

# Understand Nexus 9000 TAHUSD Buffer Syslog & Congestion

## Contents

---

### [Introduction](#)

### [Prerequisites](#)

[Requirements](#)

[Components Used](#)

### [Background Information](#)

#### [Understand Cisco Nexus 9000 Cloud Scale ASIC Buffering Architecture](#)

[Calculate Instance ID for Multiple ASIC and Slices](#)

#### [Understand Oversubscription and Output Discards](#)

[Understand the BUFFER\\_THRESHOLD\\_EXCEEDED Syslog](#)

[Understand the Output Discards Interface Counter](#)

#### [Example Oversubscription Scenario](#)

### [Next Steps](#)

### [Additional Information](#)

[BUFFER\\_THRESHOLD\\_EXCEEDED Syslog Configuration Options](#)

[Logs to Collect for Network Congestion Scenarios](#)

[Monitoring Micro-Bursts](#)

### [Related Information](#)

---

## Introduction

This document describes queueing and buffering on Cisco Nexus 9000 Series switches equipped with a Cisco Scale ASIC that runs NX-OS software.

## Prerequisites

### Requirements

Cisco recommends that you understand the basics of Ethernet switching on shared medium networks and the necessity of queueing/buffering in these networks. Cisco also recommends that you understand the basics of Quality of Service (QoS) and buffering on Cisco Nexus switches. For more information, refer to the documentation here:

- [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 10.1\(x\)](#)
- [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.3\(x\)](#)
- [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2\(x\)](#)
- [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 7.x](#)

### Components Used

The information in this document is based on Cisco Nexus 9000 series switches with the Cloud Scale ASIC

running NX-OS software release 9.3(8).

The procedure covered in this document is applicable only to the hardware shown here.

- **Nexus 9200/9300 Fixed Switches**

- N9K-C92160YC-X
- N9K-C92300YC
- N9K-C92304QC
- N9K-C92348GC-X
- N9K-C9236C
- N9K-C9272Q
- N9K-C9332C
- N9K-C9364C
- N9K-C93108TC-EX
- N9K-C93108TC-EX-24
- N9K-C93180LC-EX
- N9K-C93180YC-EX
- N9K-C93180YC-EX-24
- N9K-C93108TC-FX
- N9K-C93108TC-FX-24
- N9K-C93180YC-FX
- N9K-C93180YC-FX-24
- N9K-C9348GC-FXP
- N9K-C93240YC-FX2
- N9K-C93216TC-FX2
- N9K-C9336C-FX2
- N9K-C9336C-FX2-E
- N9K-C93360YC-FX2
- N9K-C93180YC-FX3
- N9K-C93108TC-FX3P
- N9K-C93180YC-FX3S
- N9K-C9316D-GX
- N9K-C93600CD-GX
- N9K-C9364C-GX
- N9K-C9364D-GX2A
- N9K-C9332D-GX2B

- **Nexus 9500 Modular Switch Line Cards**

- N9K-X97160YC-EX
- N9K-X9732C-EX
- N9K-X9736C-EX
- N9K-X97284YC-FX
- N9K-X9732C-FX
- N9K-X9788TC-FX
- N9K-X9716D-GX

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

## **Background Information**

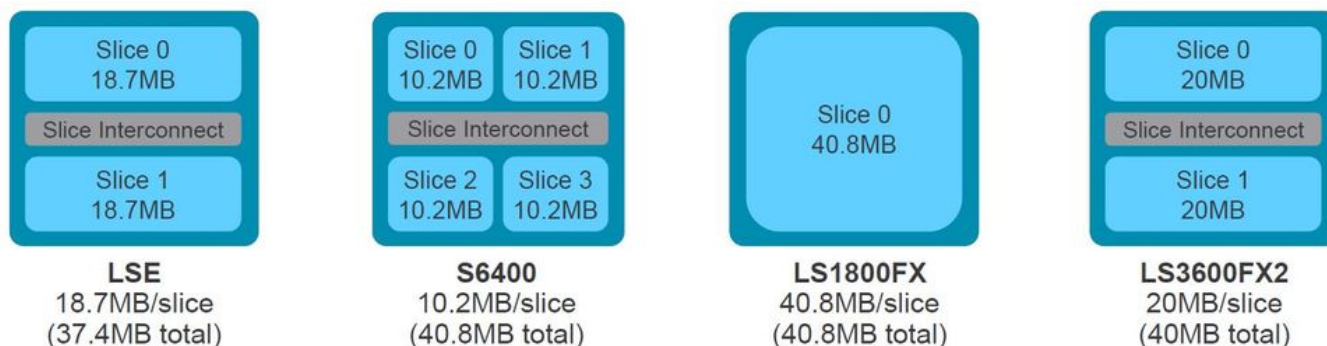
This document describes the mechanics behind queueing and buffering on Cisco Nexus 9000 Series switches equipped with a Cisco Cloud Scale ASIC (Application-Specific Integrated Circuit) running NX-OS

software. This document also describes symptoms of port oversubscription on this platform, such as non-zero output discard interface counters and syslog that indicate buffer thresholds have been exceeded.

