

Congestion Avoidance

Congestion avoidance techniques monitor traffic flow in an effort to anticipate and avoid congestion at common network bottlenecks. Avoidance techniques are implemented before congestion occurs as compared with congestion management techniques that control congestion after it has occurred.

This chapter provides details regarding congestion avoidance techniques.

- Prerequisites for Configuring Modular QoS Congestion Avoidance, on page 1
- Random Early Detection and TCP, on page 1
- Tail Drop, on page 2
- Configuring Random Early Detection, on page 2
- Configuring Weighted Random Early Detection, on page 4
- Configuring Tail Drop, on page 6

Prerequisites for Configuring Modular QoS Congestion Avoidance

This prerequisite is required for configuring QoS congestion avoidance on your network:

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Random Early Detection and TCP

The Random Early Detection (RED) congestion avoidance technique takes advantage of the congestion control mechanism of TCP. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. Assuming the packet source is using TCP, it decreases its transmission rate until all packets reach their destination, indicating that the congestion is cleared. You can use RED as a way to cause TCP to slow transmission of packets. TCP not only pauses, but it also restarts quickly and adapts its transmission rate to the rate that the network can support.

RED distributes losses in time and maintains normally low queue depth while absorbing traffic bursts. When enabled on an interface, RED begins dropping packets when congestion occurs at a rate you select during configuration.

Queue-limit for WRED

Queue-limit is used to fine-tune the number of buffers available for each queue. It can only be used on a queuing class. Default queue limit is --------- ms of the service rate for the given queue. The service rate is the sum of minimum guaranteed bandwidth and bandwidth remaining assigned to a given class either implicitly or explicitly.

The queue-limit is rounded up to one of the following values: 8 KB, 16 KB, 24 KB, 32 KB, 48 KB, 64 KB, 96 KB, 128 KB, 192 KB, 256 KB, 384 KB, 512 KB, 768 KB, 1024 KB, 1536 KB, 2048 KB, 3072 KB, 4196 KB, 8192 KB, 16394 KB, 32768 KB, 65536 KB, 131072 KB, or 262144 KB.

Tail Drop

Tail drop is a congestion avoidance technique that drops packets when an output queue is full until congestion is eliminated. Tail drop treats all traffic flow equally and does not differentiate between classes of service.

Configuring Random Early Detection

This configuration task is similar to that used for WRED except that the **random-detect precedence** command is not configured and the **random-detect** command with the **default** keyword must be used to enable RED.

Restrictions

If you configure the **random-detect default** command on any class including class-default, you must configure one of the following commands:

- **shape average or bandwidth**

For the **random-detect** command to take effect, you must configure either the **shape average or bandwidth** command in the user defined policy map class. This dependency is not applicable to the policy map class-default.

Procedure

Step 1 **configure**

Step 2 **policy-map** *policy-map-name*

Example:

```
RP/0/(config)# policy-map policy1
```

Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters the policy map configuration mode.

Step 3 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class1
```

Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration mode.

Step 4 **random-detect** {**cos** *value* | **default** | **discard-class** *value* | **dscp** *value* | **exp** *value* | **precedence** *value* | *min-threshold* [*units*] *max-threshold* [*units*] }

Example:

```
RP/0/(config-pmap-c)# random-detect default
```

Enables RED with default minimum and maximum thresholds.

Step 5 **bandwidth** {*bandwidth* [*units*] | **percent** *value*}

Example:

```
RP/0/(config-pmap-c)# bandwidth percent 30
```

(Optional) Specifies the bandwidth allocated for a class belonging to a policy map.

or

(Optional) Specifies how to allocate leftover bandwidth to various classes.

Step 6 **shape average** {**percent** *percentage* | *value* [*units*]}

Example:

```
RP/0/(config-pmap-c)# shape average percent 50
```

(Optional) Shapes traffic to the specified bit rate or a percentage of the available bandwidth.

Note
This step can be used if the bandwidth is not configured (step 5).

Step 7 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 8 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to the global configuration mode.

Step 9 **interface** *type interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/2/0/0
```

Enters the configuration mode and configures an interface.

