Configure Priority Flow Control

Priority Flow Control Overview

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Priority Flow Control on Cisco 8808 and Cisco 8812 Modular Chassis Line Cards	Release 7.5.3	Priority Flow Control is now supported on the following line card in the buffer-internal mode: • 88-LC0-34H14FH The feature is supported in the buffer-internal and buffer-extended modes on: • 88-LC0-36FH Apart from the buffer-external mode, support for this feature now extends to the buffer-internal mode on the following line cards: • 88-LC0-36FH-M • 8800-LC-48H
Shortlink Priority Flow Control	Release 7.3.3	This feature and the hw-module profile priority-flow-control command are supported on 88-LC0-36FH line card.

Feature Name	Release Information	Feature Description
Priority Flow Control Support on Cisco 8800 36x400 GbE QSFP56-DD Line Cards (88-LC0-36FH-M)	Release 7.3.15	This feature and the hw-module profile priority-flow-control command are supported on 88-LC0-36FH-M and 8800-LC-48H line cards. All previous functionalities and benefits of this feature are available on these line cards. However, the buffer-internal mode is not supported. In addition, to use the buffer-extended mode on these line cards, you are required to configure the performance capacity or headroom values. This configuration requirement ensures that you can better provision and balance workloads to achieve lossless behavior, which in turn ensures efficient use of bandwidth and resources.
Priority Flow Control	Release 7.3.1	This feature and the hw-module profile priority-flow-control command are not supported.

Priority-based Flow Control (IEEE 802.1Qbb), which is also referred to as Class-based Flow Control (CBFC) or Per Priority Pause (PPP), is a mechanism that prevents frame loss that is due to congestion. PFC is similar to 802.x Flow Control (pause frames) or link-level flow control (LFC). However, PFC functions on a per class-of-service (CoS) basis.

During congestion, PFC sends a pause frame to indicate the CoS value to pause. A PFC pause frame contains a 2-octet timer value for each CoS that indicates the length of time to pause the traffic. The unit of time for the timer is specified in pause quanta. A quanta is the time required for transmitting 512 bits at the speed of the port. The range is from 0 through 65535 quanta.

PFC asks the peer to stop sending frames of a particular CoS value by sending a pause frame to a well-known multicast address. This pause frame is a one-hop frame and isn't forwarded when received by the peer. When the congestion mitigates, the router stops sending the PFC frames to the upstream node.

You can configure PFC for each line card using the **hw-module profile priority-flow-control** command in one of two modes:

- buffer-internal
- buffer-extended

Note PFC threshold configurations are deprecated in pause command. Use the hw-module profile priority-flow-control command to configure PFC threshold configurations.

Tip You can programmatically retrieve the operational state of the PFC configuration using `Cisco-IOS-XR-ofa-npu-pfc-oper.yang` Cisco IOS XR native data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

Related Topics

- Configure Priority Flow Control, on page 5
- Priority Flow Control Watchdog Overview, on page 12

buffer-internal mode

Use this mode if PFC-enabled devices aren't more than 1 km apart.

You can set values for pause-threshold, headroom (both related to PFC), and ECN for the traffic class using the **hw-module profile priority-flow-control** command in this mode. The buffer-internal configuration applies to all ports that the line card hosts, which mean that you can configure a set of these values per line card.

The existing queue limit and ECN configuration in the queueing policy attached to the interface has no impact in this mode.

The effective queue limit for this mode = pause-threshold + headroom (in bytes)

Restrictions and Guidelines

The following restrictions and guidelines apply while configuring the PFC threshold values using the buffer-internal mode.

- The PFC feature isn't supported on fixed chassis systems.
- Ensure that there’s no breakout configured on a chassis that has the PFC configured. Configuring PFC and breakout on the same chassis may lead to unexpected behavior, including traffic loss.
- The feature isn't supported on bundle and non-bundle sub-interface queues.
- The feature is supported on 40GbE, 100 GbE, and 400 GbE interfaces.
- The feature isn't supported in the 4xVOQ queueing mode.
- The feature isn't supported when sharing of VOQ counters is configured.

buffer-extended mode

Use this mode for PFC-enabled devices with long-haul connections.

You can set the value for pause-threshold using the **hw-module profile priority-flow-control** command in this mode. You must, however, configure the queuing policy attached to the interface to set the ECN and queueing limits. The buffer-extended configuration applies to all ports that the line card hosts, which mean that you can configure a set of these values per line card.

Configuration Guidelines

- Important points while configuring the buffer-extended mode on 88-LC0-36FH-M line cards:
  - Apart from pause-threshold, you must also configure values for headroom.
  - The headroom value range is from 4 through 75000.
  - Specify pause-threshold and headroom values in units of kilobytes (KB) or megabytes (MB).
- Important points while configuring the buffer-extended mode on 8800-LC-48H line cards:
  - Configure values only for **pause-threshold**. Don’t configure headroom values.
  - Configure **pause-threshold** in units of milliseconds (ms) or microseconds.
  - Don’t use units of kilobytes (KB) or megabytes (MB) units, even though the CLI displays them as options. Only use units of milliseconds (ms) or microseconds.

