:

```
switch(config)# hardware qos etrap age-period 50 usec
```

The following are examples of configuring the AFD user profiles:

- Mesh (Aggressive with ETrap age-period: 20 µsec and AFD period: 10 µsec)

  ```
  switch(config)# hardware qos afd profile mesh
  ```

- Burst (Default with ETrap age-period: 50 µsec and AFD period: 25 µsec)

  ```
  switch(config)# hardware qos afd profile burst
  ```

- Ultra-burst (Conservative with ETrap age-period: 100 µsec and AFD period: 50 µsec)

  ```
  switch(config)# hardware qos afd profile ultra-burst
  ```

Elephant Flow

When the number of bytes received in a flow exceeds the number of bytes specified by the ETrap byte-count-threshold, the flow is considered an elephant flow or large flow.

For a flow to continue to be an elephant flow, the configured **bw_threshold** number of bytes has to be received in the configured timer period. Otherwise, the flow is evicted from the ETrap hash table.

The ingress rate of every elephant flow is calculated and forwarded to egress for the AFD algorithm to consume.

Elephant Trap

The Elephant Trap (ETrap) identifies and hashes flows and forwards the arrival rate per flow to AFD for drop probability computation. It helps in distinguishing between large and short flows, ensuring that only large flows are subject to AFD dropping.

ETrap Parameters

ETrap has the following parameters that can be configured:

- **Byte-count**

  Byte-count Is used to identify elephant flows. When number of bytes received in a flow exceeds the number of bytes specified by the byte-count-threshold, the flow is considered an elephant flow. (Default byte-count is ~ 1 MB.)

- **Age-period and Bandwidth-threshold**

  Age-period and Bandwidth-threshold are used together to track the activeness of an elephant flow.

  When the average bandwidth during the age-period time is lower than the configured bandwidth-threshold, an elephant flow is considered inactive and is timed-out and removed from the elephant flow table. (Default age-period is 50 µsec. Default bandwidth-threshold is 500 bytes.

Example:

```
switch (config)# hardware qos etrap age-period 50 usec
switch (config)# hardware qos etrap bandwidth-threshold 500 bytes
switch (config)# hardware qos etrap byte-count 1048555
```

Weighted Random Early Detection

Weighted Random Early Detection is another AQM algorithm that computes a random drop probability and drops packets indiscriminately across all flows in a traffic class. It cannot be used simultaneously with AFD, as both serve similar purposes but with different methodologies.

Comparison of WRED and AFD

Feature	WRED	AFD
Algorithm Type	Active Queue Management	Active Queue Management
Drop Mechanism/Congestion Management	Computes a random drop probability and drops packets indiscriminately across all flows in a class of traffic	Computes drop probability based on the arrival rate of incoming flows, compares it with the computed fair rate, and drops packets from large flows without impacting short flows
Priority Handling	Considers packet priority (CoS, DSCP/traffic class, or IP precedence value) to maintain higher-priority flows	Focuses on fairness by distinguishing between long-lived elephant flows and short-lived mice flows, exempting mice flows from dropping

Note AFD and WRED cannot be applied at the same time. Only one can be used in a system.

Prerequisites for Queuing and Scheduling

Queuing and scheduling have the following prerequisites:

- You must be familiar with using modular QoS CLI.
- You must be logged on to the device.

Guidelines and Limitations for Queuing and Scheduling

Queuing and scheduling have the following configuration guidelines and limitations:

Note For scale information, see the release-specific Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Configuration and Port Limitations for Queuing and Scheduling

- **Ports Limitations**

- Changes are disruptive. The traffic passing through ports of the specified port type experiences a brief period of traffic loss. All ports of the specified type are affected.
- Performance can be impacted. If one or more ports of the specified type do not have a queuing policy applied that defines the behavior for the new queue, the traffic mapping to that queue can experience performance degradation.
- WRED is not supported on ALE enabled device front panel 40G uplink ports. When WRED is configured for the system level, the setting is ignored and no error message is displayed. When WRED is configured at the port level, the setting is rejected and an error message displays.

- **Configuration Limitations**
  - The **show** commands with the **internal** keyword are not supported.
  - The device supports a system-level queuing policy, so all ports in the system are impacted when you configure the queuing policy.
  - A **type queuing policy** can be attached to the system or to individual interfaces for input or output traffic.
  - When there is a link flap on a port with active traffic, it results in a packet/traffic loss flowing through other ports on the same or different slices. To avoid the flow discards, make sure you reduce the queue limit from the default value to a lower value and apply it at the system level.
  - When configuring priority for one class map queue (SPQ), configure the priority for QoS Group 3. When configuring priority for more than one class map queue, configure the priority on the higher numbered QoS groups. In addition, the QoS groups must be next to each other. For example, if you want to have two SPQs, you have to configure the priority on QoS Group 3 and on QoS Group 2.
  - If granted buffer is not carved out using a custom input queuing policy for a specified group, only global shared buffers are used.

Switch Limitations for Queuing and Scheduling

- The minimum egress shaper granularity is 200 Mbps per queue for Cisco Nexus 9300-GX2/H2R/H1 platform switches and line cards.
- Maximum queue occupancy for Leaf Spine Engine (LSE) enabled switches is limited to 64K cells (~13MB).
- For the following Cisco Nexus series switches and line cards, the lowest value that the egress shaper can manage, per queue, is 100 Mbps:
  - Cisco Nexus 9300-FX/FX2/GX platform switches
  - Cisco Nexus X97160YC-EX, 9700-FX line cards
- The **queue-limit** configuration is applicable only in ingress queuing policy on Cisco Nexus 9500 switches with 9600-R/RX line cards.
- The **bandwidth percent** configuration is applicable only in egress queuing policy on Cisco Nexus 9500 switches with 9600-R/RX line cards. Ensure that the ingress queue-limit is configured before configuring the bandwidth percent command.

