

# QoS Troubleshooting Guide for NCS 5500 Series

## Contents

[Introduction](#)

[QoS Implementation on NCS5500](#)

[Troubleshooting Guidelines](#)

[Ingress QoS](#)

[Egress QoS](#)

[Monitor FMQs](#)

[Additional Commands to Collect](#)

[Limitations/Caveats](#)

[Marking](#)

[Bandwidth](#)

[Ingress QoS](#)

[Egress QoS](#)

[HQoS](#)

[Unsupported Configuration Warning after Commit](#)

## Introduction

This document describes the QoS troubleshooting guide for the Cisco Network Convergence System (NCS) 5500 series.

## QoS Implementation on NCS5500

- QoS in NCS5500 is performed on the ingress and egress TM of the NPU. NCS5500 implements a virtual output queue architecture, where egress queuing is based on credit requests and grant schemes.
- The actual buffering happens on the ingress forwarding devices. The egress forwarding device decides how much traffic can be sent by granting credits to any ingress forwarding device, which requests credit.
- All the queues of any egress interface or scheduling element are replicated to all ingress devices that want to send packets to that interface. These queues are called Virtual Output Queues (VOQ).
- Effectively 8 queues per interface or scheduling element are supported. On egress the scheduling element support priority, min, max, and excess. Scheduling is providing credits based on these attributes.
- Consequently, scheduling attributes of an interface are configured at egress, while congestion

control and congestion avoidance values are configured at ingress.

- When the IRPP decides to send a packet out for a given destination/output port it puts the packet into a specific VOQ. During enqueueing of the packet, it is verified if the packet can be enqueued based.
- On the VOQ's buffer occupancy and the Virtual Statistics Queue (VSQ) setting of the ingress interface. VSQ is a buffer management scheme based on traffic classes.
- If buffers are exceeded either the packet is dropped or flow-control is sent to the ingress port based on the corresponding user configuration. By default, packets are dropped if they fail the enqueue check.
- LCs of the NCS5500 don't have separate queueing ASICs. The platform implements virtual output-based queueing, which is based on scheduling. Thus there is no concept of ingress queueing. Ingress Service-policies supporting marking actions as well as policing only.
- NCS5500 implements a different concept for marking and queueing. It is based on qos-group and traffic-class. On egress, these two parameters are matched to remark traffic and to select queues.

Limitations existing for various commands. For details please *refer to the* .

## Troubleshooting Guidelines

### Ingress QoS

One of the most useful and important commands is **show qos interface <interface> input|output**

It provides useful information about your applied service-policy. It displays the configured and converted parameters( in case of policer, shaper, tail drop thresholds, etc.), the virtual queue ids, accounting type, and so forth.

Start with a simple policy on ingress.

```
class-map match-any i_match
match precedence ipv4 3
match mpls experimental topmost 3
end-class-map
!

policy-map ingress_marking
  class i_match
  set qos-group 3
  set discard-class 2
  !
  class class-default
  !
end-policy-map
!
```

```
RP/0/RP0/CPU0:xrg-ncs-04#show qos interface hundredGigE 0/0/0/35 input
```

```
Mon Dec 4 21:30:44.131 UTC
```

```
NOTE:- Configured values are displayed within parentheses
```

```
Interface HundredGigE0/0/0/35 ifh 0x138 -- input policy
```

```
NPU Id: 5
```

```
Total number of classes: 2
```

```
Interface Bandwidth: 100000000 kbps
```

```
Policy Name: ingress_marking
```

```
Accounting Type: Layer1 (Include Layer 1 encapsulation and above)
```

```
-----  
Level1 Class = i_match
```

```
New qos group = 3
```

```
New discard class = 2
```

```
Default Policer Bucket ID = 0x21
```

```
Default Policer Stats Handle = 0x0
```

```
Policer not configured for this class
```

```
Level1 Class = class-default
```

```
Default Policer Bucket ID = 0x20
```

```
Default Policer Stats Handle = 0x0
```

```
Policer not configured for this class
```

The qos\_ea is programming the policy in HW on the LC. With **show feature-mgr client qos-ea policy summary location <node>** you get an overview of configured policies on an LC.

```
RP/0/RP0/CPU0:xrg-ncs-04#show feature-mgr client qos-ea policy summary location 0/0/CPU0
```

```
Mon Dec 4 22:03:12.371 UTC
```

```
Pol-Num Flags Classes Ref-cnt In-Hash Compiled Policy Name
```

```
-----  
19 0x00004000 2 4 Yes Yes ingress_marking
```