(Also see Configure Priority Flow Control, on page 5)

Important Considerations

- If you configure PFC values in the buffer-internal mode, then the ECN value for the line card is derived from the buffer-internal configuration. If you configure PFC values in the buffer-extended mode, then the ECN value is derived from the policy map. (For details on the ECN feature, see Explicit Congestion Notification.)
- The buffer-internal and buffer-extended modes can't coexist on the same line card.
- For Cisco 8808 and Cisco 8812 chassis, configure PFC on all line cards in the chassis, regardless of whether you're configuring the buffer-internal or buffer-extended mode. Otherwise, your network may experience traffic loss.
- If you add or remove traffic-class actions on a line card, you must reload the line card.
- When using the buffer-internal mode, you can change values of the following parameters without having to reload the line card. However, if you add a new traffic class and configure these values for the first time on that traffic class, you must reload the line card for the values to come into effect.
  - pause-threshold
  - headroom
  - ECN
- If you add or remove ECN configuration using the **hw-module profile priority-flow-control** command, you must reload the line card for the ECN changes to take effect.
- The PFC threshold value ranges for the buffer-internal mode are as follows.

Threshold	Configured (bytes)
pause (min)	307200
pause (max)	422400
headroom (min)	345600
headroom (max)	537600
ecn (min)	153600
ecn (max)	403200

- For a traffic-class, the ECN value must always be lesser than the configured pause-threshold value.
- The combined configured values for pause-threshold and headroom must not exceed 844800 bytes. Else, the configuration is rejected.
- The pause-threshold value range for buffer-extended mode is from 2 milliseconds (ms) through 25 ms and from 2000 microseconds through 25000 microseconds.

Hardware Support for Priority Flow Control

The table lists the PIDs that support PFC per release and the PFC mode in which the support is available.

Table 2: PFC Hardware Support Matrix

Release	PID	PFC Mode
Release 7.5.3	88-LC0-36FH	buffer-extended and buffer-internal
	88-LC0-36FH-M	buffer-extended and buffer-internal
	8800-LC-48H	buffer-extended and buffer-internal
	88-LC0-34H14FH	buffer-internal
Release 7.3.15	• 88-LC0-36FH-M • 88-LC0-36FH	buffer-extended
Release 7.0.11	8800-LC-48H	buffer-internal

Configure Priority Flow Control

You can configure PFC to enable the no-drop behavior for the CoS as defined by the active network QoS policy.

Note The system enables shortlink PFC by default when you enable PFC.

YANG Data Model

You can programmatically manage the flow control parameters using the `openconfig-qos.yang` data model. To get started with using data models, see the Programmability Configuration Guide for Cisco 8000 Series Routers.

Configuration Example

You must accomplish the following to complete the PFC configuration:

1. Enable PFC at the interface level.
2. Configure ingress classification policy.
3. Attach the PFC policy to the interface.
4. Configure PFC threshold values using either the buffer-internal or buffer-extended mode.

```
Router# configure
Router(config)# priority-flow-control mode on
/*Configure ingress classification policy*/
Router(config)# class-map match-any prec7
Router(config-cmap)# match precedence
Router(config)# class-map match-any tc7
/*Ingress policy attach*/
Router(config-if)# service-policy input QOS_marking
/*Egress policy attach*/
Router(config-if)# service-policy output qos_queuing
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#show controllers npu priority-flow-control location <loc>
```

Running Configuration

```
*Interface Level*
interface HundredGigE0/0/0/0
   priority-flow-control mode on

*Ingress:*
class-map match-any prec7

 match precedence 7

 end-class-map

!

class-map match-any prec6

 match precedence 6

 end-class-map

!

class-map match-any prec5

 match precedence 5
```

```
 end-class-map

!

class-map match-any prec4

 match precedence 4

 end-class-map

!

class-map match-any prec3
match precedence 3
end-class-map
!
class-map match-any prec2
match precedence 2
end-class-map
!
class-map match-any prec1
match precedence 1
end-class-map
!
!
policy-map QOS_MARKING
class prec7
  set traffic-class 7
  set qos-group 7
!
class prec6
  set traffic-class 6
  set qos-group 6
!
class prec5
  set traffic-class 5
  set qos-group 5
!
class prec4
  set traffic-class 4
  set qos-group 4
!
class prec3
  set traffic-class 3
  set qos-group 3
!
class prec2
  set traffic-class 2
  set qos-group 2
!
class prec1
  set traffic-class 1
  set qos-group 1
!
class class-default
  set traffic-class 0
  set qos-group 0
!

*Egress:*
class-map match-any tc7
 match traffic-class 7
 end-class-map
!
```