- For Cisco Nexus 9300-FX Series and subsequent series switches, the minimum bandwidth allocated to a queue is 1%.
- Deep Buffer - Cisco Nexus 9332D-H2R platform switches support Deep buffers for Unicast traffic. Deep buffer allows the user to handle huge bursts of traffic in the switch by providing 8GB additional buffer in-addition to the existing buffers (40MB) in the switch. Deep buffer is enabled by default in the system across all the queues such that any queue has the flexibility to occupy those buffers during congestion scenarios. Multicast traffic is not supported on deep buffers.
- On Cisco Nexus 9332D-H2R platform switches, there are two special 33, 34 ports, mostly for management traffic. These ports have all the same features as any regular ports except for:
  - MACsec and PTP and Frequency Synchronization are not supported.
  - Deep buffer is not supported on these ports, due to limitation on the total number of queues.
  - These are lower bandwidth ports.
  - These ports operate in store-and-forward mode only.
  - These ports don’t support shaper Min/Max rate guarantees.
  - PFC and no-drop classes are not supported on these ports.
  - FC mode is not supported on these ports.

Feature Limitations for Queuing and Scheduling

- **Traffic Shaping**
  - Traffic shaping can increase the latency of packets due to queuing because it falls back to store-and-forward mode when packets are queued.
  - Traffic shaping is not supported on the Cisco Nexus 9300 ALE 40G ports. For more information on ALE 40G uplink ports, see the Limitations for ALE 40G Uplink Ports on the Cisco Nexus 9000 Series Switches.
  - When configuring traffic shaping on egress queues, the 'even range' command displays a range of 1-400000000000. However, this range must be modified to match the maximum port capacity supported by the switch.
  - Configuring traffic shaping for a queue is independent of priority or bandwidth in the same policy map.
  - The system queuing policy is applied to both internal and front panel ports. When traffic shaping is enabled on the system queuing policy, traffic shaping is also applied to the internal ports. As a best practice, do not enable traffic shaping on the system queuing policy.
  - The lowest value that the egress shaper can manage, per queue, is 100 Mbps on Cisco Nexus 9300-FX/FX2/GX switches and X97160YC-EX, 9700-FX line cards.
- **FEX**
  - FEX supports
    - System inputs (ingress) level queuing for HIF to NIF traffic.
    - The system outputs (egress) level queuing for NIF to HIF traffic and HIF to HIF traffic.

- The egress queuing feature works only for base ports and not for FEX ports.
- When the switch supported system queuing policy is configured, the FEX uses the default policy.
- The FEX QoS system level queuing policy does not support the following features.
  - WRED
  - Queue-limit
  - Traffic Shaping
  - Policing features
  - Multiple priority levels.

- **AFD**
  - Approximate Fair Drop is not supported on the Cisco Nexus 9508 switch (NX-OS 7.0(3)F3(3)).
  - AFD and WRED cannot be applied at the same time. Only one can be used in a system.
  - If an AFD policy has already been applied in system QoS and you are configuring two unique AFD queuing policies, you must apply each unique AFD policy on ports on the same slice.

    The following is an example of the system error if you do not create and apply a unique AFD policy on the same slice:

```
Eth1/50    1a006200 1    0    40    255   196   -1    1     0     0    <<<slice 1
    Eth1/51    1a006400 1    0    32    255   200   -1    0     32    56   <<<slice
 0
    Eth1/52    1a006600 1    0    64    255   204   -1    1     24    48   <<<slice
 1
    Eth1/53    1a006800 1    0    20    255   208   -1    0     20    40   <<<slice
 0

switch(config)# interface ethernet 1/50
    switch(config-if)# service-policy type queuing output LM-out-40G
    switch(config)# interface ethernet 1/51
    switch(config-if)#service-policy type queuing output LM-out-100G
    switch(config)# interface ethernet 1/52
    switch(config-if)# service-policy type queuing output LM-out-100G
    Unable to perform the action due to incompatibility:  Module 1 returned status
 "Max profiles reached for unique values of queue management parameters (alpha,
beta, max-threshold) in AFD config"
```

  - If no AFD policy has already been applied in system QoS, then you can configure the same AFD policy on ports on a different slice, or configure different AFD policies on ports in the same slice.



Note You cannot configure an AFD queuing in the System QoS later.

The following is an example of the system error when AFD queuing is already configured in the system:

```
interface Ethernet1/50
      service-policy type queuing output LM-out-40G
    interface Ethernet1/51
      service-policy type queuing output LM-out-40G
    interface Ethernet1/52
```

```
      service-policy type queuing output LM-out-100G
    interface Ethernet1/53
      service-policy type queuing output LM-out-100G
    interface Ethernet1/54
      service-policy type queuing output LM-out-100G

    (config-sys-qos)# service-policy type queuing output LM-out
    Unable to perform the action due to incompatibility:  Module 1 returned status
 "Max profiles reached for unique values of queue management parameters (alpha,
beta, max-threshold) in AFD config"
```

Order of Resolution

The following describes the order of resolution for the pause buffer configuration and the queue-limit for a priority-group.

- Pause Buffer Configuration

  The pause buffer configuration is resolved in the following order:

  - Interfaces ingress queuing policy (if applied, and pause buffer configuration is specified for that class).
  - Systems ingress queuing policy (if applied, and pause buffer configuration is specified for that class).
  - System network-QoS policy (if applied, and pause buffer configuration is specified for that class).
  - Default values with regard to the speed of the port.

- Queue-limit for Priority-Group

  The queue-limit for a priority-group is resolved in the following order:

  - Interfaces ingress queuing policy (if applied, and queue-limit configuration is specified for that class).
  - Systems ingress queuing policy (if applied, and queue-limit configuration is specified for that class).
  - The **hardware qos ing-pg-share** configuration provided value.
  - System default value.

Ingress Queuing

The following are notes about ingress queuing:

- The default systems ingress queuing policy does not exist.
- The ingress queuing policy is used to override the specified pause buffer configuration.
- When downgrading to an earlier release of Cisco Nexus 9000 NX-OS, all ingress queuing configurations have to be removed.
- The ingress queuing feature is supported only on platforms where priority flow control is supported.
- Ingress queuing is not supported on devices with 100G ports.

- The Cisco Nexus 9636C-R and 9636Q-R line cards and the Cisco Nexus 9508-FM-R fabric module (in a Cisco Nexus 9508 switch) support ingress queuing.
- The Cisco Nexus 9500 switches with 9600-R/RX line cards support only burst-mode to use the big-buffer provided by hardware.



Note The recommendation is to use the same port speeds at ingress and egress side.

Queuing Policy and Egress Queue Mapping

On Cisco Nexus 9500 switches with 9600-R, R2, RX line cards, the queuing policy and egress queue mapping differ from CloudScale switches. The queuing policies are mapped in reverse order.

