Priority Flow Control Overview

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Priority Flow Control	Release 7.3.2	Previously available in Release 6.6.3, this feature is a link-level flow control mechanism that enables you to selectively pause traffic based on its class of service (CoS).

Priority flow control (PFC; 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 transient congestion. PFC is similar to 802.3x Flow Control (pause frames) or link-level flow control (LLFC). However, instead of pausing all traffic on a link, PFC functions on a per class-of-service (CoS) basis.

During congestion, PFC sends a pause frame that indicates which CoS value must be paused. When the congestion is mitigated, the router stops sending the PFC frames to the upstream node.

For details about the Priority Flow Control feature, see the *Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers*.

Restrictions

The following restrictions apply while configuring PFC thresholds, PFC watchdog and QoS policies in PFC profile.

- The PFC feature is only supported in the non-HQoS profile.
- The PFC feature is only supported on the following line card or fixed chassis PIDs of the NCS5500 Series:
  - NC55-36X100G

  - NC55-18H18F
  - NC55-24X100G-SE
  - NC55-36X100G-S
  - NC55-24H12F-SE
  - NC55-36X100G-A-SE
  - NCS-55A1-36H-SE-S
  - NCS-55A1-36H-S
  - NCS-55A1-24H
  - NCS-55A1-48Q6H

- When PFC is enabled, only two parameter scheduling is supported. So, the egress actions could have either priority or weighted fair queue (WFQ) scheduling apart from a shaper action. For WFQ scheduling, either bandwidth remaining ratio (BRR) or bandwidth is supported and a mix of the two in the same policy-map is rejected. With PFC, BRR has a weight range of 1 to 256 as against 1 to 4096 for the non PFC, non-HQoS case.
- The hw-module profile for enabling PFC per queue and defining PFC Tx thresholds are global configurations per line card location. So, only one set of pause, resume thresholds and headroom sizes per traffic-class can be configured for all ports on a given line card location. With the introduction of this new configuration model, the pause action under an egress policy class-map is deprecated.
- While a line card reload is required for any addition or removal of traffic classes in the hw-module profile, you can update the pause, resume thresholds and buffer / headroom sizes on already configured traffic-classes without requiring a line card reload.
- PFC watchdog (PFCWD) interval has a minimum granularity of 100ms. When you configure higher values, they are rounded up to the nearest multiple of 100ms.
- There is no XML schema or yang model for the new hw-module profile. Only CLI configuration and show commands are available.
- In the PFC profile, only monotonically increasing priority levels with traffic-class are supported. For example, TC7 should be p1, TC6 should be p2, and so on, until TC1 is p7 with class-default being lowest priority. Rate classes can be mixed anywhere.
- When an egress queue is disabled as part of a watchdog queue shutdown action and once the pause frames stop, the queue drains the enqueued packets on the wire while it waits to be restored back.
- If PFCWD is disabled and a traffic-class experiences sustained PFC storm beyond 7s, then the traffic on that queue continues to be dropped even when the PFC storm stops. Traffic needs to be stopped and restarted or all the incoming ports involved in sending traffic to this egress port needs to be flapped (shut and no shut) to manually recover the egress port.
- The class-map actions of queue-limit and WRED thresholds without ECN enabled have no impact on PFC-enabled queues. Queue-limit is redundant because packet is buffered based on pause or x-off and headroom configured per source port.
- Even when PFCWD shuts down an egress queue experiencing storm, if the ingress traffic was already causing a congestion due to the application of sub-rate shapers or BRR policy, then the ingress buffers build up and PFC Tx is still generated towards the sender. In other words, PFCWD only removes the

source of a storm. It does not suppress PFC Tx generated from sustained congestion due to user configuration.

- In the PFC profile, there is a 25% reduction in available buffer descriptors. Further, 20% of the remaining buffer space is reserved for headroom. This leads to only 80% of the 75% buffers being available in the shared pool for all ingress VoQs on that NP core.

Configuring Priority Flow Control Thresholds

You can set values for pause threshold (x-off), resume threshold (x-on) and headroom for a traffic class on all PFC enabled ports on a given line card location using the hw-module profile priority-flow-control command.