Configure Priority Flow Control

```
class-map match-any tc6
 match traffic-class 6
 end-class-map
!
class-map match-any tc5
 match traffic-class 5
 end-class-map

!

class-map match-any tc4

 match traffic-class 4

 end-class-map

!

class-map match-any tc3

 match traffic-class 3

 end-class-map

!

class-map match-any tc2
match traffic-class 2
end-class-map
!
class-map match-any tc1
match traffic-class 1
end-class-map
!
policy-map QOS_QUEUING
class tc7
  priority level 1
  shape average percent 10
!
class tc6
  bandwidth remaining ratio 1
  queue-limit 100 ms
!
class tc5
  bandwidth remaining ratio 20
  queue-limit 100 ms
!
class tc4
  bandwidth remaining ratio 20
  random-detect ecn
  random-detect 6144 bytes 100 mbytes
!
class tc3
  bandwidth remaining ratio 20
  random-detect ecn
  random-detect 6144 bytes 100 mbytes
!
class tc2
  bandwidth remaining ratio 5
  queue-limit 100 ms
!
class tc1
  bandwidth remaining ratio 5
  queue-limit 100 ms
```

```
!
class class-default
  bandwidth remaining ratio 20
  queue-limit 100 ms
!
[buffer-extended]

hw-module profile priority-flow-control location 0/0/CPU0
 buffer-extended traffic-class 3 pause-threshold 10 ms
 buffer-extended traffic-class 4 pause-threshold 10 ms
!

[buffer-internal]

hw-module profile priority-flow-control location 0/1/CPU0
 buffer-internal traffic-class 3 pause-threshold 403200 bytes headroom 441600 bytes ecn
224640 bytes
 buffer-internal traffic-class 4 pause-threshold 403200 bytes headroom 441600 bytes ecn
224640 bytes
```

Verification

```
Router#sh controllers hundredGigE0/0/0/22 priority-flow-control
Priority flow control information for interface HundredGigE0/0/0/22:
Priority Flow Control:
Total Rx PFC Frames : 0
Total Tx PFC Frames : 313866
Rx Data Frames Dropped: 0
CoS Status Rx Frames
--- ------ ----------
0   on     0
1   on     0
2   on     0
3   on     0
4   on     0
5   on     0
6   on     0
7   on     0

/*[buffer-internal]*/
Router#show controllers hundredGigE 0/9/0/24 priority-flow-control

Priority flow control information for interface HundredGigE0/9/0/24:

Priority Flow Control:
Total Rx PFC Frames : 0
Total Tx PFC Frames : 313866
Rx Data Frames Dropped: 0
CoS Status Rx Frames
--- ------ ----------
0   on     0
1   on     0
2   on     0
3   on     0
4   on     0
5   on     0
6   on     0
7   on     0
…

/*[buffer-internal, tc3 & tc4 configured. TC4 doesn’t have ECN]*/
```

```
Router#show controllers npu priority-flow-control location <loc>
Location Id:                          0/1/CPU0
PFC:                                  Enabled
PFC-Mode:                             buffer-internal
TC   Pause           Headroom        ECN
-------------------------------------------------------
3    86800 bytes     120000 bytes    76800 bytes
4    86800 bytes     120000 bytes    Not-configured

/*[buffer-extended PFC, tc3 & tc4 configured]*/

Router#show controllers npu priority-flow-control location <loc>
Location Id:                          0/1/CPU0
PFC:                                  Enabled
PFC-Mode:                             buffer-extended
TC   Pause
-----------
3    5000 us
4    10000 us

/*[No PFC]*/

Router#show controllers npu priority-flow-control location <loc>
Location Id:                          0/1/CPU0
PFC:                                  Disabled
```

Related Topics

- Priority Flow Control Overview, on page 1

Related Commands hw-module profile priority-flow-control location

View Priority Flow Control TX Frames Per Traffic Class

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
View Priority Flow Control TX Frames Per Traffic Class	Release 7.5.4	You can now view an estimation of the transmission (Tx) of Priority Flow Control (PFC) frames per traffic class, which informs you that the traffic flow has crossed configured PFC pause thresholds. This information allows you to rebalance traffic flows to ensure that network resources are efficiently used. Such verification is possible because we've added counters for PFC Tx frames. This functionality modifies the following to add the PFC Tx frame counters: • YANG data model (at Github under the 754 folder): Cisco-IOS-XR-ofa-npu-pfc-oper • **show controllers**

Beginning Cisco IOS XR Release 7.5.4, we have added an additional counter for PFC Tx pause frames per traffic class in the **show controllers priority-flow-control statistics** command form. With this counter, you receive timely information that the traffic flow has exceeded the configured PFC pause thresholds and hence that PFC Tx frames are sent out of the PFC-enabled interfaces sourcing such traffic on that network processing core. You can use this information to investigate and analyze why such traffic flows are causing congestion and improve traffic distribution across interfaces to utilize network resources efficiently.