If you look at the detailed view of this command, you can see where it matches in the classes:

```
RP/0/RP0/CPU0:xrg-ncs-04#show feature-mgr client qos-ea policy policy ingress_marking location 0/0/CPU0
```

```
Mon Dec 4 22:07:53.134 UTC
```

```
Policy name: ingress_marking
```

```
Policy Number: 19
```

```
Flags: 0x4000 [DNX_FM_POL_TOS]
```

```
Class Number: 2
```

```
Reference Count: 4
```

```
In Hash: Yes
```

```
Is Compiled: Yes
```

```
Level: 0 Class Number: 0 Class Name: i_match
```

```
Match Count: 2 Flags: 0x0 Logical Op: MATCH_ANY
```

```
ACE: 0 Class:0 Match:0 proto IPV4 Flags:0x8011 [Grant]
```

```
Source IPv4 Address: 0.0.0.0 (255.255.255.255)
```

```
Destination IPv4 Address: 0.0.0.0 (255.255.255.255)
```

```
Precedence Value: 3
```

```
UNKNOWN: UNKNOWN: UNKNOWN: ACE: 0 Class:0 Match:1 proto MPLS Flags:0x208040 [Grant]
```

```
MPLS EXP Value: 3
```

```
UNKNOWN: UNKNOWN: UNKNOWN: Final List
```

```
ACE: 0 Class:0 Match:0 proto IPV4 Flags:0x8011 [Grant]
```

```
Source IPv4 Address: 0.0.0.0 (255.255.255.255)
Destination IPv4 Address: 0.0.0.0 (255.255.255.255)
Precedence Value: 3
UNKNOWN: UNKNOWN: UNKNOWN: ACE: 1 Class:0 Match:1 proto MPLS Flags:0x208040 [Grant]
MPLS EXP Value: 3
UNKNOWN: UNKNOWN: UNKNOWN:
Level: 0 Class Number: 1 Class Name: class-default
Match Count: 1 Flags: 0x0 Logical Op: MATCH_ANY
ACE: 0 Class:1 Match:0 proto ANY Flags:0x28000 [Grant][Default]
UNKNOWN: UNKNOWN: UNKNOWN: Final List
ACE: 2 Class:1 Match:0 proto ANY Flags:0x28000 [Grant][Default]
UNKNOWN: UNKNOWN: UNKNOWN:
```

QOS\_EA information can also be displayed from a feature manager perspective. First display the summary:

```
RP/0/RP0/CPU0:xrg-ncs-04#show feature-mgr client qos-ea feature summary location 0/0/CPU0
Mon Dec 4 22:30:09.362 UTC
IFH NPU DIR Lookup-type ACL-ID Refcnt Feature-Name
-----
0x138 5 IN IPV4_QOS 14 1 ingress_marking:0
0x138 5 IN IPV6_QOS 14 1 ingress_marking:0
0x138 5 IN MPLS_QOS 14 1 ingress_marking:0
0x138 5 IN L2_QOS 14 1 ingress_marking:0
```

In the next step you can display the details:

```
RP/0/RP0/CPU0:xrg-ncs-04#show feature-mgr client qos-ea feature feature-name ingress_marking:0
direction ingress lookup mpls interface HundredGigE 0/0/0/35 location 0/0/CPU0
Mon Dec 4 22:31:20.588 UTC

Feature name: ingress_marking:0
Lookup Type: MPLS_QOS
Direction: IN
Reference Count: 1
NPU: 5
Interfaces Attached: HundredGigE0_0_0_35
Number of Entries: 2
Number of Verified Entries: 0
Number of ACEs: 2
ACL ID: 14
ACL ID Ref Cnt: 4
ACL ID Min Length(bits): 0
ACL ID Max Length(bits): 5
ACL ID Num Entries: 30
ACL ID Num Free Entries: 29
Compression Level: 0(UNCOMPRESSED)
```

The qos\_ea trace displays the order of events when the service-policy is applied on an interface. It would display problems in case it fails.

```
RP/0/RP0/CPU0:xrg-ncs-04#show qos ea trace all reverse location 0/0/CPU0
Mon Dec 4 22:17:26.297 UTC
3077 wrapping entries (36992 possible, 18816 allocated, 0 filtered, 3077 total)
Dec 4 21:30:18.280 qos_ea/int_evts 0/0/CPU0 t14563 CAPS: (line# 2882) : Responded to IM (batch
```

size: 1)

Dec 4 21:30:18.280 qos\_ea/int\_evts 0/0/CPU0 t14563 CAPS: post\_caps\_processing : Registering with ETHER EA for subid notification. Bulk count 1

Dec 4 21:30:18.280 qos\_ea/int\_evts 0/0/CPU0 t14563 CAPS: caps\_add\_notify\_remote Input num of interfaces 1

**Dec 4 21:30:18.280 qos\_ea/int\_evts 0/0/CPU0 t14563 CAPS: process\_caps\_add : End: count = 1 ok\_count 1. New num\_policies\_in 1 num\_policies\_out 2**

Dec 4 21:30:18.280 qos\_ea/int\_evts 0/0/CPU0 t14563 SCOL: scol\_interface\_add : ifh 0x138 ea\_intf\_type MainIf : interface added to SCOL tree

Dec 4 21:30:18.280 qos\_ea/int\_evts 0/0/CPU0 t14563 SCOL: scol\_interface\_add\_internal : ifh 0x138: after avl\_tree\_insert, new tree size is 1

Dec 4 21:30:18.280 qos\_ea/int\_evts 0/0/CPU0 t14563 UTIL: (line# 1665) Child info aggr bw add: Current propagated\_bw\_sum in npu\_ctx after adding propagated\_bw in if\_qos 0 is 0, ifh 0x138