The existing queue-limit for that traffic class on an egress queuing policy on that line card will have no impact and the effective queue limit is pause threshold + headroom.

Note For optimal functionality in hardware, for a given traffic class, the resume threshold should not be more than 10% of the pause threshold and the headroom should be at least 100KB.

For details on how to configure PFC on an interface, see the *Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers*.

Configuration Example

Apart from the line card level PFC parameters configured through the hw-module profile, PFC Receive functionality requires an egress policy to be applied and egress traffic to be mapped to unique queues per CoS value on all PFC enabled interfaces.

You can accomplish this with the following sequence of configuration.

1. Configure ingress classification policy to map incoming traffic to the right CoS / priority queue.
2. Configure egress queuing policy with the relevant shaping, priority and weighted fair queue scheduling actions
3. Attach the ingress policy on all interfaces where traffic can come in.
4. Attach the egress policy on all PFC enabled interfaces in the system.
5. Configure hw-module priority-flow-control profile and configure pause, resume and headroom for all PFC traffic-classes on all line card locations requiring PFC feature.

Reload the line card for all traffic-class addition and deletions. Parameters within an already configured traffic-class can be edited 'in place' without requiring a line card reload.

```
Hw-module configuration:
====================
RP/0/RP1/CPU0:NCS5504(config)#hw-module profile priority-flow-control location 0/0/CPU0
tRP/0/RP1/CPU0:NCS5504(config-pfc-loc)# traffic-class 3 pause-threshold 403200 bytes
resume-threshold 40320 bytes headroom 441600 bytes
RP/0/RP1/CPU0:NCS5504(config-pfc-loc)# traffic-class 4 pause-threshold 403200 bytes
resume-threshold 40320 bytes headroom 441600 bytes
RP/0/RP1/CPU0:NCS5504(config-pfc-loc)#

Class-map configuration:
```

```
==================
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any MAIN_IN_CMAP_1
match prRP/0/RP1/CPU0:NCS5504(config-cmap)# match precedence 1
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any MAIN_IN_CMAP_2
RP/0/RP1/CPU0:NCS5504(config-cmap)# match precedence 2
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any MAIN_IN_CMAP_3
RP/0/RP1/CPU0:NCS5504(config-cmap)# match precedence 3
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any MAIN_IN_CMAP_4
RP/0/RP1/CPU0:NCS5504(config-cmap)# match precedence 4
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any MAIN_IN_CMAP_5
RP/0/RP1/CPU0:NCS5504(config-cmap)# match precedence 5
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any MAIN_IN_CMAP_6
RP/0/RP1/CPU0:NCS5504(config-cmap)# match precedence 6
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any MAIN_IN_CMAP_7
RP/0/RP1/CPU0:NCS5504(config-cmap)# match precedence 7
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any PFC_OUT_CMAP_1
RP/0/RP1/CPU0:NCS5504(config-cmap)# match traffic-class 1
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any PFC_OUT_CMAP_2
RP/0/RP1/CPU0:NCS5504(config-cmap)# match traffic-class 2
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any PFC_OUT_CMAP_3
RP/0/RP1/CPU0:NCS5504(config-cmap)# match traffic-class 3
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any PFC_OUT_CMAP_4
RP/0/RP1/CPU0:NCS5504(config-cmap)# match traffic-class 4
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any PFC_OUT_CMAP_5
RP/0/RP1/CPU0:NCS5504(config-cmap)# match traffic-class 5
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any PFC_OUT_CMAP_6
RP/0/RP1/CPU0:NCS5504(config-cmap)# match traffic-class 6
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#class-map match-any PFC_OUT_CMAP_7
RP/0/RP1/CPU0:NCS5504(config-cmap)# match traffic-class 7
RP/0/RP1/CPU0:NCS5504(config-cmap)# end-class-map
RP/0/RP1/CPU0:NCS5504(config)#!
RP/0/RP1/CPU0:NCS5504(config)#

Ingress Policy-map configuration:
=========================
RP/0/RP1/CPU0:NCS5504(config)#policy-map MAIN_OUT_TC_SUPPORTING_UUT
RP/0/RP1/CPU0:NCS5504(config-pmap)# class MAIN_IN_CMAP_1
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# set traffic-class 1
```

```
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class MAIN_IN_CMAP_2
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# set traffic-class 2
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class MAIN_IN_CMAP_3
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# set traffic-class 3
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class MAIN_IN_CMAP_4
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# set traffic-class 4
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class MAIN_IN_CMAP_5
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# set traffic-class 5
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class MAIN_IN_CMAP_6
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# set traffic-class 6
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class MAIN_IN_CMAP_7
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# set traffic-class 7
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)#
RP/0/RP1/CPU0:NCS5504(config-pmap-c)#


Egress policy-map configuration:
============================
RP/0/RP1/CPU0:NCS5504(config)#policy-map PFC_UUT_bwrr_microsoft
asRP/0/RP1/CPU0:NCS5504(config-pmap)# class PFC_OUT_CMAP_1
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# bandwidth remaining ratio 5
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# queue-limit 192 us
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class PFC_OUT_CMAP_2
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# bandwidth remaining ratio 5
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# queue-limit 192 us
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class PFC_OUT_CMAP_3
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# bandwidth remaining ratio 20
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# random-detect ecn
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# random-detect 224 kbytes 275 kbytes
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class PFC_OUT_CMAP_4
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# bandwidth remaining ratio 20
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# random-detect ecn
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# random-detect 224 kbytes 275 kbytes
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class PFC_OUT_CMAP_5
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# bandwidth remaining ratio 20
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# queue-limit 192 us
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class PFC_OUT_CMAP_6
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# bandwidth remaining ratio 1
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# queue-limit 192 us
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class PFC_OUT_CMAP_7
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# shape average percent 10
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# priority level 1
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# queue-limit 192 us
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# class class-default
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# bandwidth remaining ratio 20
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# queue-limit 192 us
RP/0/RP1/CPU0:NCS5504(config-pmap-c)# !

Applying policy to ingress and egress interface:
========================================
```

```
RP/0/RP1/CPU0:NCS5504#show running-config interface hundredGigE 0/0/0/0
interface HundredGigE0/0/0/0
 service-policy input MAIN_OUT_TC_SUPPORTING_UUT
 ipv4 address 100.1.9.1 255.255.255.0
 ipv6 address 100:1:9::1/96
 priority-flow-control mode on

interface HundredGigE0/1/0/19
 service-policy output PFC_ECN_UUT_bwrr_microsoft
 ipv4 address 10.1.9.1 255.255.255.0
 ipv6 address 10:1:9::1/96
 priority-flow-control mode on
!
```