Run the **show controllers priority-flow-control statistics** to view an estimation of the statistics for PFC Tx frames per traffic class (**Tx Frames**):

```
Router#show controllers hundredGigE 0/0/0/4  priority-flow-control statistics

Priority flow control information for interface HundredGigE0/0/0/4:
Priority Flow Control:
    Total Rx PFC Frames: 0
    Total Tx PFC Frames: 4832680
    Rx Data Frames Dropped: 1442056 (possible overflow)
    CoS  Status  Rx Frames   Tx Frames
    ---  ------  ----------  ----------
      0  on             0           0
      1  on             0           0
      2  on             0           0
      3  on             0     2416374
      4  on             0     2416306
      5  on             0           0
      6  on             0           0
      7  on             0           0
```

Guidelines and Limitations for Viewing Priority Flow Control TX Frames Per Traffic Class

This functionality is supported on:

- 8800-LC-36FH-M
- 8800-LC-36FH
- 88-LC0-34H14FH

Priority Flow Control Watchdog Overview

PFC Watchdog is a mechanism to identify any PFC storms (queue-stuck condition) in the network. It also prevents the PFC from propagating on the network and running in a loop. You can configure a PFC watchdog interval to detect whether packets in a no-drop queue are drained within a specified time period. When the time period is exceeded, all outgoing packets are dropped on interfaces that match the PFC queue that is not being drained.

This requires monitoring PFC receiving on each port and detecting ports seeing an unusual number of sustained pause frames. Once detected, the watchdog module can enforce several actions on such ports, which include generating a syslog message for network management systems, shutting down the queue, and autorestoring the queue (after the PFC storm stops).

Here's how the PFC Watchdog works:

1. The Watchdog module monitors the PFC-enabled queues to determine the reception of an unusual amount of PFC pause frames in a given interval (Watchdog interval.)
2. Your hardware notifies the Watchdog module when too many PFC frames are received and traffic on the corresponding queues is halted for a time interval.
3. On receiving such notifications, the Watchdog module starts the shutdown timer and moves the queue state to wait-to-shutdown state.
4. At regular intervals during the shutdown interval, the queue is checked for PFC frames and if the traffic in the queue is stuck. If the traffic isn't stuck because the queue didn't receive any PFC frames, the queue moves back to the monitored state.
5. If the traffic is stuck for a longer time and the shutdown-timer expires, the queue switches to a drop state and the PFC Watchdog begins to drop all packets.
6. At regular intervals, the Watchdog checks the queue for PFC frames and whether the traffic in the queue is still stuck. If traffic is stuck in the queue as PFC packets keep arriving, the queue remains in the drop or shutdown state.
7. When the traffic's no longer stuck, the autorestore timer starts. At regular intervals, the module checks if the queue is stuck because of PFC frames.
8. If the queue receives PFC frames during the last autorestore interval, the auto-restore timer is reset upon expiry.
9. If the queue receives no PFC frames during the last autorestore interval, the Watchdog module restores the queue, and traffic resumes.

Related Topics

- Priority Flow Control Overview, on page 1

Configure a Priority Flow Control Watchdog Interval

You can configure PFC Watchdog parameters (Watchdog interval, shutdown multiplier, auto-restore multiplier) at the global or interface levels. Note that:

- When global Watchdog mode is disabled or off, Watchdog is disabled on all interfaces. This condition is regardless of the interface level Watchdog mode settings.
- When global Watchdog mode is enabled or on, the interface level Watchdog mode configuration settings override the global Watchdog mode values.
- When you configure interface level Watchdog attributes such as interval, shutdown multiplier, and auto-restore multiplier, they override the global Watchdog attributes.

Note Configuring the PFC mode and its policies is a prerequisite for PFC Watchdog.

Configuration Example

You can configure the Watchdog at the global or at the interface level.

Note Watchdog is enabled by default, with system default values of:

Watchdog interval = 100 ms

Shutdown multiplier = 1

Auto-restore multiplier = 10

```
P/0/RP0/CPU0:ios#show controllers hundredGigE 0/2/0/0 priority-flow-control

Priority flow control information for interface HundredGigE0/2/0/0:

Priority flow control watchdog configuration:
(D) : Default value
U : Unconfigured
------------------------------------------------------------------------------
Configuration Item         Global  Interface  Effective
------------------------------------------------------------------------------
PFC watchdog state :        U        U        Enabled(D)
Poll interval :             U        U        100(D)
Shutdown multiplier :       U        U        1(D)
Auto-restore multiplier :   U        U        10(D)
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#priority-flow-control watchdog mode off
RP/0/RP0/CPU0:ios(config)#commit

RP/0/RP0/CPU0:ios(config)#do show controllers hundredGigE 0/2/0/0 priority-flo$
```

```
Priority flow control information for interface HundredGigE0/2/0/0:


Priority flow control watchdog configuration:
(D) : Default value
U : Unconfigured
------------------------------------------------------------------------------
Configuration Item          Global  Interface  Effective
------------------------------------------------------------------------------
PFC watchdog state :        Disabled  U        Disabled
Poll interval :              U        U        100(D)
Shutdown multiplier :        U        U        1(D)
Auto-restore multiplier :    U        U        10(D)

RP/0/RP0/CPU0:ios(config)#interface hundredGigE 0/2/0/0 priority-flow-control $
RP/0/RP0/CPU0:ios(config)#commit

RP/0/RP0/CPU0:ios(config)#do show controllers hundredGigE 0/2/0/0 priority-flo$

Priority flow control information for interface HundredGigE0/2/0/0:


Priority flow control watchdog configuration:
(D) : Default value
U : Unconfigured
------------------------------------------------------------------------------
Configuration Item             Global   Interface    Effective
------------------------------------------------------------------------------
PFC watchdog state :           Disabled  Enabled      Disabled
Poll interval :                  U         U          100(D)
Shutdown multiplier :            U         U          1(D)
Auto-restore multiplier :        U         U          10(D)
RP/0/RP0/CPU0:ios(config)#interface hundredGigE 0/2/0/1 priority-flow-control $
RP/0/RP0/CPU0:ios(config)#commit

RP/0/RP0/CPU0:ios(config)#do show controllers hundredGigE 0/2/0/1 priority-flo$

Priority flow control information for interface HundredGigE0/2/0/1:


Priority flow control watchdog configuration:
(D) : Default value
U: Unconfigured
------------------------------------------------------------------------------
Configuration Item               Global   Interface   Effective
------------------------------------------------------------------------------
PFC watchdog state :             Enabled  Disabled     Disabled
Poll interval :                    U        U          100(D)
Shutdown multiplier :              U        U          1(D)
Auto-restore multiplier :          U        U          10(D)
```