Dec 4 21:30:18.280 qos\_ea/int\_evts 0/0/CPU0 t14563 UTIL: insert\_if\_qos\_from\_list\_by\_npu\_ctx : Inserting to list: if\_qos 0x3087c0bb78 prev 0x0 next 0x0 ifh 0x138 head 0x0 tail 0x0 count 0

**Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 ACTN: results\_add for ifh 0x138 with pmap ingress\_marking**

**Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 CAPS: fm\_program : npu\_id 5 ifh 0x138 policy\_name ingress\_marking :Compiling policy and sending VMR to FM**

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 ACTN: (line# 1597) : class\_name class-default offset 0 policer\_id 32: configuring policer...

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 POLICER-INT: policer\_get : cir\_kbps 720000000 cir\_burst\_kbits 32768 cir\_burst 4194304pir\_kbps 720000000 cir\_burst\_kbits 32768 pir\_burst 4194304

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 POLICER-INT: policer\_get : ifh 0x138 npu 5 offset 0 policer\_id 32 mode/policer\_type 0/0 inlif 0

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 POLICER-INT: policer\_get : for ifh 0x138 for npu 5 offset 0

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 ACTN: (line# 1597) : class\_name i\_match offset 1 policer\_id 33: configuring policer...

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 POLICER-INT: policer\_get : cir\_kbps 720000000 cir\_burst\_kbits 32768 cir\_burst 4194304pir\_kbps 720000000 cir\_burst\_kbits 32768 pir\_burst 4194304

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 POLICER-INT: policer\_get : ifh 0x138 npu 5 offset 1 policer\_id 33 mode/policer\_type 0/0 inlif 0

Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 POLICER-INT: policer\_get : for ifh 0x138 for npu 5 offset 1

**Dec 4 21:30:18.217 qos\_ea/int\_evts 0/0/CPU0 t14563 ACTN: mark\_actn\_process Unconditional mark set for cmaps i\_match**

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 ACTN: resource\_cfg for ifh 0x138 ul\_ifh 0x0 with pmap ingress\_marking

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 CMAP: classmap\_process : chd 0x7feedd339b98, if\_qos 0x3087c0bb78, class\_info 0x3087c0b368

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 PWALK: preprocess\_action : ifh 0x138 ul\_ifh 0x0 policy\_name ingress\_marking class\_name class-default

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 PWALK: pwalk\_process\_class : depth 0, usr\_data->class\_list 0x3087c0bb90, class\_name class-default

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 CMAP: classmap\_process : chd 0x7feedd33b478, if\_qos 0x3087c0bb78, class\_info 0x3087c0b770

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 PWALK: preprocess\_action : ifh 0x138 ul\_ifh 0x0 policy\_name ingress\_marking class\_name i\_match

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 PWALK: pwalk\_process\_class : depth 0, usr\_data->class\_list 0x3087c0bb90, class\_name i\_match

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 PWALK: pwalk\_start\_level : depth 0, stack 0x7fff8a11be40 ctx 0x7fff8a11c670

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 PWALK: policy\_walk : policy\_oper 0 policy\_name ingress\_marking ifh 0x138 ul\_ifh 0x0 walk\_flags 0x0

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 UTIL: if\_qos\_insert\_to\_db : Inserting to hash table: npu\_id 5 ifh 0x138, ul\_ifh 0x0, dir 0

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 RATE: dnx\_gosea\_policy\_resolve : policy\_name ingress\_marking ifh 0x138 ul\_ifh 0x0 intf\_kbps 100000000

Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 CAPS: policy\_add : npu\_id 5 ifh 0x138 ul\_ifh 0x0 dir ingress policy\_name ingress\_marking num\_classes 0 max\_hierarchy 0

**Dec 4 21:30:18.100 qos\_ea/int\_evts 0/0/CPU0 t14563 VRFY: policy\_verify : Verifying policy**

**ingress\_marking pmaphd 0x7feedd2c1088**

```
Dec 4 21:30:18.100 qos_ea/int_evts 0/0/CPU0 t14563 UTIL: if_qos_init : Initializing if_qos
0x3087c0bb78 with npu_id 5 ifh 0x138 ul_ifh 0x0 direction ingress ea_intf_type MainIf
Dec 4 21:30:18.100 qos_ea/int_evts 0/0/CPU0 t14563 VRFY: check_policy_on_parent_subintf : Did
not find any reference to parent interface (0x138) in par_child_info_array
Dec 4 21:30:18.100 qos_ea/int_evts 0/0/CPU0 t14563 VRFY: check_policy_on_parent_subintf :
parent_ifh: 0x138, dir: 0 (ingress)
Dec 4 21:30:18.100 qos_ea/int_evts 0/0/CPU0 t14563 UTIL: intf_bw_get : Bandwidth of ifh 0x138 is
intf_kbps: 100000000 kbps
Dec 4 21:30:18.100 qos_ea/int_evts 0/0/CPU0 t14563 CAPS: caps_add_int : ifh 0x138 ea_intf_type
MainIf dir ingress
Dec 4 21:30:18.100 qos_ea/int_evts 0/0/CPU0 t14563 CAPS: caps_add : ifh 0x138 ea_intf_type
MainIf dir ingress policy_name ingress_marking
Dec 4 21:30:18.100 qos_ea/int_evts 0/0/CPU0 t14563 CAPS: process_caps_add : 1 of 1 direction
ingress on interface 0x138 policy_name ingress_marking
Dec 4 21:30:18.093 qos_ea/int_evts 0/0/CPU0 t14563 CAPS: process_caps_add : batch count = 1 .
Current num_policies_in 0 num_policies_out 2
Dec 4 21:30:18.093 qos_ea/int_evts 0/0/CPU0 t14563 INT: im_msg_handler : Rx: caps add message
(batch size: 1)
```