Running Configuration

```
RP/0/RP1/CPU0:NCS5504# show running-config hw-module profile priority-flow-control location
 0/0/CPU0
Tue Oct 13 10:31:58.035 UTC
hw-module profile priority-flow-control location 0/0/CPU0
traffic-class 3 pause-threshold 403200 bytes resume-threshold 40320 bytes headroom 441600
bytes
traffic-class 4 pause-threshold 403200 bytes resume-threshold 40320 bytes headroom 441600
bytes
!

RP/0/RP1/CPU0:NCS5504#show running-config class-map
Tue Oct 13 10:32:02.400 UTC
class-map match-any MAIN_IN_CMAP_1
match precedence 1
end-class-map
!
class-map match-any MAIN_IN_CMAP_2
match precedence 2
end-class-map
!
class-map match-any MAIN_IN_CMAP_3
match precedence 3
end-class-map
!
class-map match-any MAIN_IN_CMAP_4
match precedence 4
end-class-map
!
class-map match-any MAIN_IN_CMAP_5
match precedence 5
end-class-map
!
class-map match-any MAIN_IN_CMAP_6
match precedence 6
end-class-map
!
class-map match-any MAIN_IN_CMAP_7
match precedence 7
end-class-map
!
class-map match-any PFC_OUT_CMAP_1
match traffic-class 1
end-class-map
!
class-map match-any PFC_OUT_CMAP_2
match traffic-class 2
end-class-map
!
```

```
class-map match-any PFC_OUT_CMAP_3
match traffic-class 3
end-class-map
!
class-map match-any PFC_OUT_CMAP_4
match traffic-class 4
end-class-map
!
class-map match-any PFC_OUT_CMAP_5
match traffic-class 5
end-class-map
!
class-map match-any PFC_OUT_CMAP_6
match traffic-class 6
end-class-map
!
class-map match-any PFC_OUT_CMAP_7
match traffic-class 7
end-class-map
!

RP/0/RP1/CPU0:NCS5504#show running-config policy-map PFC_UUT_bwrr_microsoft
Tue Oct 13 10:32:18.009 UTC
policy-map PFC_UUT_bwrr_microsoft
class PFC_OUT_CMAP_1
bandwidth remaining ratio 5
queue-limit 192 us
!
class PFC_OUT_CMAP_2
bandwidth remaining ratio 5
queue-limit 192 us
!
class PFC_OUT_CMAP_3
bandwidth remaining ratio 20
random-detect ecn
random-detect 224 kbytes 275 kbytes
!
class PFC_OUT_CMAP_4
bandwidth remaining ratio 20
random-detect ecn
random-detect 224 kbytes 275 kbytes
!
class PFC_OUT_CMAP_5
bandwidth remaining ratio 20
queue-limit 192 us
!
class PFC_OUT_CMAP_6
bandwidth remaining ratio 1
queue-limit 192 us
!
class PFC_OUT_CMAP_7
shape average percent 10
priority level 1
queue-limit 192 us
!
class class-default
bandwidth remaining ratio 20
queue-limit 192 us
!
end-policy-map
!

RP/0/RP1/CPU0:NCS5504#show running-config policy-map MAIN_OUT_TC_SUPPORTING_UUT
Tue Oct 13 10:32:31.430 UTC
```

```
policy-map MAIN_OUT_TC_SUPPORTING_UUT
class MAIN_IN_CMAP_1
set traffic-class 1
!
class MAIN_IN_CMAP_2
set traffic-class 2
!
class MAIN_IN_CMAP_3
set traffic-class 3
!
class MAIN_IN_CMAP_4
set traffic-class 4
!
class MAIN_IN_CMAP_5
set traffic-class 5
!
class MAIN_IN_CMAP_6
set traffic-class 6
!
class MAIN_IN_CMAP_7
set traffic-class 7
!
class class-default
!
end-policy-map
!
```