Verification

To verify that the PFC Watchdog is enabled globally, run the **sh run priority-flow-control watchdog mode** command.

```
Router#sh run priority-flow-control watchdog mode
priority-flow-control watchdog mode on
```

To verify your PFC Watchdog global configuration, run the **priority-flow-control watchdog** command.

```
Router#sh run priority-flow-control watchdog
priority-flow-control watchdog interval 100
priority-flow-control watchdog auto-restore-multiplier 2
```

```
priority-flow-control watchdog mode on
priority-flow-control watchdog shutdown-multiplier 2
```

Related Topics

- Priority Flow Control Watchdog Overview, on page 12

PFC Watchdog Statistics to Monitor Traffic Drops

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
PFC Watchdog Statistics to Monitor Traffic Drops	Release 7.5.4	We have enhanced the PFC Watchdog counters to ensure you get an accurate view of the packets dropped across queues. You now get separate statistic counters for the number of packets dropped and the total number of packets dropped. By providing deeper insights into Watchdog packet drops, we aim to help you plan your traffic flows better and avoid potential PFC storms. Earlier releases didn't support counters for packet drops. You can view the counters using: • YANG data model (at Github under the 753 folder): Cisco-IOS-XR-ofa-npu-pfc-oper • CLI: **show controllers priority-flow-control watchdog-stats** command form.

Here's how the PFC Watchdog works when it detects that a queue has stalled:

1. It begins dropping packets that arrive at the stalled queue. (See Priority Flow Control Watchdog Overview, on page 12 for details.)
2. After the congestion clears and no more PFC frames arrive at the queue, the Watchdog restores the queue and flushes all packets that are stuck in the Virtual Output Queue (VOQ) and the output queue.

This functionality with an enhanced statistics view provides two new counters for all the dropped packets.

PFC Watchdog Statistics for Two Counters

With Release 7.5.4, the PFC Watchdog statistics account for two counters:

- Dropped Packets = Cumulative VOQ drops + output queue drops in the most recent Watchdog shutdown event
- Total Dropped Packets = Cumulative VOQ drops + output drops in all the watchdog events from the time you ran the **clear controller priority-flow-control watchdog-stats** command to clear the statistics counter.

Thus, when you run the **show controllers priority-flow-control watchdog-stats** command, the output displays the statistics for dropped packets and total dropped packets.

PFC Watchdog Statistics Benefits

- With the additional insights the two counters provide, you can study traffic drop patterns across longer timelines.
- You can analyze traffic drops for every congestion recovery that a queue makes.
- With access to such deep analysis, you can plan your traffic flows better. You can adjust your traffic flow rates and prevent potential PFC storms based on previous drop patterns and trends.

View PFC Watchdog Statistics

```
Router#show  controllers hundredGigE 0/1/0/43 priority-flow-control watchdog-stats


Priority flow control information for interface HundredGigE0/1/0/43:


Priority flow control watchdog statistics:
SAR: Auto restore and shutdown
-------------------------------------------------------------------------------------------------
Traffic Class            :       0        1        2         3         4        5         6
        7
-------------------------------------------------------------------------------------------------
Watchdog Events          :       0        0        0         3         3        0         0
        0
Shutdown Events          :       0        0        0         3         3        0         0
        0
Auto Restore Events      :       0        0        0         3         3        0         0
        0
SAR Events               :       0        0        0      3510      3510        0         0
        0
SAR Instantaneous Events :       0        0        0      1172      1172        0         0
        0
Total Dropped Packets    :       0        0        0 941505767 941488166        0         0
        0
Dropped Packets          :       0        0        0 314855466 314887161        0         0
        0
```

Note Disregard the **SAR Events** and **SAR Instantaneous Events** entries because those numbers have no bearing on your operations.