When you change the policy and add for instance a policer to class i\_match it will be visible in the qos\_ea trace. In-place modification of a policy map is supported on NCS5500.

```
RP/0/RP0/CPU0:xrg-ncs-04(config-pmap-c)#police rate percent 50
RP/0/RP0/CPU0:xrg-ncs-04(config-pmap-c-police)#commit
```

```
Dec 4 22:49:39.567 qos_ea/int_evts 0/0/CPU0 t14563 POLICER-INT: policer_get : for ifh 0x138 for
npu 5 offset 0
Dec 4 22:49:39.567 qos_ea/int_evts 0/0/CPU0 t14563 POLICER-INT: (line# 147) : cir_kbps 50000000
cir_burst 4194304pir_kbps 0 pir_burst 0
Dec 4 22:49:39.567 qos_ea/int_evts 0/0/CPU0 t14563 POLICER-INT: (line# 144) : ifh 0x138 npu 5
offset 1 policer_type = 0 is_parent=0policer_id = 33
Dec 4 22:49:39.567 qos_ea/int_evts 0/0/CPU0 t14563 ACTN: (line# 1822) : class_name i_match
offset 0 policer_id 33: configuring policer...
Dec 4 22:49:39.567 qos_ea/int_evts 0/0/CPU0 t14563 ACTN: (line# 1597) : class_name i_match
offset 1 policer_id 33: configuring policer...
```

The show qos interface command reflects the addition as well:

```
RP/0/RP0/CPU0:xrg-ncs-04#show qos interface hundredGigE 0/0/0/35 input
Mon Dec 4 23:25:19.684 UTC
NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/0/0/35 ifh 0x138 -- input policy
NPU Id: 5
Total number of classes: 2
Interface Bandwidth: 100000000 kbps
Policy Name: ingress_marking
Accounting Type: Layer1 (Include Layer 1 encapsulation and above)
-----
Level1 Class = i_match
New qos group = 3
New discard class = 2

Policer Bucket ID = 0x21
Policer Stats Handle = 0x0
Policer committed rate = 49680000 kbps (50 %)
Policer conform burst = 4194304 bytes (default)

Level1 Class = class-default
```

```
Default Policer Bucket ID = 0x20
Default Policer Stats Handle = 0x0
Policer not configured for this class
```

The qos\_ea trace displays the converted value as well when the above command is executed.

The policer uses an incremental step size of 22 kbps. The configured value is rounded to the nearest multiple of 22kbps when used in the formula on the line card.

```
Dec 4 23:25:19.773 qos_ea/int_evts 0/0/CPU0 t14563 POLICER-INT: policer_get : cir_kbps 49680000
cir_burst_kbits 32768 cir_burst 4194304pir_kbps 49680000 cir_burst_kbits 32768 pir_burst 4194304
Dec 4 23:25:19.773 qos_ea/int_evts 0/0/CPU0 t14563 POLICER-INT: policer_get : ifh 0x138 npu 5
offset 1 policer_id 33 mode/policer_type 0/0 inlif 0
Dec 4 23:25:19.773 qos_ea/int_evts 0/0/CPU0 t14563 POLICER-INT: policer_get : for ifh 0x138 for
npu 5 offset 1
```

## Egress QoS

Egress QoS is supporting queueing actions. On NCS5500 one can have two service-policies in the egress direction. One for marking and one for queueing. This is for a marking action:

```
class-map match-any e_mark
match qos-group 3
end-class-map
!
```

```
policy-map egress_marking
class e_mark
set mpls experimental imposition 2
!
class class-default
!
end-policy-map
!
```

```
RP/0/RP0/CPU0:xrg-ncs-06#show feature-mgr client qos-ea policy summary location 0/0/CPU0
Thu Dec 21 10:44:33.978 UTC
Pol-Num Flags Classes Ref-cnt In-Hash Compiled Policy Name
-----
1 0x01000000 2 1 Yes Yes egress_marking
```

```
RP/0/RP0/CPU0:xrg-ncs-06#show feature-mgr client qos-ea feature summary location 0/0/CPU0
Thu Dec 21 10:59:08.872 UTC
IFH NPU DIR Lookup-type ACL-ID Refcnt Feature-Name
-----
0x340 0 OUT MAP_ID_QOS 0 1 egress_marking:0
```