## Understand Cisco Nexus 9000 Cloud Scale ASIC Buffering Architecture

Cisco Nexus 9000 Series switches with the Cisco Cloud Scale ASIC implement a “shared-memory” egress buffer architecture. An ASIC is divided into one or more “slices”. Each slice has its own buffer, and only ports within that slice can use that buffer. Physically, each slice is divided into “cells”, which represent portions of the buffer. Slices are partitioned into “pool-groups”. A certain number of cells are allocated to each pool-group, and they are not shared among separate pool-groups. Each pool-group has one or more “pools”, which represent a class of service (CoS) for unicast or multicast traffic. This helps each pool-group guarantee buffer resources for the types of traffic the pool-group serves.

The image here visually demonstrates how various models of Cisco Cloud Scale ASIC are divided into slices. The image also demonstrates how each slice is allocated a certain amount of buffer through cells.



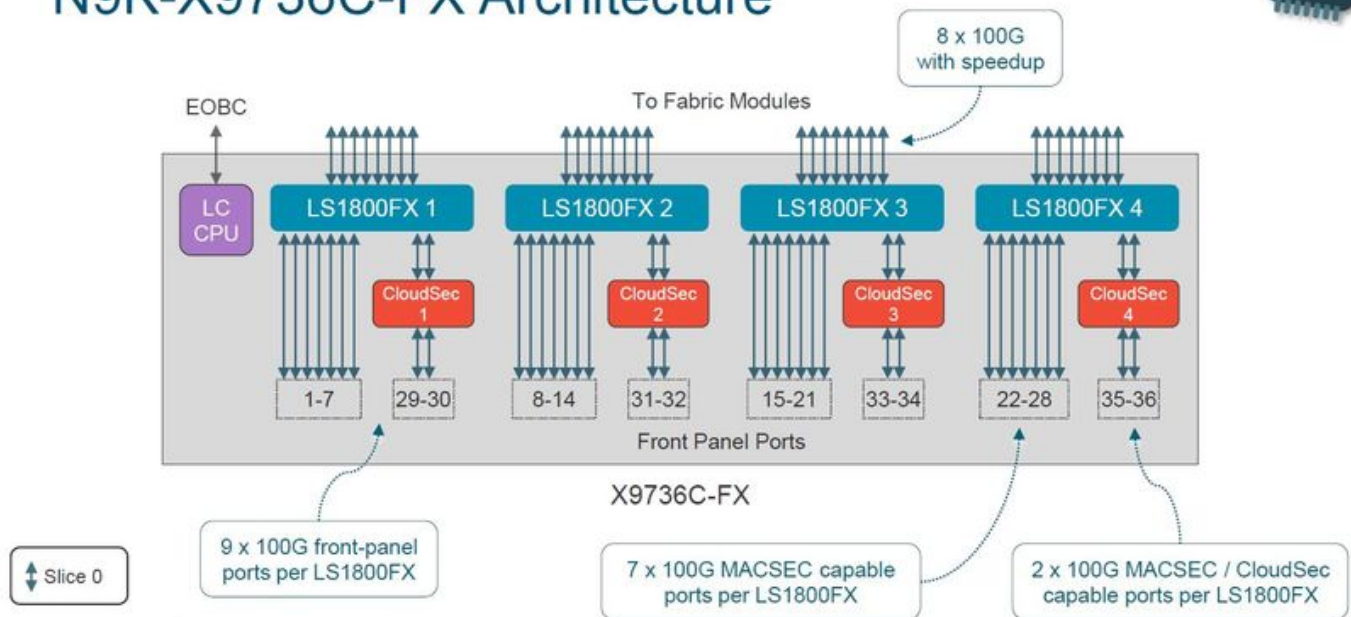
Each model of Nexus 9000 Series switch and Nexus 9500 line card has a different number of Cisco Cloud Scale ASICs inside, as well as a different layout that dictates which front-panel ports connect to which ASIC. Two examples that use the N9K-X9736C-FX line card and the N9K-C9336C-FX2 switch are shown in the images here.