Detect High Bandwidth Memory Congestion

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Detect High Bandwidth Memory Congestion	Release 7.5.3	We provide detailed insights into congestion on the High Bandwidth Memory (HBM), such as the devices on which congestion has occurred, the time stamps, and when the device returned to its normal state. With such details, you can investigate the cause of congestion and identify the source ports causing congestion for future preventive actions. You must configure PFC in the buffer-extended mode for this option. The feature introduces the following to enable the option to detect HBM congestion: • YANG data model (at Github under the 753 folder): Cisco-IOS-XR-um-8000-hw-module-profile-cfg • CLI: **hw-module profile npu memory buffer-extended location bandwidth-congestion-detection enable** It also introduces the following to view the congestion and memory usage details: • YANG data model (at Github under the 753 folder): Cisco-IOS-XR-8000-platforms-npu-memory-oper • CLI: **show controllers npu packet-memory**

Here's how congestion build-up in queues drives the usage of memory on your router:

1. Under normal queue conditions, packets egressing from an interface are enqueued into the Shared Memory System (SMS) or internal buffer memory.
2. When congestion occurs, the SMS remains the buffer for packets until the congestion exceeds the per-queue usage criteria. This usage criterion is a combination of the threshold buffer space per VOQ and the age of the packet (defined in units of milliseconds).

3. At this stage, the packets are evicted to the High Bandwidth Memory (HBM) or external buffer memory. This provides additional buffer memory to absorb short bursts of congestion.
4. However, when packets from multiple VOQs are evicted to the HBM, the HBM bandwidth begins to experience congestion. The HBM congestion causes data loss for lossless traffic classes.

It's for insights into these extreme events that we've provided the option to detect congestion in the HBM. This information is meant for post-event analysis and reporting. You can enable this functionality by configuring the command **hw-module profile npu memory buffer-extended bandwidth-congestion-detection enable**. By doing so, you can view information such as:

- Devices on which congestion has occurred and the time stamps
- The current buffer memory usage and the highest memory usage watermark reached since the last reading

Benefits of Detecting High Bandwidth Memory Congestion

When you enable the ability to detect HBM congestion, the **show controllers npu packet-memory** command displays:

- Details of congestion along with the affected devices.
- A snapshot of the maximum memory usage with time stamps.

Using this information, you can conduct post-event analysis such as:

- investigate the cause of the congestion.
- identify the ports and flows that are the source of this congestion.

High Bandwidth Memory Congestion Detection: Guidelines and Limitations

- You must configure PFC in the buffer-extended mode for this functionality.
- There's no requirement for a line card reload after you configure **hw-module profile npu memory buffer-extended bandwidth-congestion-detection enable**.
- This functionality is supported on:
  - Cisco Silicon One Q200-based routers and line cards
  - 8201-32FH routers
  - 88-LC0-48TH-MO line cards
  - 88-LC0-36FH line cards
  - 88-LC0-36FH-M line cards

Configure High Bandwidth Memory Congestion Detection

To enable the detection of HBM congestion, configure the **hw-module profile npu memory buffer-extended bandwidth-congestion-detection enable** command.

```
Router#config
Router(config)#hw-module profile npu buffer-extended location 0/6/CPU0
bandwidth-congestion-detection enable
Router(config)#commit
Router(config)#exit
```

Note There's no requirement to reload the line card after the configuration.

Verification

The following table lists the various verification commands you can run depending on your required information. The outputs for these commands are available after the table.

Table 6: Verification Commands

Type of Information	Commands	Details
Buffer usage	• **show controllers npu packet-memory usage instance all location all** • **show controllers npu packet-memory usage verbose instance all location all**	• These commands provide data for the current use and the highest watermark reached since the last reading for both SMS and HBM. • The refresh interval for the information is 30 seconds.
HBM bandwidth congestion	• **show controllers npu packet-memory congestion instance all location all** • **show controllers npu packet-memory congestion detail instance all location all** • **show controllers npu packet-memory congestion verbose instance all location all**	• These commands provide data for when the HBM congestion occurred or is about to happen. • The output maintains a history of the last 120 events regardless of the elapsed time. • The refresh interval for new events to be added is 30 seconds.

Run the **show controllers npu packet-memory usage instance all location all** command to view the following details:

- the timestamp at which data is sampled
- the network processor name (**Device**)
- packet memory usage for that timestamp for SMS (**Buff-int Usage** in units of buffers) and HBM (**Buff-ext Usage** in units of 8 KB blocks)
- highest maximum watermark reached for SMS (**Buff-int Max WM**) and HBM (**Buff-ext Max WM**) since the last reading

```
Router#show controllers npu packet-memory usage instance all location all
HW memory Information For Location: 0/6/CPU0

-------------------------------------------------------------------------------------
Timestamp(msec)                   | Device | Buff-int | Buff-int | Buff-ext | Buff-ext
                                            Usage      Max WM   | Usage    | Max WM
-------------------------------------------------------------------------------------
Wed 2022-09-21 01:54:11.154 UTC       0          7          8          0          0
Wed 2022-09-21 01:54:12.154 UTC       0          7          8          0          0
Wed 2022-09-21 01:54:24.023 UTC       1         22         22          0          0
Wed 2022-09-21 01:54:34.088 UTC       2         11         12          0          0
Wed 2022-09-21 01:54:35.088 UTC       2         11         12          0          0
Wed 2022-09-21 01:54:36.088 UTC       2         11         12          0          0
Wed 2022-09-21 01:54:37.088 UTC       2         11         12          0          0
Wed 2022-09-21 01:54:38.089 UTC       2         11         12          0          0
Wed 2022-09-21 01:54:39.089 UTC       2         11         12          0          0
Wed 2022-09-21 01:54:40.089 UTC       2         11         12          0          0
```