## Egress Queueing Policy

```
class-map match-any tc1
match traffic-class 1
end-class-map
!
```

```

class-map match-any tc2
  match traffic-class 2
end-class-map
!
class-map match-any tc3
  match traffic-class 3
end-class-map
!
class-map match-any tc4
  match traffic-class 4
end-class-map
!
class-map match-any tc5
  match traffic-class 5
end-class-map
!
policy-map POLICY_QOS_OUT
  class tc1
    priority level 1
    shape average percent 19
    queue-limit 500 us
  !
  class tc2
    priority level 2
    shape average percent 20
    queue-limit 500 us
  !
  class tc3
    bandwidth percent 1
  !
  class tc4
    bandwidth percent 20
  !
  class tc5
    bandwidth percent 20
  !
  class class-default
    bandwidth percent 20
  !
end-policy-map
!

```

One can quick verify where a QoS policy is applied.

```

RP/0/RP0/CPU0:xrg-ncs-04#show policy-map targets pmap-name POLICY_QOS_OUT location 0/0/CPU0
Mon Dec 18 21:13:01.463 UTC
1) Policymap: POLICY_QOS_OUT Type: qos
Targets (applied as main policy):
  HundredGigE0/0/0/35 output
Total targets: 1

Targets (applied as child policy):
Total targets: 0

```

The show QoS interface command displays the configured parameters and also the queue IDs for each class with a queueing action configured. Furthermore, you can obtain defaults like the queue size.

```

RP/0/RP0/CPU0:xrg-ncs-04#show qos interface hundredGigE 0/0/0/35 output

```



Thu Dec 14 20:56:29.796 UTC

NOTE:- Configured values are displayed within parentheses

Interface HundredGigE0/0/0/35 ifh 0x138 -- output policy

NPU Id: 5

Total number of classes: 6

Interface Bandwidth: 100000000 kbps

**VOQ Base: 1032**

Accounting Type: Layer1 (Include Layer 1 encapsulation and above)

-----  
Level1 Class (HP1) = tc1  
**Egressq Queue ID = 1033 (HP1 queue)**  
Queue Max. BW. = 0 kbps (19 %)  
**Guaranteed service rate = 19000000 kbps**  
TailDrop Threshold = 1187840 bytes / 500 us (500 us)  
WRED not configured for this class

Level1 Class (HP2) = tc2  
Egressq Queue ID = 1034 (HP2 queue)  
Queue Max. BW. = 0 kbps (20 %)  
**Guaranteed service rate = 20000000 kbps**  
TailDrop Threshold = 1253376 bytes / 501 us (500 us)  
WRED not configured for this class

Level1 Class = tc3  
**Egressq Queue ID = 1035 (LP queue)**  
Queue Max. BW. = 100884395 kbps (default)  
**Queue Min. BW. = 1020015 kbps (1 %)**  
Inverse Weight / Weight = 1 / (BWR not configured)  
**Guaranteed service rate = 1000000 kbps**  
TailDrop Threshold = 1253376 bytes / 10 ms (default)  
WRED not configured for this class

Level1 Class = tc4  
**Egressq Queue ID = 1036 (LP queue)**

Queue Max. BW. = 100824615 kbps (default)  
**Queue Min. BW. = 20164923 kbps (20 %)**  
Inverse Weight / Weight = 1 / (BWR not configured)  
**Guaranteed service rate = 20000000 kbps**  
TailDrop Threshold = 25034752 bytes / 10 ms (default)  
WRED not configured for this class

Level1 Class = tc5  
**Egressq Queue ID = 1037 (LP queue)**  
Queue Max. BW. = 100824615 kbps (default)  
Queue Min. BW. = 20164923 kbps (20 %)  
Inverse Weight / Weight = 1 / (BWR not configured)  
Guaranteed service rate = 20000000 kbps  
TailDrop Threshold = 25034752 bytes / 10 ms (default)  
WRED not configured for this class

Level1 Class = class-default  
**Egressq Queue ID = 1032 (Default LP queue)**  
Queue Max. BW. = 100824615 kbps (default)  
Queue Min. BW. = 20164923 kbps (20 %)  
Inverse Weight / Weight = 1 / (BWR not configured)  
Guaranteed service rate = 20000000 kbps  
TailDrop Threshold = 25034752 bytes / 10 ms (default)  
WRED not configured for this class

RP/0/RP0/CPU0:xrg-ncs-04#show feature-mgr client qos-ea policy summary location 0/0/CPU0

Thu Dec 14 21:27:30.486 UTC

Pol-Num	Flags	Classes	Ref-cnt	In-Hash	Compiled	Policy Name
3	0x00010000	6	0	Yes	No	POLICY_QOS_OUT
2	0x00000000	1	0	Yes	No	p_out
1	0x00004000	2	4	Yes	Yes	ingress_marking

RP/0/RP0/CPU0:xrg-ncs-04#show feature-mgr client qos-ea policy policy POLICY\_QOS\_OUT location