The N9K-C9736C-FX line card has 4 Cisco Cloud Scale LS1800FX ASICs with one slice per ASIC. Internally, each ASIC is referred to as a "unit". Each slice is referred to as an "instance" and is assigned a zero-based integer that uniquely identifies that slice within the chassis. This results in the permutations shown here:

- Unit 0, slice 0 is referred to as instance 0
- Unit 1, slice 0 is referred to as instance 1
- Unit 2, slice 0 is referred to as instance 2
- Unit 3, slice 0 is referred to as instance 3



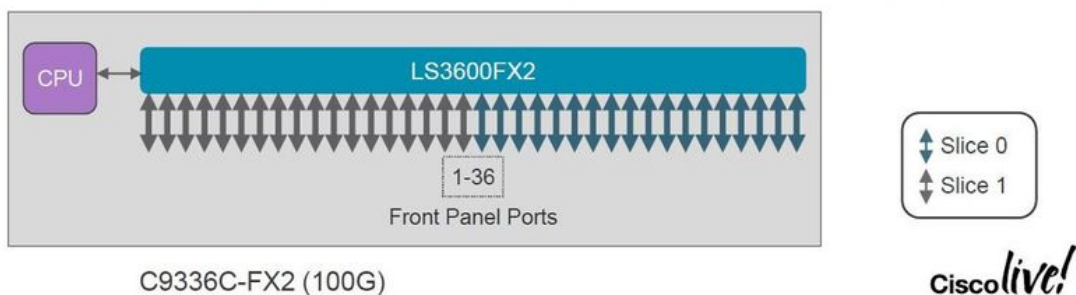
# N9K-X9736C-FX Architecture



The N9K-C9336C-FX2 switch has one Cisco Cloud Scale LS3600FX2 ASIC with two slices per ASIC. Internally, each ASIC is referred to as a "unit". Each slice is referred to as an "instance" and is assigned a zero-based integer that uniquely identifies that slice within the chassis. This results in the permutations shown here:

- Unit 0, slice 0 is referred to as instance 0
- Unit 0, slice 1 is referred to as instance 1

# Nexus 9300-FX2 Switch Architecture



Every line card and switch has a different layout and result in different instance numbers. For you to design your network around bandwidth-intensive traffic flows you need to understand the switch or line card layout you want to work with. The **show interface hardware-mappings** command can be used to correlate each front-panel port to a unit (ASIC) and slice number. An example of this is shown here, where interface Ethernet2/16 of a Nexus 9504 switch with an N9K-X9736C-FX line card inserted in slot 2 of the chassis maps to Unit 1, Slice 0.

```
<#root>
```

```
switch#
```

show interface hardware-mappings

Legends:

- SMod - Source Mod. 0 is N/A
- Unit - Unit on which port resides. N/A for port channels
- HPort - Hardware Port Number or Hardware Trunk Id:
- HName - Hardware port name. None means N/A
- FPort - Fabric facing port number. 255 means N/A
- NPort - Front panel port number
- VPort - Virtual Port Number. -1 means N/A
- Slice - Slice Number. N/A for BCM systems
- SPort - Port Number wrt Slice. N/A for BCM systems
- SrcId - Source Id Number. N/A for BCM systems
- MacIdx - Mac index. N/A for BCM systems
- MacSubPort - Mac sub port. N/A for BCM systems