In addition, the **show controllers npu packet-memory usage verbose instance all location all** command displays timestamp in milliseconds:

```
RP/0/RP0/CPU0:Router#show controllers npu packet-memory usage verbose instance all location
 all

HW memory Information For Location: 0/RP0/CPU0

* Option 'verbose' formatted data is for internal consumption.
-----------------------------------------------------------------------
Timestamp(msec) | Device | Buff-int | Buff-int | Buff-ext | Buff-ext
                           Usage      Max WM   | Usage    | Max WM
-----------------------------------------------------------------------
   1663958881006       0       2455       2676        637        640
   1663958882007       0       2461       2703        635        640
   1663958883007       0       2364       2690        635        640
   1663958884007       0      71603      75325       3183      18336
   1663958885008       0       2458       2852       1275       1279
   1663958886008       0       2484       2827       1275       1279
```

Run the **show controllers npu packet-memory congestion instance all location all** command to view if HBM congestion has occurred and the timestamp of the congestion state.

- Each row indicates:
  - the timestamp at which data is sampled
  - the network processor name (**Device**)
  - the event type (**Normal** or **Congest**)
- The data displayed is for the last 120 events, and new events get added every 30 seconds. To view the updated data, re-run the command.
- After 120 events, the latest entry replaces the oldest entry. You can't clear the events from the list.

```
Router#show controllers npu packet-memory congestion instance all location all
HW memory Information For Location: 0/6/CPU0

-------------------------------------------------------------
Timestamp(msec)                  |      Buff-ext       | Device
                                 |      Event Type     |
-------------------------------------------------------------
Wed 2022-09-21 02:14:41.709 UTC            Congest         1
```

```
Wed 2022-09-21 02:14:41.959 UTC            Congest         1
Wed 2022-09-21 02:14:42.960 UTC            Congest         1
Wed 2022-09-21 02:14:43.960 UTC            Congest         1
Wed 2022-09-21 02:14:45.210 UTC            Congest         1
Wed 2022-09-21 02:14:45.710 UTC            Congest         1
Wed 2022-09-21 02:14:47.711 UTC            Normal          1
```

Run the **show controllers npu packet-memory congestion detail instance all location all** command to view the following details:

- the timestamp at which data is sampled
- the event type (**Normal** or **Congest**)
- the network processor name (**Device**) and the slice number (**Slice**) for that device. Every network processor has a fixed number of slices, and each slice, in turn, has a set number of ports.
- single VOQ buffer and aggregated SMS VOQ buffers
- packet memory usage for that timestamp for SMS (**Buff-int Usage** in units of buffers) and HBM (**Buff-ext Usage** in units of 8 KB blocks)
- highest maximum watermark reached for SMS (**Buff-int Max WM**) and HBM (**Buff-ext Max WM**) since the last reading

```
RRP/0/RP0/CPU0:ios#show controllers npu packet-memory congestion detail instance all location
 all
Fri Sep 23 18:49:50.640 UTC
HW memory Information For Location: 0/RP0/CPU0

* Option 'detail' formatted data is for internal consumption.
```

Timestamp(msec)	Buff-ext	Device	Slice	VOQ	VOQ-buff	Evicted-buff	Buff-int	Buff-int	Buff-int	Buff-ext	Buff-ext						
	Event Type																int-WM
	int-WM	UC-WM	Usage	Max WM	Usage	Max WM											
Fri 2022-09-23 18:42:30.349 UTC				Congest			0		5		534		16011				
	63969	65451	70410	70410	34405	34405											
Fri 2022-09-23 18:42:31.101 UTC				Normal			0		5		534		0				
	0	900	2440	2440	0	0											
Fri 2022-09-23 18:42:37.354 UTC				Congest			0		5		534		16011				
	63984	65493	70573	70573	34408	34408											
Fri 2022-09-23 18:42:38.354 UTC				Normal			0		5		534		0				
	0	915	2455	2455	0	0											
Fri 2022-09-23 18:42:44.606 UTC				Congest			0		5		534		16011				
	64002	65520	70081	70081	34532	34532											

… … …

Run the **show controllers npu packet-memory congestion verbose instance all location all** command to view the following details:

- timestamp in milliseconds when the data is sampled
- the network processor name (**Device**) and the slice number (**Slice**) for that device. Every network processor has a fixed number of slices, and each slice, in turn, has a set number of ports
- event type, where 0 = single VOQ-based congestion and 1 = single VOQ-based congestion backoff (**VOQ-buff int-WM**), 2 = congestion in aggregated SMS buffers for VOQ and 3 = congestion backoff in aggregated SMS buffers for VOQ (**Evicted-buff int-WM**)
- buffer internal for unicast, which is for information only.