0/0/CPU0

Thu Dec 14 21:28:22.182 UTC

Policy name: **POLICY\_QOS\_OUT**  
Policy Number: 3  
Flags: 0x10000 [DNX\_FM\_POL\_TC]  
Class Number: 6  
Reference Count: 0  
In Hash: Yes  
Is Compiled: No

Level: 0 Class Number: 0 Class Name: tc1

Match Count: 1 Flags: 0x0 Logical Op: MATCH\_ANY

ACE: 0 Class:0 Match:0 proto ANY Flags:0x8080 [Grant]

DNX\_FM\_RANGE\_TC: (1, 1)

UNKNOWN: UNKNOWN:

Level: 0 Class Number: 1 Class Name: tc2

Match Count: 1 Flags: 0x0 Logical Op: MATCH\_ANY

ACE: 0 Class:1 Match:0 proto ANY Flags:0x8080 [Grant]

DNX\_FM\_RANGE\_TC: (2, 2)

UNKNOWN: UNKNOWN:

Level: 0 Class Number: 2 Class Name: tc3

Match Count: 1 Flags: 0x0 Logical Op: MATCH\_ANY

ACE: 0 Class:2 Match:0 proto ANY Flags:0x8080 [Grant]

DNX\_FM\_RANGE\_TC: (3, 3)

UNKNOWN: UNKNOWN:

Level: 0 Class Number: 3 Class Name: tc4

Match Count: 1 Flags: 0x0 Logical Op: MATCH\_ANY

ACE: 0 Class:3 Match:0 proto ANY Flags:0x8080 [Grant]

DNX\_FM\_RANGE\_TC: (4, 4)

UNKNOWN: UNKNOWN:

Level: 0 Class Number: 4 Class Name: tc5

Match Count: 1 Flags: 0x0 Logical Op: MATCH\_ANY

ACE: 0 Class:4 Match:0 proto ANY Flags:0x8080 [Grant]

DNX\_FM\_RANGE\_TC: (5, 5)

UNKNOWN: UNKNOWN:

Level: 0 Class Number: 5 Class Name: class-default

Match Count: 1 Flags: 0x0 Logical Op: MATCH\_ANY

ACE: 0 Class:5 Match:0 proto ANY Flags:0x28000 [Grant][Default]

UNKNOWN: UNKNOWN:

Adding WRED:

```
policy-map POLICY_QOS_OUT
class tc4
random-detect discard-class 1 5 ms 15 ms
!
class class-default
!
end-policy-map
!
end
```

```
Level1 Class = tc4
Egressq Queue ID = 1036 (LP queue)
Queue Max. BW. = 100824615 kbps (default)
Queue Min. BW. = 20164923 kbps (20 %)
Inverse Weight / Weight = 1 / (BWR not configured)
Guaranteed service rate = 20000000 kbps
TailDrop Threshold = 75497472 bytes / 30 ms (default)
```

WRED profile for **Discard\_Class 1**

```
WRED Min. Threshold = 12517376 bytes (5 ms)
WRED Max. Threshold = 37748736 bytes (15 ms)
```

Default RED profile

```
WRED Min. Threshold = 0 bytes (0 ms)
WRED Max. Threshold = 0 bytes (0 ms)
```

RP/0/RP0/CPU0:xrg-ncs-04#show controllers npu voq-usage interface HundredGigE 0/0/0/35 instance 5 location 0/0/CPU0

Tue Nov 7 17:24:03.570 UTC

-----

Node ID: 0/0/CPU0

Intf	Intf	NPU	NPU	PP	Sys	VOQ	Flow	VOQ	Port
name	handle	#	core	Port	Port	base	base	port	speed
	(hex)							type	(Gbps)

-----  
Hu0/0/0/35 138 5 0 1 321 1032 1064 local 100 <- 1032 (Default LP queue) - displays the base queue -> refer to show qos interface commaand.

Display the queue details:

RP/0/RP0/CPU0:xrg-ncs-04#**show controllers fia diagshell 5 "diag cosq voq id=1036 det=1" location 0/0/CPU0** <- 1036 (LP queue) - refer to show qos interface commaand.

Thu Dec 14 21:51:33.184 UTC

Node ID: 0/0/CPU0

Core 0:

Basic info

Q type: voq

num cos: 8, cosq class: 4

Base queue id: 1032, base queue gport: 0x24000408

Credit request type: BCM\_COSQ\_DELAY\_TOLERANCE\_100G\_SLOW\_ENABLED Adjusted for slow enabled 100Gb ports