Step 10 **service-policy** {**input** | **output**} *policy-map*

Example:

```
RP/0/(config-if)# service-policy output policy1
```

Attaches a policy map to an input or output interface to be used as the service policy for that interface. In this example, the traffic policy evaluates all traffic leaving that interface.

Step 11 **commit**

Configuring Weighted Random Early Detection

WRED drops packets selectively based on IP precedence. IP precedences are assigned to packets as they enter the network. WRED uses these precedences to determine how to treat different types of traffic.

Configure WRED using the **random-detect** command and different CoS, DSCP, EXP, and discard-class values. The value can be range or a list of values that are valid for that field. You can also use minimum and maximum queue thresholds to determine the dropping point.

When a packet arrives, the following actions occur:

- The average queue size is calculated.
- If the average queue size is less than the minimum queue threshold, the arriving packet is queued.
- If the average queue size is between the minimum queue threshold for that type of traffic and the maximum threshold for the interface, the packet is either dropped or queued, depending on the packet drop probability for that type of traffic.
- If the average queue size is greater than the maximum threshold, the packet is dropped.

Restrictions

You cannot configure WRED in a class that has been set for priority queueing (PQ).

You cannot use the **random-detect** command in a class configured with the **priority** command.

Procedure

Step 1 **configure**

Step 2 **policy-map** *policy-name*

Example:

```
RP/0/(config)# policy-map policy1
```

Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters the policy map configuration mode.

Step 3 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class1
```

Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration mode.

Step 4 **random-detect discard-class** *class-id* [*min-packets max-packets*] | **default**

Example:

```
RP/0/(config-pmap-c)# random-detect discard-class 1 1 packet 2 packets
```

Modifies the minimum and maximum packet thresholds for the discard class.

Step 5 **bandwidth** {*bandwidth* [*units*] | **percent** *value*}

Example:

```
RP/0/(config-pmap-c)# bandwidth percent 30
```

(Optional) Specifies the bandwidth allocated for a class belonging to a policy map. This example guarantees 30 percent of the interface bandwidth to class class1.

Step 6 **shape average** {**percent** *percentage* | *value* [*units*]}

Example:

```
RP/0/(config-pmap-c)# shape average percent 50
```

(Optional) Shapes traffic to the specified bit rate or a percentage of the available bandwidth.

Note
This step can be used if the bandwidth was not configured earlier (using step 5).

Step 7 **queue-limit** *value* [*units*]

Example:

```
RP/0/(config-pmap-c)# queue-limit 50 ms
```

(Optional) Changes queue-limit to fine-tune the amount of buffers available for each queue. The default queue-limit is 100 ms of the service rate for a non-priority class and 10ms of the service rate for a priority class.

Step 8 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to the global configuration mode.

Step 9 **interface** *type interface-path-id*

Example:

```
RP/0/(config)# interface 0/2/0/0
```

Enters the configuration mode and configures an interface.

Step 10 **service-policy** {**input** | **output**} *policy-map*

Example:

```
RP/0/(config-if)# service-policy output policy1
```

Attaches a policy map to an input or output interface to be used as the service policy for that interface.

- In this example, the traffic policy evaluates all traffic leaving that interface.
- Ingress policies are not valid; the **bandwidth** and **bandwidth remaining** commands cannot be applied to ingress policies.

Step 11 **commit**

Running configuration for WRED