- packet memory usage for that timestamp for SMS (**Buff-int Usage** in units of buffers) and HBM (**Buff-ext Usage** in units of 8 KB blocks)
- highest maximum watermark reached for SMS (**Buff-int Max WM**) and HBM (**Buff-ext Max WM**) since the last reading

```
Router#show controllers npu packet-memory congestion verbose instance all location all
HW memory Information For Location: 0/RP0/CPU0
* Option 'verbose' formatted data is for internal consumption.
```

Timestamp(msec)	Event	Device	Slice	VOQ	VOQ-buff int-WM	Evicted-buff int-WM	Buff-int UC-WM	Buff-int Usage	Buff-int Max WM	Buff-ext Type	Buff-ext Usage	Buff-ext Max WM
1663958550349	0	0	5	534	16011	63969	65451	70410	70410	34405	34405	
1663958551101	1	0	5	534	0	0	900	2440	2440	0	0	
1663958557354	0	0	5	534	16011	63984	65493	70573	70573	34408	34408	
1663958558354	1	0	5	534	0	0	915	2455	2455	0	0	
1663958564606	0	0	5	534	16011	64002	65520	70081	70081	34532	34532	
1663958565356	1	0	5	534	0	0	915	2417	2417	0	0	

Global Pause Frames for High Bandwidth Memory Congestion

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Global Pause Frames for High Bandwidth Memory Congestion	Release 7.5.4	We ensure no packet drops on PFC-enabled queues due to High Bandwidth Memory (HBM) congestion. Such prevention of drops is possible because we have enabled the triggering of global pause frames (X-Off) whenever there's HBM congestion. This functionality is disabled by default. You have the following options to enable it: • **CLI**: **hw-module profile npu memory buffer-extended bandwidth-congestion-protect enable** • **YANG Data Model**: **Cisco-IOS-XR-um-8000-hw-module-profile-cfg** (see GitHub, YANG Data Models Navigator) This feature introduces the **show hw-module bandwidth-congestion-protect** command to view the status of the global X-Off configuration.

When High Bandwidth Memory congestion occurs (see Detect High Bandwidth Memory Congestion, on page 17 for details), global pause frames (X-Off) are triggered for all PFC-enabled queues, regardless of whether those queues are the aggressor queues, hence the name 'global.' The only queues that don't transmit the X-Off trigger are those that don't receive any traffic. Such action ensures no packet drops on lossless queues, allowing you to meet your traffic bandwidth commitments for specific customers and requirements. It also ensures that the X-Off isn't triggered prematurely, which would have affected uncongested queues as well, causing a drop in performance.

To enable this functionality, configure the **hw-module profile npu memory buffer-extended bandwidth-congestion-protect enable** command.

Global Pause Frames for High Bandwidth Memory Congestion: Guidelines and Limitations

- This functionality isn't supported for the buffer-extended mode where the devices are more than 0.5 km apart.
- Configuring the **hw-module profile npu memory buffer-extended bandwidth-congestion-protect enable** command for line cards where you've configured headroom values exceeding 6144000 bytes could result in a commit error or the feature not being enabled.
- You must reload the line card for the **hw-module profile npu memory buffer-extended bandwidth-congestion-protect enable** command to take effect.
- This functionality is supported on:
  - 88-LC0-36FH line cards
  - 88-LC0-36FH-M line cards

Configure Global Pause Frames for High Bandwidth Memory Congestion

To enable triggering of global pause frames (X-off) whenever there's HBM congestion in the buffer-extended mode, configure the **hw-module profile npu memory buffer-extended bandwidth-congestion-protect enable** command.

```
Router#config
Router(config)#hw-module profile npu buffer-extended location 0/1/CPU0
bandwidth-congestion-protect enable
Router(config)#commit
```

Verification

Run the **show hw-module bandwidth-congestion-protect** command to view details about global X-off.

```
RP/0/RP1/CPU0:router#show hw-module bandwidth-congestion-protect location 0/1/CPU0

Location   Configured     Applied          Action
--------------------------------------------------------------------------------

0/1/CPU0     Yes             No               Reload
```

The table lists the various possibilities for the command output based on your activity.

If you...	Configured field displays…	Applied field displays…	Action field displays…
Configure the **hw-module profile npu memory buffer-extended** command	**Yes**	**No**	**Reload**
Use the **no** form of the **hw-module profile npu memory buffer-extended** command after configuring it, but before reloading the line card	**No**	**No**	**N/A**
Configure the **hw-module profile npu memory buffer-extended** command for a supported variant and reload the line card	**Yes**	**Yes, Active** **Note** **Yes** indicates that the configuration is programmed to the hardware, **Active** indicates that the global X-off functionality is active on the hardware.	**N/A**
Use the **no** form of the **hw-module profile npu memory buffer-extended** command when it is active, and commit the **no** form but don't reload the line card	**No** **Note** At this stage, the output displays the user action and not the hardware status.	**No** **Note** At this stage, the output displays the user action and not the hardware status.	**Reload**
Reload the line card after committing the **no** form of the **hw-module profile npu memory buffer-extended** command	**No** **Note** At this stage, the output displays the hardware status.	**No** **Note** At this stage, the output displays the hardware status.	**N/A**