Watchdog enable in common status message mode

Is queue in credit watchdog queue range:True

Credit watchdog message time: 0

Delete queue time: 512

Backoff enter queue credit balance threshold: 40960

Backoff exit queue credit balance threshold: 40960

Backlog enter queue credit balance threshold: 40960

Backlog exit queue credit balance threshold: 40960

Empty queue satisfied credit balance threshold: 0

Max empty queue credit balance threshold: 16384

Exceed max empty queue credit balance threshold: 1

Off-To-Slow credit balance threshold: 0

Off-To-Normal credit balance threshold: 38912

Slow-To-Normal credit balance threshold: 38912

Normal-To-Slow credit balance threshold: 38912

Delay Tolerance is NOT OCB only

Delay Tolerance is NOT High Q Priority

Slow Level Thresh Down in slow level number: 0 is: 0

Slow Level Thresh Down in slow level number: 1 is: 0

Slow Level Thresh Down in slow level number: 2 is: 0

Slow Level Thresh Down in slow level number: 3 is: 0

Slow Level Thresh Down in slow level number: 4 is: 0

Slow Level Thresh Down in slow level number: 5 is: 0

Slow Level Thresh Down in slow level number: 6 is: 0

Slow Level Thresh Up in slow level number: 0 is: 0

Slow Level Thresh Up in slow level number: 1 is: 0

Slow Level Thresh Up in slow level number: 2 is: 0

Slow Level Thresh Up in slow level number: 3 is: 0

Slow Level Thresh Up in slow level number: 4 is: 0

Slow Level Thresh Up in slow level number: 5 is: 0

Slow Level Thresh Up in slow level number: 6 is: 0

Credit value(local): 1024, Credit value(remote): 2048

Credit discount value: -2

Rate class info

WRED info:

Green : enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)

**Yellow: enable(True) min\_thresh(12517376) max\_thresh(37748736)  
drop\_probability(9)**

Red : enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)

Black : enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)

Tail drop info:

Green : max queue size in bytes(75497472), max queue size in BDs(294912)

Yellow: max queue size in bytes(75497472), max queue size in BDs(294912)

Red : max queue size in bytes(75497472), max queue size in BDs(294912)

Black : max queue size in bytes(75497472), max queue size in BDs(294912)

Guaranteed info:

Green : min queue size in bytes(0), min queue size in BDs(0)

Yellow: min queue size in bytes(0), min queue size in BDs(0)

Red : min queue size in bytes(0), min queue size in BDs(0)

Black : min queue size in bytes(0), min queue size in BDs(0)

Fair adaptive tail drop info:

Enable: True

Green : alpha(-2)

Yellow: alpha(-2)

Red : alpha(-2)

Black : alpha(-2)

ECN wred info: enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)

ECN max queue size in bytes(524288), max queue size in BDs(524288)

VSQ-related

Category class: 2

Traffic class: 4

Connection class: 0

Green : Admission-test-templates(0)

Yellow: Admission-test-templates(0)

Red : Admission-test-templates(0)

Black : Admission-test-templates(0)

AdmissionTestProfileA[0]: 'category, traffic class'

AdmissionTestProfileB[0]: 'category'

**current queue size: 0 bytes, current queue bds size: 0**

Attached VOQ connector: 0x00000428

Destination sys port: 0x00007fff

OCB eligiblity: True

[SNIP]

output for core 1 is ommitted.

To monitor the queue size you can run the above command and grep for the corresponding line.

Example: **show controllers fia diagshell 5 "diag cosq voq id=1036 core=0 det=1" location 0/0/CPU0 | i current queue size**

Statistics can be displayed with the **show policy-map interface** command.

Example: **show policy-map interface hundredGigE 0/0/0/35 output location 0/0/CPU0**

Note: The output is not detailed as we know from other XR platforms. It does not display lines for the corresponding WRED or RED profile.

```
RP/0/RP0/CPU0:xrg-ncs-04#show policy-map interface hundredGigE 0/0/0/35 output location 0/0/CPU0
```

```
Thu Dec 14 22:05:50.971 UTC
```

```
Interface:HundredGigE0/0/0/35 Location: node0_0_CPU0 output: POLICY_QOS_OUT
```

```
Class tc1
```

Classification statistics	(packets/bytes)	(rate - kbps)
Matched :	0/0	0
Transmitted :	0/0	0
Total Dropped :	0/0	0

```
Queueing statistics
```

Queue ID	: 1033
Taildropped(packets/bytes)	: 0/0

```
Class tc2
```

Classification statistics	(packets/bytes)	(rate - kbps)
Matched :	0/0	0
Transmitted :	0/0	0
Total Dropped :	0/0	0

```
Queueing statistics
```

Queue ID	: 1034
Taildropped(packets/bytes)	: 0/0

```
Class tc3
```



Classification statistics	(packets/bytes)	(rate - kbps)
Matched :	0/0	0
Transmitted :	0/0	0
Total Dropped :	0/0	0

Queueing statistics

Queue ID	: 1035
Taildropped(packets/bytes)	: 0/0

Class tc4

Classification statistics	(packets/bytes)	(rate - kbps)
Matched :	0/0	0
Transmitted :	0/0	0
Total Dropped :	0/0	0

Queueing statistics

Queue ID	: 1036
Taildropped(packets/bytes)	: 0/0

Class tc5

Classification statistics	(packets/bytes)	(rate - kbps)
Matched :	0/0	0
Transmitted :	0/0	0
Total Dropped :	0/0	0

Queueing statistics

Queue ID	: 1037
Taildropped(packets/bytes)	: 0/0

Class class-default

Classification statistics	(packets/bytes)	(rate - kbps)
Matched :	0/0	0
Transmitted :	0/0	0
Total Dropped :	0/0	0