Verification

Incoming PFC frames are displayed on the PFC Rx statistics that are shown in the command below. If the interface is enabled for PFC and has an egress queuing policy applied to it, then the queue corresponding to the incoming CoS value on the PFC frames is paused

When PFC pause thresholds are crossed, PFC Tx frames are sent out of the PFC enabled interfaces sourcing such traffic on that network processing core. This can be verified against the PFC Tx statistics that are shown on this command. After the congestion condition ceases and the buffer resource usage goes below the PFC resume thresholds, a PFC resume frame is sent and the PFC Tx frames are stopped.

```
RP/0/RP1/CPU0:NCS5504#show controllers hundredGigE 0/0/0/0  priority-flow-control statistics

Mon Oct 12 12:22:39.362 UTC

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


Priority Flow Control:
    Total Rx PFC Frames: 0
    Total Tx PFC Frames: 1764273
    Rx Data Frames Dropped: 0
    CoS  Status  Rx Frames   Tx Frames
    ---  ------  ----------  ----------
      0  off              0           0
      1  off              0           0
      2  off              0           0
      3  on               0      882032
      4  on               0      882241
      5  off              0           0
      6  off              0           0
      7  off              0           0
```

The status of the traffic-class and the configured parameters under it on the hw-module priority-flow-control profile can be checked using the following command.

```
RP/0/RP1/CPU0:NCS5504#show controllers npu priority-flow-control loc 0/0/CPU0
Mon Oct 12 14:35:17.531 UTC

Location:    0/0/CPU0
PFC:          Enabled
TC    Pause-threshold    Resume-Threshold     Headroom
-------------------------------------------------------
3       403200 bytes        40320 bytes        441600 bytes
4       403200 bytes        40320 bytes        441600 bytes
```

Priority Flow Control Watchdog Overview

PFC Watchdog is a mechanism to identify any PFC storms (queue-stuck condition) in the network, and to prevent 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 being 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 amount 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 auto-restoring the queue (after the PFC storm stops).

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.