R Series Example:

- Queuing Policy 7 → Egress Queue 0,
- Queuing Policy 6 → Egress Queue 1, and so on

Supported Platform and Release for Queuing and Scheduling

Supported Release	Supported Platform	Limitation
9.3(3) and later	Cisco Nexus 9300-FX/FX2/GX Series switches	
9.3(5) and later	Cisco Nexus 9300-FX3 Series switches	
10.1(2) and later	N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.	For R2, though different priority levels can be set through CLI, only priority level 1 is supported in queuing policy.
10.2(3)F and later	Cisco Nexus 9300-GX2 Series switches	
10.4(1)F and later	Cisco Nexus 9332D-H2R switches	
	Cisco Nexus C9348GCFX3 and C9348GC-FX3PH	On C9348GC-FX3PH switch: • Queuing and Scheduling policies are supported on switch except for the ports 41–48. • Configuring WRED on egress queues is not supported.

Supported Release	Supported Platform	Limitation
10.4(2)F and later	Cisco Nexus C93108TC-FX3 switches Cisco Nexus 93400LD-H1 switches	
	Cisco Nexus C9232E-B1 switches	• Eight queues - SPAN and CPU Queues with eight user queues are supported. • SP, DWRR, and Shaper are supported. • Queuing statistics is supported.
10.4(3)F and later	Cisco Nexus 9364C-H1 switches	

Note

- AFD, WRED is not supported on the Cisco Nexus 9508 switch (NX-OS 7.0(3)F3(3)).
- PVLANs do not provide support for PVLAN QoS.

Guidelines and Limitations for Queuing and Scheduling on Cisco Nexus 9800 Series switches

Table 1: Supported Platform and Release

Supported Release	Supported Platform
10.3(1)F and later	Cisco Nexus 9808 Series switches
10.4(1)F and later	Cisco Nexus 9804 Series switches

Supported or unsupported features on Cisco Nexus 9800 Series switches.

- Queuing statistics is supported.
- Ingress queuing is supported.
- The queue depth counter per queue is not supported but additional queuing counters on VOQ tail drops are supported.
- AFD is not supported on the Cisco Nexus 9808 switch.
- Supports only the eight queue configuration in Queuing and Scheduling policies. Fewer queues can be configured but are not supported.
- Eight queues - SPAN and CPU Queues that are overloaded with eight user queues are supported.
- SP, DWRR, WRED, and ECN are supported. However, the shaper and DWRR accuracy will have a 5% variance.
- Maximum shaper and static limit are supported.

- Micro-Burst Monitoring is not supported.
- Link Level Flow Control is not supported.
- Dynamic queue-limit is not supported.
- Multicast Queuing statistics is not supported.
- Beginning with Cisco NX-OS Release 10.5(3)F, Fast ECN marking is supported to route all traffic through the fabric, which allows ECN marking at dequeue on Cisco Nexus 9800 Series switches with N9K-X9836DM-A and N9K-X98900CD-A line cards.
- During congestion on the egress interface, the VoQ architecture does not guarantee fair packet drops among ingress interfaces within the same slice. Fairness in packet drops can be expected only when ingress interfaces belong to different slices.

Guidelines and Limitations for Queuing and Scheduling on Cisco Nexus 9364E-SG2-Q and 9364E-SG2-O switches

Table 2: Supported Platform and Release

Supported Release	Supported Platform
10.5(3)F and later	Cisco Nexus 9364E-SG2-Q and 9364E-SG2-O switches

Supported or unsupported features on Cisco Nexus 9364E-SG2-Q and 9364E-SG2-O switches:

- Eight queues - SPAN and CPU Queues that are overloaded with eight user queues are supported.
- SP, DWRR, WRED, and ECN are supported. However, the shaper and DWRR accuracy will have a 5% variance.
- Maximum shaper and static limit are supported.
- Micro-Burst Monitoring is not supported.
- Link Level Flow Control is not supported.
- Dynamic queue-limit is not supported.
- Multicast Queuing statistics is not supported.
- AFD is not supported.

Configuration limitation:

- After executing the **clear mac address-table dynamic** command, during the transition of traffic from an Layer 3 SVI to an Layer 2 interface, a small number of packets are temporarily queued in Queue 0 on the Layer 2 SVI ports.

Guidelines and Limitations for Queuing and Scheduling on Cisco Nexus 9300-SE1 switches

Table 3: Supported Platform and Release

Supported Release	Supported Platform
10.6(1)F and later	Cisco Nexus 9336C-SE1 switches
10.6(2)F and later	Cisco Nexus 9396Y12C-SE1 and 9396T12C-SE1 switches
10.6(3)F and later	Cisco Nexus 9348Y12C-SE1 switches

Supported or unsupported features:

- Eight queues - SPAN and CPU Queues that are overloaded with eight user queues are supported.
- SP and DWRR are supported. However, the shaper and DWRR accuracy will have a 5% variance.
- QoS statistics are supported.
- Maximum shaper and static limit are supported.
- Micro-Burst Monitoring is not supported.
- Link Level Flow Control is not supported.
- Dynamic queue-limit is not supported.
- Dynamic buffer sharing is not supported.
- Multicast Queuing statistics is not supported.

Feature limitation:

- During congestion on the egress interface, the VoQ architecture does not guarantee fair packet drops among ingress interfaces within the same slice. Fairness in packet drops can be expected only when ingress interfaces belong to different slices.

Guidelines and Limitations for Queuing and Scheduling on Cisco N9300 Smart switches

Table 4: Supported Platform and Release

Supported Release	Supported Platform
10.6(2)F and later	Cisco N9324C-SE1U and N9348Y2C6D-SE1U Smart switches.

Supported features:

- Eight queues - SPAN and CPU Queues that are overloaded with eight user queues are supported.
- SP and DWRR are supported. However, the shaper and DWRR accuracy will have a 5% variance.
- QoS statistics are supported.
- Maximum shaper and static limit are supported.

Unsupported features:

- Micro-Burst monitoring is not supported.
- Link Level Flow Control (LLFC) is not supported.
- Dynamic queue-limit is not supported.
- Multicast queuing statistics is not supported.
- Dynamic buffer sharing is not supported.

Feature limitation:

- During congestion on the egress interface, the VoQ architecture does not guarantee fair packet drops among ingress interfaces within the same slice. Fairness in packet drops can be expected only when ingress interfaces belong to different slices.