Queueing statistics

Queue ID	: 1032
Taildropped(packets/bytes)	: 0/0

## Monitor FMQs

Multicast traffic is not scheduled. It is using Fabric Multicast Queues (FMQ) instead of the conventional VOQs. On that score egress multicast, QoS is not supported. To display the statistics of the FMQs this command is available from 6.5.2 onwards:

```
RP/0/RP0/CPU0:NCS5508#show controllers npu stats voq base 0 instance 0 location 0/0/CPU0
Mon Jul 8 08:59:22.465 UTC
```

```
Asic Instance = 0
```

```
VOQ Base = 0
```

	ReceivedPkts	ReceivedBytes	DroppedPkts	DroppedBytes
TC_0 = 224		15673	49	3468
TC_1 = 0		0	0	0
TC_2 = 0		0	0	0
TC_3 = 0		0	0	0
TC_4 = 0		0	0	0
TC_5 = 0		0	0	0
TC_6 = 0		0	0	0
TC_7 = 0		0	0	0

It is also possible to follow a packet across the NPU and check if for instance the appropriate Traffic Class (TC) is applied. Please have a look at

Section **Follow a transit packet** for the corresponding diag shell commands and filter.

## Additional Commands to Collect

These are some additional commands:

- **show tech-support qos pi**
- **show tech-support qos platform**
- **show tech cef platform**

If a problem is suspected on an individual interface:

- **show run interface <name>**
- **show run policy-map <>** (for the policy-maps applied on the interface)
- **show qos interface <name> input|output**
- **show policy-map interface <name>**
- **show qos remote location <node>** (for egress policy)

If the interface is a bundle, provide complete bundle member information.

- **show bundle <>**

## Limitations/Caveats

### Marking

- L2(COS, DEI) and L3(MPLS EXP) marking are not supported in the same policy-map
- The policer is not working when it is combined with **set discard-class**.
- match qos-group is for marking only.

- qos-group 0 is reserved for the default class.
- statistics and counters are not displayed for a marking policy.

## Bandwidth

- The sum of all **bandwidth** statements in a policy-map must not be > 100%
- The sum of all **bandwidth remaining** statements in a policy-map must not be > 100%
- A shaper is not shared between classes and hence there are no restrictions to the summary of bandwidth statements.

## Ingress QoS

Queuing actions are not supported in an ingress service-policy.

## Egress QoS

- Marking capabilities/actions are very limited.
- An egress policy match on traffic-class (TG) , qos-group (QG) and discard-class (DC) are only supported. TC - for queuing actions QG - for marking actions DC - WRED
- A service-policy connecting the following policy-map would be rejected on an egress interface:\

```
policy-map SET-AND-QUEUE-OUT
  class GOLD
    priority level 1
    shape average percent 20
    queue-limit 500 us
    set mpls experimental imposition 5
```

The error message you get is: !!% "DNX\_QOSEA" detected the 'warning' condition "Invalid egress policy-map configuration. Use qos-group for marking features and traffic-class for queuing features".

It is possible to apply two separate service policies, one for marking and the other one for queuing.

## HQoS

- Apply a service-policy on a subinterface requires the HQoS profile independent if it is flat.
- Apply a service-policy on a sub-interface containing a priority class requires a shaper in this class independent if it is hierarchical or flat.
- In HQoS mode, priority classes must have a rate limit configuration (shaper). The effective shaper value is taken as a priority bandwidth reservation.
- In HQoS mode, bandwidth (BW) and bandwidth remaining (BWR) are not supported simultaneously under the same policy-map configuration. Either bandwidth or bandwidth remaining can be configured in the same policy-map.
- BW respective BWR along with shape is not supported in the same policy-map before release 6.3.1.
- WRED is not allowed in the parent class

- A maximum of 4 priority levels are supported in HQoS mode
- Ingress HQoS policing and Ingress peering profile are not supported together.
- For the two-level policy, user-defined classes at the parent level are not supported.
- Because of a TCAM limitation, Policy-Based Tunnel Selection (PBTS) does not work when the HQoS profile is enabled.

## Unsupported Configuration Warning after Commit

In some cases you may see a warning log after committing a QoS config like the following example:

```
% Validation warnings detected as a result of the commit operation. Please issue 'show configuration warnings' to view the warnings
```

If you execute the suggested CLI you see:

```
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.
```

<Here the reason is printed>

Nevertheless, the config is applied. In this case it might come from the yang verification.

Please check if **configuration validation enable** is configured.

If this is the case it is likely that a yang rule has fired.

You can switch on **debug configuration validation ccv detail** and commit the config again.

You should see the yang rule, which is triggering the "Unsupported" message in the debug output.

Example:

```
RP/0/RP0/CPU0:Jan 22 15:57:44.441 UTC: ccv[189]: ...message : 'Service Policy on interface
"TenGigE0/0/0/7/1.102" is not supported', rule group name: 'qos-ma', rule name: 'cac-ser-pol-
qos-in-unsupp-intf').
```

Please contact the QoS PI team in this case.