Note PFC Watchdog also monitors, detects, and generates a syslog message every 5 minutes if global pause frames or link level flow control frames are received on a PFC-enabled port.

Such frames are ignored and discarded on the PFC-enabled port.

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-restart multiplier = 10

```
RP/0/RP1/CPU0:NCS5504# show controllers hundredGigE 0/0/0/0 priority-flow-control
watchdog-config
Mon Oct 12 14:32:47.056 UTC

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


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/RP1/CPU0:NCS5504# show controllers hundredGigE 0/0/0/0 priority-flow-control
watchdog-stat
watchdog-state  watchdog-stats
RP/0/RP1/CPU0:NCS5504# show controllers hundredGigE 0/0/0/0 priority-flow-control
watchdog-state
Mon Oct 12 14:32:56.760 UTC

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


Priority flow control watchdog state machine state:
D - Disabled
M - Monitoring
S - Waiting For Shutdown
R - Waiting to Restore
------------------------------------------------------------
PFC Watchdog      : Enabled
Watchdog SM state :  Traffic Class
                     7 6 5 4 3 2 1 0
                     - - - D D - - -
RP/0/RP1/CPU0:NCS5504# show controllers hundredGigE 0/0/0/0 priority-flow-control
watchdog-stats
Mon Oct 12 14:33:09.321 UTC
```

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


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        0        0        0        0
      0
Shutdown Events          :       0        0        0        0        0        0        0
      0
Auto Restore Events      :       0        0        0        0        0        0        0
      0
SAR Events               :       0        0        0        0        0        0        0
      0
SAR Instantaneous Events :       0        0        0        0        0        0        0
      0
Total Dropped Packets    :       0        0        0        0        0        0        0
      0
Dropped Packets          :       0        0        0        0        0        0        0
      0
RP/0/RP1/CPU0:NCS5504#

RP/0/RP1/CPU0:NCS5504#show controllers npu priority-flow-control loc 0/0/CPU0
Mon Oct 12 14:35:17.531 UTC

Location:    0/0/CPU0
PFC:          Enabled
TC    Pause-threshold    Resume-Threshold     Headroom
-------------------------------------------------------
3       403200 bytes        40320 bytes        441600 bytes
4       403200 bytes        40320 bytes        441600 bytes
RP/0/RP1/CPU0:NCS5504#

RP/0/RP1/CPU0:NCS5504#clear controller hundredGigE 0/0/0/0 priority-flow-control
watchdog-stats ?
  traffic-class  Traffic class to be cleared
  <cr>
RP/0/RP1/CPU0:NCS5504#clear controller hundredGigE 0/0/0/0 priority-flow-control
watchdog-stats
Mon Oct 12 14:36:12.407 UTC
RP/0/RP1/CPU0:NCS5504#clear controller hundredGigE 0/0/0/0 priority-flow-control
watchdog-stats traffic-class ?
  <0-7>  Traffic class
RP/0/RP1/CPU0:NCS5504#clear controller hundredGigE 0/0/0/0 priority-flow-control
watchdog-stats traffic-class 3
NOTE: Use the clear commands to clear the statistics displayed by the show commands.
```

To restore a queue manually that has been shut down by the PFC watchdog action, run the command **set controller <> priority-flow-control recover traffic-class [0-7]**. Completion of this operation resets all internal watchdog state machines and the queue is back to a monitoring state.

```
RP/0/RP1/CPU0:NCS5504#show controllers hundredGigE 0/1/0/19 priority-flow-control
watchdog-state

Priority flow control information for interface HundredGigE0/1/0/19:


Priority flow control watchdog state machine state:
D - Disabled
M - Monitoring
```

```
S - Waiting For Shutdown
R - Waiting to Restore
------------------------------------------------------------
PFC Watchdog : Enabled
Watchdog SM state : Traffic Class
7 6 5 4 3 2 1 0
- - - M R - - -
RP/0/RP1/CPU0:NCS5504#

RP/0/RP1/CPU0:NCS5504#set controller hundredGigE 0/1/0/19 priority-flow-control recover
traffic-class 3

RP/0/RP1/CPU0:NCS5504#show controllers hundredGigE 0/1/0/19 priority-flow-control
watchdog-state