Guidelines and Limitations for Queuing and Scheduling on Cisco Nexus 9164E-NS4-O switches

Table 5: Supported Platform and Release

Supported Release	Supported Platform
10.6(2n) and later	Cisco Nexus 9164E-NS4-O switches

Supported or unsupported features on Cisco Nexus 9164E-NS4-O switches:

- Eight data queues and one CPU queue are supported.
- Dynamic queue-limits are supported.
- Strict priority, DWRR, WRED, and ECN are supported.
- Shaper feature is supported.
- LLFC is not supported.
- Static queue-limits for ingress and egress queuing policy are not supported.
- Multicast traffic is not supported.
- L2 traffic is not supported.

Configuration limitations:

- Weight of WRED/ECN is a global parameter and not per profile.
- The maximum weight index supported by WRED/ECN is 11.

Configure Queuing and Scheduling

Queuing and scheduling are configured by creating policy maps of type queuing that you apply to an egress interface. You cannot modify system-defined class maps, which are used in policy maps to define the classes of traffic to which you want to apply policies.

The system-defined policy map, default-out-policy, is attached to all ports to which you do not apply a queuing policy map. The default policy maps cannot be configured.

You can perform the following Queuing and Scheduling configurations:

- **Type Queuing Policies**
  - Type queuing policies for egress are used for scheduling and buffering the traffic of a specific system class. A type queuing policy is identified by its QoS group and can be attached to the system or to individual interfaces for input or output traffic.



Note The ingress queuing policy is used to configure pause buffer thresholds. For more details, see the Priority Flow Control section.

- **Congestion Avoidance**
  - **Tail drop configuration**: You can configure tail drop on egress queues by setting thresholds. The device drops any packets that exceed the thresholds. You can specify a threshold based on the queue size or buffer memory that is used by the queue.
  - **WRED configuration**: You can configure WRED on egress queues to set minimum and maximum packet drop thresholds. The frequency of dropped packets increases as the queue size exceeds the minimum threshold. When the maximum threshold is exceeded, all packets for the queue are dropped.
  - **AFD configuration**: AFD can be configured for an egress queuing policy.

- **Congestion Management**
  - **Bandwidth and bandwidth remaining configuration**: You can configure the bandwidth and bandwidth remaining on the ingress and egress queue to allocate a minimum percentage of the interface bandwidth to a queue.
  - **Priority configuration**: If you do not specify the priority, the system-defined egress priority queue(pq) queues behave as normal queues.
    - You can configure only one level of priority on an egress priority queue. Use the system-defined priority queue class for the type of module to which you want to apply the policy map.
    - For the nonpriority queues, you can configure how much of the remaining bandwidth to assign to each queue. By default, the device evenly distributes the remaining bandwidth among the nonpriority queues.



Note
- When a priority queue is configured, the other queues can only use the remaining bandwidth in the same policy map.
- When configuring priority for one class map queue (SPQ), you must configure the priority for QoS Group 3. When configuring priority for more than one class map queue, you must configure the priority on the higher numbered QoS groups. In addition, the QoS groups must be next to each other. For example, if you want to have two SPQs, you have to configure the priority on QoS Group 3 and on QoS Group 2.

- **Traffic shaping configuration**: You can configure traffic shaping on an egress queue to impose a minimum and maximum rate on it.

Configure Type Queuing Policies

Follow these steps to configure type queuing policies.

Procedure

Step 1 Run the **policy-map type queuing** *policy-name* command in global configuration mode, to create a named object that represents a set of policies that are to be applied to a set of traffic classes.

Example:

```
switch# configure terminal
switch(config)# policy-map type queuing shape_queues
switch(config-pmap-que)#
```

Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 2 Run the **class type queuing** *class-name* command to associate a class map with the policy map, and enter the specified system class configuration mode.

Example:

```
switch(config-pmap-que)# class type queuing c-out-q-default
switch(config-pmap-c-que)#
```

Step 3 Run the **priority** command to specify that traffic in this class is mapped to a strict-priority queue.

Example:

```
switch(config-pmap-c-que)# priority
```

Run the **no priority** command to remove the strict priority queuing from the traffic in this class.

Step 4 Run the **shape min** *Target-bit-rate* [**kbps** | **mbps** | **gbps** | **mbps** | **pps**] **max** *Target-bit-rate* [**kbps** | **mbps** | **gbps** | **mbps** | **pps**] command to specify the maximum and minimum shape size for the queue.

Example:

```
switch(config-pmap-c-que)# shape min 100 mbps max 150 mbps
```

Step 5 Run the **bandwidth percent** *percentage* command to assign a minimum rate of the interface bandwidth to an output queue as the percentage of the underlying interface link rate.

Example:

```
switch(config-pmap-c-que)# bandwidth percent 25
```

The class receives the assigned percentage of interface bandwidth if there are no strict-priority queues. If there are strict-priority queues, however, the strict-priority queues receive their share of the bandwidth first. The remaining bandwidth is shared in a weighted manner among the class configured with a bandwidth percent. For example, if strict-priority queues take 90 percent of the bandwidth, and you configure 75 percent for a class, the class receives 75 percent of the remaining 10 percent of the bandwidth.

Note

Before you can successfully allocate bandwidth to the class, you must first reduce the default bandwidth configuration on class-default and class-fcoe.

Run the **no bandwidth percent** *percentage* command to remove the bandwidth specification from this class.

Step 6 (Optional) Run the **priority level** *level* command to specify the strict-priority levels for the switches.

Example:

```
switch(config-pmap-c-que)# priority level 3
```

Range: 1 to 7.

Step 7 (Optional) Run the **queue-limit** *queue size* [**dynamic** *dynamic threshold*] command to specify the static or dynamic shared limit available to the queue for switches.

Example:

```
switch(config-pmap-c-que)# queue-limit 1000 mbytes
```

- The static queue limit defines the fixed size to which the queue can grow.

  Note
  The minimum *queue size* must be at least 50 kilobytes.

- The **dynamic** queue limit allows the queue's threshold size to be decided depending on the number of free cells available, in terms of the alpha value.

  Note
  - Starting from Release 10.4(1)F, the enhanced queue limit range is from 0 to 9437184. The maximum threshold supported in Cisco Nexus 9332D-H2R platform switches is 256MB.

