Configuring Modular QoS 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.

Congestion avoidance is achieved through packet dropping. Cisco IOS XR software supports these quality of service (QoS) congestion avoidance techniques that drop packets:

- Random early detection (RED)
- Weighted random early detection (WRED)
- Tail drop

The module describes the concepts and tasks related to these congestion avoidance techniques.

Line Card, SIP, and SPA Support

Feature	ASR 9000 Ethernet Line Cards	SIP 700 for the ASR 9000
Random Early Detection	yes	yes
Weighted Random Early Detection	yes	yes
Tail Drop	yes	yes

Feature History for Configuring Modular QoS Congestion Avoidance on Cisco ASR 9000 Series Routers

Release	Modification
Release 3.7.2	The Congestion Avoidance feature was introduced on ASR 9000 Ethernet Line Cards. The Random Early Detection, Weighted Random Early Detection, and Tail Drop features were introduced on ASR 9000 Ethernet Line Cards.
Release 3.9.0	The Random Early Detection, Weighted Random Early Detection, and Tail Drop features were supported on the SIP 700 for the ASR 9000.

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.

Information About Configuring Modular QoS Congestion Avoidance

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 100 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 the nearest power of 2, and depending on the line cards on your system, the queue-limit values vary. To check the current queue-limit for class-default, use the **show qos interface** command.

Because WRED needs a queue to operate on, the class that WRED is applied on must have either a bandwidth statement or a parent policy with a shaper if WRED is applied only on a class default queue.

Examples

The following policy configuration does not use the queue limit because the policy is flat and doesn’t have a designated queue on which it operates.

```
policy-map incorrect-flat
class class-default
  random-detect dscp 16 250 packets 500 packets
  queue-limit 158000 kbytes
```

The following policy configuration can use the queue limit because it uses a parent policy map with the shape average command.

```
policy-map parent
class class-default
shape average 100 mbps
service-policy child

policy-map child
class class-default
  random-detect dscp 16 250 packets 500 packets
  queue-limit 158000 kbytes
```

The following policy configuration can use the queue limit because it provides a flat policy with a shaped queue through the bandwidth command for the class-default.

```
policy-map correct-flat
class class-default
  bandwidth 100 mbps
  random-detect dscp 16 250 packets 500 packets
  queue-limit 158000 kbytes
```

Tail Drop and the FIFO Queue

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. It manages the packets that are unclassified, placed into a first-in, first-out (FIFO) queue, and forwarded at a rate determined by the available underlying link bandwidth.

See the “Default Traffic Class” section of the “Configuring Modular Quality of Service Packet Classification and Marking on Cisco ASR 9000 Series Routers”.

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**
- **bandwidth**
- **bandwidth remaining**

SUMMARY STEPS

1. **configure**

2. **policy-map** *policy-map-name*
3. **class** *class-name*
4. **random-detect** {**cos** *value* | **default** | **discard-class** *value* | **dscp** *value* | **exp** *value* | **precedence** *value* | *min-threshold* [*units*] *max-threshold* [*units*] }
5. **bandwidth** {*bandwidth* [*units*] | **percent** *value*} or **bandwidth remaining** [**percent** *value* | **ratio** *ratio-value*
6. **shape average** {**percent** *percentage* | *value* [*units*]}
7. **exit**
8. **exit**
9. **interface** *type interface-path-id*
10. **service-policy** {**input** | **output**} *policy-map*
11. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**policy-map** *policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(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/RSP0/CPU0:router(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/RSP0/CPU0:router(config-pmap-c)# random-detect default`
		Enables RED with default minimum and maximum thresholds.

Step 5	**bandwidth** {*bandwidth* [*units*] | **percent** *value*} or **bandwidth remaining** [**percent** *value* | **ratio** *ratio-value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth percent 30` or	(Optional) Specifies the bandwidth allocated for a class belonging to a policy map. or (Optional) Specifies how to allocate leftover bandwidth to various classes.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth remaining percent 20`	

Step 6	**shape average** {**percent** *percentage* | *value* [*units*]}
	Example:
	`RP/0/RSP0/CPU0:router(config-pmap-c)# shape average percent 50`
		(Optional) Shapes traffic to the specified bit rate or a percentage of the available bandwidth.

Step 7	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# exit`	Returns the router to policy map configuration mode.
Step 8	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# exit`	Returns the router to global configuration mode.
Step 9	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(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/RSP0/CPU0:router(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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes, and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration mode, without committing the configuration changes.

Configuring Weighted Random Early Detection

WRED drops packets selectively based on any specified criteria, such as CoS, DSCP, EXP, discard-class, or precedence . WRED uses these matching criteria 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:

- If the queue size is less than the minimum queue threshold, the arriving packet is queued.
- If the 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 queue size is greater than the maximum threshold, the packet is dropped.

Restrictions

- When configuring the **random-detect dscp** command, you must configure one of the following commands: **shape average**, **bandwidth**, and **bandwidth remaining**.

Note The Cisco ASR 9000 Series ATM SPA supports only time-based WRED thresholds. Therefore, if you try to configure the WRED threshold using the **random-detect default** command with bytes or packet as the threshold units, the "Unsupported WRED unit on ATM interface" error occurs.