Priority flow control information for interface HundredGigE0/1/0/19:


Priority flow control watchdog state machine state:
D - Disabled
M - Monitoring
S - Waiting For Shutdown
R - Waiting to Restore
------------------------------------------------------------
PFC Watchdog : Enabled
Watchdog SM state : Traffic Class
7 6 5 4 3 2 1 0
- - - M M - - -
```

Related Topics

- Priority Flow Control Overview, on page 1

Monitoring and Logging Packet Drops on Lossless PFC-Enabled Queues

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Monitoring and Logging Packet Drops on Lossless PFC-Enabled Queues	Release 7.3.2	In case of packet drops on lossless PFC-enabled queues that also have the PFC Watchdog feature enabled, this functionality generates syslogs in 10-minute intervals. Such timely alerts help you troubleshoot quickly, isolate issues, and reroute traffic if necessary, with minimal impact on end-user services.

A Brief Background

Lossless PFC-enabled queues avoid queue tail drops during temporary congestion by sending PFC pause frames back to the previous network element to transmit lesser packets. Depending on the duration of the congestion, the previous network element may send PFC pause frames further upstream, hence triggering subsequent network elements to pause all the way up to the sender. Suppose that the congestion lasts longer or is persistent. In that case, the buffers on some of these network elements may still overflow, causing some packet drops eventually. PFC watchdog isolates such cases of persistent congestion and excessive PFC pause frames. PFC watchdog (enabled by default on all PFC enabled queues) monitors neighboring network elements that send excessive PFC pause frames and shuts down such queues until the PFC storm subsides. This action isolates the issue to just those queues, stops new PFC pause frame generation, and frees up the device buffers for other traffic flows, thus preventing the PFC storm from propagating throughout the network and affecting all other traffic flows.

From IOS XR Release 7.3.2, the PFC watchdog also monitors and periodically reports packet drops that occur due to excessive pausing, congestion, or PFC watchdog-triggered queue shutdown. While the **show controller** commands continue to provide an on-demand display of the total number of packets dropped for a given PFC enabled queue, the new logging mechanism provides timely alerts for proactive user intervention when drops occur on lossless PFC-enabled queues.

Highlights of Logging and Monitoring Packet Drops

From Release 7.3.2 onwards, packet drops on lossless PFC-enabled queues trigger syslogs that record the drop statistics, alerting you about the event. These syslogs enable monitoring of such queues and provide timely updates and alerts that help you take proactive action. You can troubleshoot quickly, isolate issues, and reroute traffic if necessary, with minimal impact on end-user services.

Here are some important highlights of the monitoring and logging of packet drops:

- *Only PFC-enabled queues* that also have the PFC Watchdog enabled are monitored.
- The syslog generation is event-based, where the event trigger is a packet drop on PFC Watchdog-monitored lossless queue between two consecutive measurement polls at a 10-minute interval. If the drop counters increment between such polls within a 10-minute interval, then this monitoring and logging functionality generates a syslog.
- Syslogs are generated for the following monitored parameters:
  - **Rx Data Frames Dropped** under **show controllers priority-flow-control statistics**. This counter accounts for virtual output queue (VOQ) tail drop statistics from all network processors across all ingress line cards in the system for a given egress port. VOQ tail drops occur when the PFC-enabled lossless queues build up beyond the user-configured headroom.
  - **Dropped Packets** under **show controllers priority-flow-control watchdog statistics**. This parameter accounts for all queue drops on the PFC-enabled lossless queues in the network processor on the egress line card.
- The **show controllers** command continues to display the total number of packets dropped.
- A typical syslog output looks like this:

```
%PLATFORM-PFC_WATCHDOG-5-PACKETDROP : PFC watchdog detected 150 packet drops on lossless
 priority 3 of interface HundredGigE0_7_0_30. Total drops 5678 packets.
```

Restrictions for Monitoring and Logging Packet Drops

The following restrictions apply to the monitoring and logging of packet drops for lossless PFC-enabled queues:

- The 10-minute polling interval is fixed, and you can't modify it.
- The monitoring and logging functionality is enabled by default on all PFC Watchdog-enabled queues, and you can't disable it.