Configure Congestion Avoidance

You can configure congestion avoidance with tail drop or WRED features. Both features can be used in egress policy maps.

Note WRED and tail drop cannot be configured in the same class.

Configure Tail Drop on Egress Queues

Follow these steps to configure tail drop on egress queues

Procedure

Step 1 Run the **hardware qos q-noise percent** *value* command in global configuration mode, to tune the random noise parameter.

Example:

```
switch# configure terminal
switch(config)# hardware qos q-noise percent 30
```

Default: 20 percent.

Step 2 Run the **policy-map** [**type queuing**] [**match-first**] [*policy-map-name*] command to configures the policy map of type queuing and then enters policy-map mode for the policy-map name you specify.

Example:

```
switch(config)# policy-map type queuing shape_queues
switch(config-pmap-que)#
```

Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 3 Run the **class type queuing** *class-name* command to configure the class map of type queuing and then enter policy-map class queuing mode.

Example:

```
switch(config-pmap-que)# class type queuing c-out-q1
switch(config-pmap-c-que)#
```

Class queuing names are listed in the previous System-Defined Type queuing Class Maps table.

Step 4 Run the **queue-limit** {*queue-size* [**bytes** | **kbytes** | **mbytes**] | **dynamic** *value*} command to assign a tail drop threshold based on the queue size.

Example:

```
switch(config-pmap-c-que)# queue-limit 1000 mbytes
```

Queue size is in bytes, kilobytes, megabytes or allows the queue's threshold size to be determined dynamically depending on the number of free cells available. The device drops packets that exceed the specified threshold.

The valid values for byte-based queue size: 1 to 83886080.

The valid values for dynamic queue size: 0 to 10 as specified in the following table:

Value of alpha	Network Forwarding Engine (NFE) enabled switches		Leaf Spine Engine (LSE) enabled switches		
	Definition	Max % per queue	Definition	Max % per queue	ASIC value
0	1/128	~0.8%	1/8	~11%	0
1	1/64	~1.5%	1/4	~20%	1
2	1/32	~3%	1/2	~33%	3
3	1/16	~6%	3/4	~42%	5
4	1/8	~11%	1 1/8	~53%	8
5	1/4	20%	1 3/4	~64%	14
6	1/2	~33%	3	~75%	16
7	1	50%	5	~83%	18
8	2	~66%	8	~89%	21

Value of alpha	Network Forwarding Engine (NFE) enabled switches		Leaf Spine Engine (LSE) enabled switches		
	Definition	Max % per queue	Definition	Max % per queue	ASIC value
9	4	~80%	14	~92.5	27
10	8	~89%	18	~95%	31

For example, if you configure a dynamic queue size of 6, then the alpha value is ½. If you configure a dynamic queue size of 7, then the alpha value is 1.

To calculate the queue-limit consider the following:

queue-limit = (alpha/(1 + alpha)) x total buffers

For example, if you configure a queue-limit with a dynamic queue size of 7, then the queue-limit can grow up to (1/(1+1)) x total buffers. This means that queue-limit = ½ x total buffers.

Note
Although the above calculations determine the maximum queue occupancy, the maximum queue occupancy is limited to 64K cells in all cases for Application Spine Engine (ASE2, ASE3) and Leaf Spine Engine (LSE) enabled switches.

Note
Setting the threshold on ALE enabled devices is only supported for the system level. It is not supported for the port level.

Step 5 Repeat Steps 3 and 4 to assign tail drop thresholds for other queue classes.

Step 6 Run the **show policy-map** [**type queuing** [*policy-map-name* | **default-out-policy**]] command to displays information about all configured policy maps, all policy maps of type queuing, a selected policy map of type queuing, or the default output queuing policy.

Example:

```
switch(config)# show policy-map type queuing shape_queues
```

Step 7 Run the **copy running-config startup-config** command to save the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config
startup-config
```

Configure WRED on Egress Queues

Follow these steps to configure WRED on egress queues.

Procedure

Step 1 Run the **policy-map** [**type queuing**] [**match-first**] [*policy-map-name*] command to configure the policy map of type queuing and then enters policy-map mode for the policy-map name you specify.

Example:

```
switch(config)# policy-map type queuing shape_queues
switch(config-pmap-que)#
```

Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 2 Run the **class type queuing** *class-name* command to configure the class map of type queuing and then enter policy-map class queuing mode.

Example:

```
switch(config-pmap-que)# class type queuing c-out-q1
switch(config-pmap-c-que)#
```

Class queuing names are listed in the previous System-Defined Type queuing Class Maps table.

Step 3 Run the **random-detect** [**minimum-threshold** *min-threshold* {**packets** | **bytes** | **kbytes** | **mbytes**} **maximum-threshold** *max-threshold* {**packets** | **bytes** | **kbytes** | **mbytes**} **drop-probability** *value* **weight** *value*] [**threshold** {**burst-optimized** | **mesh-optimized**}] [**ecn | non-ecn**] [**queue length weight** *value*] command to configure WRED on the specified queuing class.

Example:

WRED configuration

```
switch(config-pmap-c-que)# random-detect
minimum-threshold 10 mbytes
maximum-threshold 20 mbytes
```

Example:

WRED configuration with non ECN option

```
switch(config-pmap-c-que)# random-detect non-ecn
minimum-threshold 1000 kbytes
maximum-threshold 4000 kbytes
drop-probability 100
```

```
switch(config-pmap-c-que)# show queuing interface eth 1/1 | grep WRED
WRED Drop Pkts   0
WRED Non ECN Drop Pkts 0
switch(config-pmap-c-que)#
```

You can specify minimum and maximum thresholds used to drop packets from the queue. The thresholds are specified by the number of packets, bytes, kilobytes, or megabytes. The minimum and maximum thresholds must be of the same type. Range: 1 to 52428800.

Alternatively, you can specify a threshold that is optimized for burst or mesh traffic, or you can configure WRED to drop packets based on explicit congestion notification (ECN). Beginning with Cisco NX-OS Release 7.0(3)I6(1), the Network Forwarding Engine (NFE) platform supports the non-ecn option to configure drop thresholds for non-ECN flows.

Note

- The minimum-threshold and maximum-threshold parameters are not supported on the Cisco Nexus 9300 platform switches.
- Starting from Release 10.4(1)F, the WRED and ECN queue limit range is 0 to 52428800. The maximum threshold supported in Cisco Nexus 9332D-H2R platform switches is 256MB.

When random-detect is configured under policy-map, the default thresholds and drop probabilities are as following:

a. On newer platforms, the threshold is 0 and then the drop probabilities would be enforced irrespective of buffer utilization.

b. On older platforms, the threshold is min 100KB, max 120KB.

The drop probabilities are consistently 10% and 90% for burst-optimized and mesh-optimized respectively on all platforms

You can also specify the queue length weight for the traffic. The range of the queue length is 0-15.

Step 4 (Optional) Repeat Steps 2 and 3 to configure WRED for other queuing classes.

Step 5 (Optional) Configure the **congestion-control random-detect forward-nonecn** command to allow non-ECN-capable traffic to bypass WRED thresholds and grow up to the maximum size allowed for the queue (additional packets are tail dropped).

Example:

```
switch(config-pmap-c-que)# congestion-control random-detect forward-nonecn
```

This is a global command intended to be used with a WRED and ECN configuration and when the intention is to avoid WRED drops of non-ECN-capable traffic.

This option is supported on CloudScale switches and line cards.

Configure AFD on Egress Queues

Follow these steps to configure AFD on egress queues.

Procedure

Step 1 Run the **policy-map** [**type queuing**] [**match-first**] [*policy-map-name*] command in global configuration mode, to configure the policy map of type queuing and then enters policy-map mode for the policy-map name you specify.

Example:

```
switch# configure terminal
switch(config)# policy-map type queuing afd_8q-out
switch(config-pmap-que)#
```

Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 2 Run the **class type queuing** *class-name* command to configure the class map of type queuing and then enter policy-map class queuing mode.

Example:

```
switch(config-pmap-que)# class type queuing c-out-8q-q3
switch(config-pmap-c-que)#
```

Class queuing names are listed in the previous System-Defined Type queuing Class Maps table.

Step 3 Run the **afd queue-desired** <*number*> [**bytes** | **kbytes** | **mbytes**] [**ecn**] command to specify queue-desired.

Example:

Configuring AFD without ECN

```
switch(config)# policy-map type queuing afd_8q-out
switch(config-pmap-que)# class type queuing c-out-8q-q3
switch(config-pmap-c-que)# afd queue-desired 600 kbytes
```

Configuring AFD with ECN

```
switch(config)# policy-map type queuing afd-ecn_8q-out
switch(config-pmap-que)# class type queuing c-out-8q-q3
switch(config-pmap-c-que)# afd queue-desired 150 kbytes ecn
```

The following are recommended values for **queue-desired** for different port speeds:

Port Speed	Value for Queue
10G	150 kbytes
40G	600 kbytes
100G	1500 kbytes

Note

- Values for queue are user configurable.
- Starting from Release 10.4(1)F, AFD queue limit range is 0 to 52428800. The maximum threshold supported in Nexus 9332D-H2R platform switches is 256MB.

What to do next

After AFD is configured, you can apply the policy to the system or to an interface as follows:

- System

```
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing output afd_8q-out
```

- Interface

```
switch(config)# int e1/1
switch(config-if)# service-policy type queuing output afd_8q-out
```

Configure Congestion Management

You can configure only one of the following congestion management methods in a policy map:

- Allocate a minimum data rate to a queue by using the **bandwidth** and **bandwidth remaining** commands.
- Allocate all data for a class of traffic to a priority queue by using the **priority** command. You can use the **bandwidth remaining** command to distribute the remaining traffic among the nonpriority queues. By default, the system evenly distributes the remaining bandwidth among the nonpriority queues.
- Allocate a minimum and maximum data rate to a queue by using the **shape** command.

In addition to the congestion management feature that you choose, you can configure one of the following queue features in each class of a policy map:

- Tail drop thresholds based on the queue size and the queue limit usage. For more information, see Configure Tail Drop on Egress Queues, on page 17.
- WRED for preferential packet drops. For more information, see the Configuring WRED on Egress Queues section.

Note WRED is not supported on the Cisco Nexus 9508 switch (NX-OS 7.0(3)F3(3).

Configure Bandwidth and Bandwidth Remaining

You can configure the bandwidth and bandwidth remaining on the egress queue to allocate a minimum percentage of the interface bandwidth to a queue.

Note When a guaranteed bandwidth is configured, the priority queue must be disabled in the same policy map.

Follow these steps to configure bandwidth on egress queues.

Note If you are configuring bandwidth and bandwidth remaining on the egress queue for FEX, ensure that **feature-set fex** is enabled.

Procedure

Step 1 Run the **policy-map** [**type queuing**] [**match-first**] [*policy-map-name*] command in global configuration mode, to configure the policy map of type queuing and then enters policy-map mode for the policy-map name you specify.

Example:

```
switch# configure terminal
switch(config)# policy-map type queuing shape_queues
switch(config-pmap-que)#
```

Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 2 Run the **class type queuing** *class-name* command to configure the class map of type queuing and then enter policy-map class queuing mode.

Example:

```
switch(config-pmap-que)# class type queuing c-out-q1
switch(config-pmap-c-que)#
```

Class queuing names are listed in the previous System-Defined Type queuing Class Maps table.

Step 3 Run the **bandwidth** {**percent** *percent*}command to assign a minimum rate of the interface bandwidth to an output queue as the percentage of the underlying interface link rate.

Example:

```
switch(config-pmap-c-que)# bandwidth percent 25
```

Assigns a minimum rate of the interface bandwidth to an output queue as the percentage of the underlying interface link rate. The range: 0 to 100.

The example shows how to set the bandwidth to a minimum of 25 percent of the underlying link rate.

Step 4 Run the **bandwidth remaining percent** *percent* command to assign the percentage of the bandwidth that remains.

Example:

```
switch(config-pmap-c-que)# bandwidth remaining percent 25
```

Assigns the percentage of the bandwidth that remains to this queue. The range: 0 to 100.

The example shows how to set the bandwidth for this queue to 25 percent of the remaining bandwidth.

Step 5 (Optional) Repeat Steps 3 and 4 or 5 to assign bandwidth for other queue classes.

Step 6 Run the exit command to exit the policy-map queue mode and enter global configuration mode.

Example:

```
switch(config-cmap-que)# exit
switch(config)#
```

Step 7 (Optional) Run the **show policy-map** [**type queuing** [*policy-map-name* | **default-out-policy**]] command to display information about all configured policy maps, all policy maps of type queuing, a selected policy map of type queuing, or the default output queuing policy.

Example:

```
switch(config)# show policy-map type queuing shape_queues
```

Step 8 Run the **copy running-config startup-config** command to save the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Configure Priority

Follow these steps to specify the priority on egress queues.

Note If you are configuring priority on the egress queue for FEX, ensure that **feature-set fex** is enabled.

Procedure

Step 1 Run the **policy-map** [**type queuing**] [**match-first**] [*policy-map-name*] command in global configuration mode, to configure the policy map of type queuing and then enters policy-map mode for the policy-map name you specify.

Example:

```
switch# configure terminal
switch(config)# policy-map type queuing inq_pri
switch(config-pmap-que)#
```

Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 2 Run the **class type queuing** *class-name* command to configure the class map of type queuing and then enter policy-map class queuing mode.

Example:

```
switch(config-pmap-que)# class type queuing c-in-q3
switch(config-pmap-c-que)#
```

Class queuing names are listed in the previous System-Defined Type queuing Class Maps table.

Step 3 Run the **priority** [**level** *value*] command to select this queue as a priority queue. Only one priority level is supported.

Example:

```
switch(config-pmap-c-que)# priority
```

Note
FEX QoS priority is supported only on the **c-out-q3 class map**.

Step 4 (Optional) Run the **class type queuing***class-name* command to configures the class map of type queuing and then enters policy-map class queuing mode.

Example:

```
switch(config-pmap-c-que)# class type queuing c-in-q2
switch(config-pmap-c-que)#
```

Class queuing names are listed in the previous System-Defined Type queuing Class Maps table.

Choose a nonpriority queue where you want to configure the remaining bandwidth. By default, the system evenly distributes the remaining bandwidth among the nonpriority queues.

Step 5 Run the **bandwidth remaining percent** *percent* command to assign the percentage of the bandwidth that remains.

Example:

```
switch(config-pmap-c-que)# bandwidth remaining percent 25
```

Assigns the percentage of the bandwidth that remains to this queue. The range: 0 to 100.

The example shows how to set the bandwidth for this queue to 25 percent of the remaining bandwidth.

Step 6 (Optional) Repeat Steps 4 to 5 to assign the priority for the other nonpriority queues.

Step 7 (Optional) Run the **show policy-map** [**type queuing** [*policy-map-name* | **default-out-policy**]] command to display information about all configured policy maps, all policy maps of type queuing, a selected policy map of type queuing, or the default output queuing policy.

Example:

```
switch(config)# show policy-map type queuing shape_queues
```

Step 8 Run the **copy running-config startup-config** command to save the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Configure Traffic Shaping

Follow these steps to configure traffic shaping.

Before you begin

Configure random detection minimum and maximum thresholds for packets.

Procedure

Step 1 Run the **policy-map** [**type queuing**] [**match-first**] [*policy-map-name*] command in global configuration mode, to configure the policy map of type queuing and then enters policy-map mode for the policy-map name you specify.

Example:

```
switch# configure terminal
switch(config)# policy-map type queuing shape_queues
switch(config-pmap-que)#
```

Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.

Step 2 Run the **class type queuing** *class-name* command to configure the class map of type queuing and then enter policy-map class queuing mode.

Example:

```
switch(config-pmap-que)# class type queuing c-out-q1
switch(config-pmap-c-que)#
```

Class queuing names are listed in the previous System-Defined Type queuing Class Maps table.

Step 3 Run the **shape min** *value* {**bps** | **gbps** | **kbps** | **mbps** | **pps**} **max** *value* {**bps** | **gbps** | **kbps** | **mbps** | **pps**}command to assign a minimum and maximum bit rate on an output queue.

Example:

```
switch(config-pmap-c-que)# shape min 100 mbps max 150 mbps
```

The default bit rate is in bits per second (bps).

The example shows how to shape traffic to a minimum rate of 100 megabits per second (mbps) and a maximum rate of 150 mbps.

Note
Most scenarios where traffic shaping is needed requires the configuration of only the max shaper value. For instance, if you want traffic shaped and limited to a maximum desired rate, configure the min shaper value as 0 and the max shaper value as the maximum desired rate.

You should only configure the min shaper value for specific scenarios where a guaranteed rate is desired. For instance, if you want traffic to have a guaranteed rate, configure the min shaper value as the guaranteed rate and the max value as something greater than guaranteed rate (or the maximum of the port speed rate).

Step 4 (Optional) Repeat Steps 2 and 3 to assign shape traffic for other queue classes.

Step 5 (Optional) Run the **show policy-map** [**type queuing** [*policy-map-name* | **default-out-policy**]] command to display information about all configured policy maps, all policy maps of type queuing, a selected policy map of type queuing, or the default output queuing policy.

Example:

```
switch(config)# show policy-map type queuing shape_queues
```

Step 6 Run the **copy running-config startup-config** command to save the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Apply a Queuing Policy on a System

Follow these steps to apply a queuing policy globally on a system.

Procedure

Step 1 Run the **system qos** command in global configuration mode, to enter system QoS mode.

Example:

```
switch# configure terminal
switch (config)# system qos
switch (config-sys-qos)#
```

Step 2 Run the **service-policy type queuing output** {*policy-map-name* | **default-out-policy**} command to add the policy map to the input or output packets of the system.

Example:

```
switch (config-sys-qos)# service-policy type queuing map1
```

Note

- The **output** keyword specifies that this policy map should be applied to traffic sent from an interface.
- To restore the system to the default queuing service policy, use the **no** form of this command.

Verify the Queuing and Scheduling Configuration

Use the following commands to verify the queuing and scheduling configuration:

Command	Purpose
show class-map [**type queuing** [*class-name*]]	Displays information about all configured class maps, all class maps of type queuing, or a selected class map of type queuing.

Command	Purpose
show policy-map [**type queuing** [*policy-map-name* | **default-out-policy**]]	Displays information about all configured policy maps, all policy maps of type queuing, a selected policy map of type queuing, or the default output queuing policy.
show policy-map system	Displays information about all configured policy maps on the system.

Control the QoS Shared Buffer

The QoS buffer provides support per port/queue and shared space. You can control the QoS buffer that is shared by all flows by disabling or restricting reservations.

The **hardware qos min-buffer** command is used to control the QoS shared buffer.

hardware qos min-buffer [**all** | **default** | **none**]	• **all** Current behavior where all reservations are enabled ON). • **default** Enables reservations only for qos-group-0. • **none** Disables reservations for all qos-groups.

The **show hardware qos min-buffer** command is used to display the current buffer configuration.

Manage Dynamic Buffer Sharing

Beginning with NX-OS 7.0(3)I7(4), dynamic buffer sharing (egress buffering) across slices is configured with the **hardware qos dynamic-buffer-sharing** command. Following the command, you must reload the switch to enable the dynamic buffering.

Buffer sharing is enabled by dynamic bank allocation (1 bank = 4k cells, 1 cell = 416 bytes) and controlled by a global controller (eCPU) that manages the banks being distributed among slices. Dynamic buffer sharing provides six reserved banks (10MB) for each slice and twelve banks for sharing across slices (20MB).

Note Dynamic Buffer Sharing is supported only on Nexus 9300-FX2 platform switches, see Nexus Switch Platform Support Matrix

Monitor the QoS Packet Buffer

The Cisco Nexus 9000 Series device has a 12-MB buffer memory that divides into a dedicated per port and dynamic shared memory. Each front-panel port has four unicast queues and four multicast queues in egress. In the scenario of burst or congestion, each egress port consumes buffers from the dynamic shared memory.

You can display the real-time and peak status of the shared buffer per port. All counters are displayed in terms of the number of cells. Each cell is 208 bytes in size. You can also display the global level buffer consumption in terms of consumption and available number of cells.



Note Monitoring the shared buffer on ALE enabled devices is not supported for the port level.



Note In the examples shown in this section, the port numbers are Broadcom ASIC ports.

This example shows how to clear the system buffer maximum cell usage counter.

```
switch# clear counters buffers
Max Cell Usage has been reset successfully
```

This example shows how to set a buffer utilization threshold for a specific module.

```
switch(config)# hardware profile buffer info port-threshold module 1 threshold 10
Port threshold changed successfully
```



Note

- The buffer threshold feature is not enabled for ports if they have a no-drop class configured (PFC).
- The configured threshold buffer count is checked every 5 seconds against all the buffers used by that port across all the queues of that port.
- You can configure the threshold percentage configuration for all modules or for a specific module, which is applied to all ports. The default threshold value is 90% of the switch cell count of shared pool SP-0. This configuration applies to both Ethernet (front panel) and internal (HG) ports.
- The buffer threshold feature is not supported for ACI capable device ports.

This example shows how to display the interface hardware mappings.

```
switch# show interface hardware-mappings
Legends:
       SMod  - Source Mod. 0 is N/A
       Unit  - Unit on which port resides. N/A for port channels
       HPort - Hardware Port Number or Hardware Trunk Id:
       FPort - Fabric facing port number. 255 means N/A
       NPort - Front panel port number
       VPort - Virtual Port Number. -1 means N/A
```

```
--------------------------------------------------------------------
Name       Ifindex  Smod Unit HPort FPort NPort VPort
--------------------------------------------------------------------
Eth2/1     1a080000 4    0    13    255   0     -1
Eth2/2     1a080200 4    0    14    255   1     -1
Eth2/3     1a080400 4    0    15    255   2     -1
Eth2/4     1a080600 4    0    16    255   3     -1
Eth2/5     1a080800 4    0    17    255   4     -1
Eth2/6     1a080a00 4    0    18    255   5     -1
Eth2/7     1a080c00 4    0    19    255   6     -1
Eth2/8     1a080e00 4    0    20    255   7     -1
Eth2/9     1a081000 4    0    21    255   8     -1
Eth2/10    1a081200 4    0    22    255   9     -1
Eth2/11    1a081400 4    0    23    255   10    -1
Eth2/12    1a081600 4    0    24    255   11    -1
Eth2/13    1a081800 4    0    25    255   12    -1
Eth2/14    1a081a00 4    0    26    255   13    -1
Eth2/15    1a081c00 4    0    27    255   14    -1
Eth2/16    1a081e00 4    0    28    255   15    -1
Eth2/17    1a082000 4    0    29    255   16    -1
Eth2/18    1a082200 4    0    30    255   17    -1
Eth2/19    1a082400 4    0    31    255   18    -1
Eth2/20    1a082600 4    0    32    255   19    -1
Eth2/21    1a082800 4    0    33    255   20    -1
Eth2/22    1a082a00 4    0    34    255   21    -1
Eth2/23    1a082c00 4    0    35    255   22    -1
Eth2/24    1a082e00 4    0    36    255   23    -1
```

Configuration Examples for Queuing and Scheduling

In this section, you can find examples of configuring queuing and scheduling.

Note The default system classes type queuing match based on qos-group (by default all traffic matches to qos-group 0, and this default queue gets 100% bandwidth). Create a type QoS policy that first sets the qos-group in order to drive the correct matching for the type queuing classes and policies.

Example: Configuring WRED on Egress Queues

The following example shows how to configure the WRED feature on an egress queue:

```
configure terminal
  class-map type queuing match-any c-out-q1
    match qos-group 1
  class-map type queuing match-any c-out-q2
    match qos-group 1
  policy-map type queuing wred
    class type queuing c-out-q1
      random-detect minimum-threshold 10 bytes maximum-threshold 1000 bytes
    class type queuing c-out-q2
      random-detect threshold burst-optimized ecn
```

Example: Configuring Traffic Shaping

The following example shows how to configure traffic shaping using 500 mbps and 1000 mbps for respective classes::

```
configure terminal
  class-map type queuing match-any c-out-q1
    match qos-group 1
  class-map type queuing match-any c-out-q2
    match qos-group 1
policy-map type queuing pqu
  class type queuing c-out-8q-q3
    bandwidth percent 20
    shape min 100 mbps max 500 mbps
  class type queuing c-out-8q-q2
    bandwidth percent 30
    shape min 200 mbps max 1000 mbps
  class type queuing c-out-8q-q-default
    bandwidth percent 50
  class type queuing c-out-8q-q1
    bandwidth percent 0
  class type queuing c-out-8q-q4
    bandwidth percent 0
  class type queuing c-out-8q-q5
    bandwidth percent 0
  class type queuing c-out-8q-q6
    bandwidth percent 0
  class type queuing c-out-8q-q7
    bandwidth percent 0
system qos
  service-policy type queuing output pqu
```