```
policy-map egress_WRED_POLICY_L3
class class2
  shape average 5 gbps
  random-detect discard-class 0 0 bytes 10 bytes
!
```

Configuring Tail Drop

Packets satisfying the match criteria for a class accumulate in the queue reserved for the class until they are serviced. The **queue-limit** command is used to define the maximum threshold for a class. When the maximum threshold is reached, enqueued packets to the class queue result in tail drop (packet drop).

The **queue-limit** value uses the guaranteed service rate (GSR) of the queue as the reference value for the **queue_bandwidth**. If the class has bandwidth percent associated with it, the **queue-limit** is set to a proportion of the bandwidth reserved for that class.

If the GSR for a queue is zero, use the following to compute the default **queue-limit**:

- 1 percent of the interface bandwidth for queues in a nonhierarchical policy.
- 1 percent of parent maximum reference rate for hierarchical policy.

  The parent maximum reference rate is the minimum of parent shape, policer maximum rate, and the interface bandwidth.

Note The default **queue-limit** is set to bytes of 100 ms of queue bandwidth. The following formula is used to calculate the default queue limit (in bytes):??bytes = (100 ms / 1000 ms) * queue_bandwidth kbps)) / 8

The default **queue-limit** is set as follows:

default queue limit (in bytes) = (<200|100|10> ms * queue_bandwidth kbps) / 8

Note You can configure the queue limit in all the priority classes.

The default **queue-limit** is set as follows:

- **For priority class**

  default queue limit (in bytes) = (<10> ms * queue_bandwidth kbps) / 8

- **For non-priority class**

  default queue limit (in bytes) = (<100> ms * queue_bandwidth kbps) / 8



Note You can configure a maximum queue threshold up to 1GB, which translates to 80ms of buffering for 100Gbps queue.

Restrictions

- When configuring the **queue-limit** command in a class, you must configure one of the following commands: **priority**, **shape average**, **bandwidth**, or **bandwidth remaining**, except for the default class.

Procedure

Step 1 **configure**

Step 2 **policy-map** *policy-name*

Example:

```
RP/0/(config)# policy-map policy1
```

Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and also enters the policy map configuration mode.

Step 3 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class1
```

Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration mode.

Step 4 (optional) **queue-limit** *value* [*units*]

Example:

```
RP/0/(config-pmap-c)# queue-limit 1000000 bytes
```

Specifies or modifies the maximum the queue can hold for a class policy configured in a policy map. The default value of the *units* argument is **packets**. In this example, when the queue limit reaches 1,000,000 bytes, enqueued packets to the class queue are dropped.

Step 5 **priority** [**level** *priority-level*]

Example:

```
RP/0/(config-pmap-c)# priority level 1
```

Specifies priority to a class of traffic belonging to a policy map.

Step 6 **police rate percent** *percentage*

Example:

```
RP/0/(config-pmap-c)# police rate percent 30
```

Configures traffic policing.

Step 7 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class2
```

Specifies the name of the class whose policy you want to create or change. In this example, class2 is configured.

Step 8 **bandwidth** {*bandwidth* [*units*] | **percent** *value*}

Example:

```
RP/0/(config-pmap-c)# bandwidth percent 30
```

(Optional) Specifies the bandwidth allocated for a class belonging to a policy map. This example guarantees 30 percent of the interface bandwidth to class class2.

Step 9 (optional) **queue-limit** *value* [*units*]

Example:

```
RP/0/(config-pmap-c)# queue-limit 1000000 bytes
```

Specifies or modifies the maximum the queue can hold for a class policy configured in a policy map. The default value of the *units* argument is **packets**. In this example, when the queue limit reaches 1,000,000 bytes, enqueued packets to the class queue are dropped.

Step 10 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class2
```

Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration mode.

Step 11 **bandwidth remaining percent** *value*

Example:

```
RP/0/(config-pmap-c)# bandwidth remaining percent 20
```

(Optional) Specifies how to allocate leftover bandwidth to various classes. This example allocates 20 percent of the leftover interface bandwidth to class class2.

Step 12 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class3
```

Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration mode.

Step 13 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 14 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to the global configuration mode.

Step 15 **interface** *type interface-path-id*

Example:

```
RP/0/(config)# interface HundredGigE 0/7/0/0
```

Enters the configuration mode and configures an interface.

Step 16 **service-policy** {**input** | **output**} *policy-map*

Example:

```
RP/0/(config-if)# service-policy output policy1
```

Attaches a policy map to an input or output interface to be used as the service policy for that interface. In this example, the traffic policy evaluates all traffic leaving that interface.

Step 17 **commit**

Running configuration for tail drop

```
policy-map egress_BRRpriority_POLICY
class CLASS_3_egress_BRRpriorityIPV4DSCP
  bandwidth remaining ratio 1
!
class CLASS_1_egress_BRRpriorityIPV4DSCP
  bandwidth remaining ratio 10
!
class class-default
!
end-policy-map
!
```