- Only two minimum and maximum thresholds (each with different match criteria) can be configured per class.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy-name*
3. **class** *class-name*
4. **random-detect dscp** *dscp-value min-threshold* [*units*] *max-threshold* [*units*]
5. **bandwidth** {*bandwidth* [*units*] | **percent** *value*} or **bandwidth remaining** [**percent** *value* | **ratio** *ratio-value]*
6. **bandwidth** {*bandwidth* [*units*] | **percent** *value*}
7. **bandwidth remaining percent** *value*
8. **shape average** {**percent** *percentage* | *value* [*units*]}
9. **queue-limit** *value* [*units*]
10. **exit**
11. **interface** *type interface-path-id*
12. **service-policy** {**input** | **output**} *policy-map*
13. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**policy-map** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(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/RSP0/CPU0:router(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 dscp** *dscp-value min-threshold* [*units*] *max-threshold* [*units*] **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# random-detect dscp af11 1000000 bytes 2000000 bytes`	Modifies the minimum and maximum packet thresholds for the DSCP value. • Enables WRED. • *dscp-value*—Number from 0 to 63 that sets the DSCP value. Reserved keywords can be specified instead of numeric values. • *min-threshold*—Minimum threshold in the specified units. When the average queue length reaches the minimum threshold, WRED randomly drops some packets with the specified DSCP value. • *max-threshold*—Maximum threshold in the specified units. When the average queue length exceeds the maximum threshold, WRED drops all packets with the specified DSCP value. • *units*—Units of the threshold value. This can be **bytes**, **gbytes**, **kbytes**, **mbytes**, **ms** (milliseconds), **packets**, or **us** (microseconds). The default is **packets**. • This example shows that for packets with DSCP AF11, the WRED minimum threshold is 1,000,000 bytes and maximum threshold is 2,000,000 bytes.
Step 5	**bandwidth** {*bandwidth* [*units*] | **percent** *value*} or **bandwidth remaining** [**percent** *value* | **ratio** *ratio-value]* **Example:** `RP/0/RSP0/CPU0:router(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.

	Command or Action	Purpose
	or `RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth remaining percent 20`	

Step 6	**bandwidth** {*bandwidth* [*units*] | **percent** *value*}
	Example:
	`RP/0/RSP0/CPU0:router(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 7	**bandwidth remaining percent** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth remaining percent 20`	(Optional) Specifies how to allocate leftover bandwidth to various classes. • The remaining bandwidth of 70 percent is shared by all configured classes. • In this example, class class1 receives 20 percent of the 70 percent.

Step 8	**shape average** {**percent** *percentage* | *value* [*units*]}
	Example:
	`RP/0/RSP0/CPU0:router(config-pmap-c)# shape average percent 50`
		(Optional) Shapes traffic to the specified bit rate or a percentage of the available bandwidth.

Step 9	**queue-limit** *value* [*units*] **Example:** `RP/0/RSP0/CPU0:router(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. **Note** Even though this command is optional, it is recommended that you use it to fine-tune the queue limit, instead of relying on your system default settings. If the queue limit is too large, the buffer consumption goes up, resulting in delays. On the other hand, too small a queue limit may result in extra drops while allowing for faster rate adaption.
Step 10	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# exit`	Returns the router to global configuration mode.
Step 11	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/2/0/0`	Enters the configuration mode and configures an interface.

	Command or Action	Purpose
Step 12	**service-policy** {**input** | **output**} *policy-map* **Example:** `RP/0/RSP0/CPU0:router(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 13	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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

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.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy-name*
3. **class** *class-name*
4. **queue-limit** *value* [*units*]
5. **priority** [**level** *priority-level*]
6. **police rate percent** *percentage*
7. **class** *class-name*
8. **bandwidth** {*bandwidth* [*units*] | **percent** *value*}
9. **bandwidth remaining percent** *value*
10. **exit**
11. **exit**
12. **interface** *type interface-path-id*
13. **service-policy** {**input** | **output**} *policy-map*
14. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**policy-map** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(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/RSP0/CPU0:router(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	**queue-limit** *value* [*units*] **Example:** `RP/0/RSP0/CPU0:router(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/RSP0/CPU0:router(config-pmap-c)# priority level 1`	Specifies priority to a class of traffic belonging to a policy map.

	Command or Action	Purpose
Step 6	**police rate percent** *percentage* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# police rate percent 30`	Configures traffic policing.

Step 7	**class** *class-name*
	Example:
	`RP/0/RSP0/CPU0:router(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/RSP0/CPU0:router(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	**bandwidth remaining percent** *value* **Example:** `RP/0/RSP0/CPU0:router(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 10	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# exit`	Returns the router to policy map configuration mode.
Step 11	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# exit`	Returns the router to global configuration mode.
Step 12	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface POS 0/2/0/0`	Enters the configuration mode and configures an interface.
Step 13	**service-policy** {**input** | **output**} *policy-map* **Example:** `RP/0/RSP0/CPU0:router(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 14	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

	Command or Action	Purpose
		• **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Additional References

These sections provide references related to implementing QoS congestion avoidance.

Related Documents

Related Topic	Document Title
Initial system bootup and configuration	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
QoS commands	*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*
User groups and task IDs	"Configuring AAA Services on Cisco ASR 9000 Series Router" module of Cisco Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html