Name	Ifindex	Smod													
Unit															
HPort FPort NPort VPort															
Slice															
SPort	SrcId	MacId	MacSP	VIF	Block	BlkSrcID									
Eth2/1	1a080000	5	0	16	255	0	-1	0	16	32	4	0	145	0	32
Eth2/2	1a080200	5	0	12	255	4	-1	0	12	24	3	0	149	0	24
Eth2/3	1a080400	5	0	8	255	8	-1	0	8	16	2	0	153	0	16
Eth2/4	1a080600	5	0	4	255	12	-1	0	4	8	1	0	157	0	8
Eth2/5	1a080800	5	0	0	255	16	-1	0	0	0	0	0	161	0	0
Eth2/6	1a080a00	5	0	56	255	20	-1	0	56	112	14	0	165	1	40
Eth2/7	1a080c00	5	0	52	255	24	-1	0	52	104	13	0	169	1	32
Eth2/8	1a080e00	6	1	16	255	28	-1	0	16	32	4	0	173	0	32
Eth2/9	1a081000	6	1	12	255	32	-1	0	12	24	3	0	177	0	24
Eth2/10	1a081200	6	1	8	255	36	-1	0	8	16	2	0	181	0	16
Eth2/11	1a081400	6	1	4	255	40	-1	0	4	8	1	0	185	0	8
Eth2/12	1a081600	6	1	0	255	44	-1	0	0	0	0	0	189	0	0
Eth2/13	1a081800	6	1	56	255	48	-1	0	56	112	14	0	193	1	40
Eth2/14	1a081a00	6	1	52	255	52	-1	0	52	104	13	0	197	1	32
Eth2/15	1a081c00	7	2	16	255	56	-1	0	16	32	4	0	201	0	32
<b>Eth2/16</b>															
1a081e00 7															
2															
12 255 60 -1															
0															
12 24 3 0 205 0 24															
Eth2/17	1a082000	7	2	8	255	64	-1	0	8	16	2	0	209	0	16
Eth2/18	1a082200	7	2	4	255	68	-1	0	4	8	1	0	213	0	8
Eth2/19	1a082400	7	2	0	255	72	-1	0	0	0	0	0	217	0	0
Eth2/20	1a082600	7	2	56	255	76	-1	0	56	112	14	0	221	1	40
Eth2/21	1a082800	7	2	52	255	80	-1	0	52	104	13	0	225	1	32
Eth2/22	1a082a00	8	3	16	255	84	-1	0	16	32	4	0	229	0	32
Eth2/23	1a082c00	8	3	12	255	88	-1	0	12	24	3	0	233	0	24
Eth2/24	1a082e00	8	3	8	255	92	-1	0	8	16	2	0	237	0	16
Eth2/25	1a083000	8	3	4	255	96	-1	0	4	8	1	0	241	0	8
Eth2/26	1a083200	8	3	0	255	100	-1	0	0	0	0	0	245	0	0
Eth2/27	1a083400	8	3	56	255	104	-1	0	56	112	14	0	249	1	40

Eth2/28	1a083600	8	3	52	255	108	-1	0	52	104	13	0	253	1	32
Eth2/29	1a083800	5	0	48	255	112	-1	0	48	96	12	0	257	1	24
Eth2/30	1a083a00	5	0	44	255	116	-1	0	44	88	11	0	261	1	16
Eth2/31	1a083c00	6	1	48	255	120	-1	0	48	96	12	0	265	1	24
Eth2/32	1a083e00	6	1	44	255	124	-1	0	44	88	11	0	269	1	16
Eth2/33	1a084000	7	2	48	255	128	-1	0	48	96	12	0	273	1	24
Eth2/34	1a084200	7	2	44	255	132	-1	0	44	88	11	0	277	1	16
Eth2/35	1a084400	8	3	48	255	136	-1	0	48	96	12	0	281	1	24
Eth2/36	1a084600	8	3	44	255	140	-1	0	44	88	11	0	285	1	16

## Calculate Instance ID for Multiple ASIC and Slices

When interpreting the syslog, the instance ID is calculated based on the contiguous unit and slice combination order. For example, if a Nexus 9500 module or a Nexus 9300 TOR (Top-of-Rack) has two units (ASICs) and two slices per unit, the instance IDs can be as follows:

- Unit 0, slice 0 is referred to as Instance 0
- Unit 0, slice 1 is referred to as Instance 1
- Unit 1, slice 0 is referred to as Instance 2
- Unit 1, slice 1 is referred to as Instance 3

If a module has one unit and four slices, the Instance IDs can be:

- Unit 0, slice 0 is referred to as Instance 0
- Unit 0, slice 1 is referred to as Instance 1
- Unit 0, slice 2 is referred to as Instance 2
- Unit 0, slice 3 is referred to as Instance 3

## Understand Oversubscription and Output Discards

Interfaces attached to an Ethernet network are only able to transmit a single packet at a time. When two packets need to egress an Ethernet interface at the same time, the Ethernet interface transmits one packet while buffering the other packet. Once the first packet is transmitted, the Ethernet interface transmits the second packet from the buffer. When the total sum of traffic that needs to egress, an interface exceeds the interface bandwidth, the interface is considered to be *oversubscribed*. For example, if a total of 15Gbps of traffic instantaneously enters the switch and needs to egress a 10Gbps interface, the 10Gbps interface is oversubscribed because it is not able to transmit 15Gbps of traffic at a time.

A Cisco Nexus 9000 Series switch with a Cloud Scale ASIC handles this resource contention by buffering traffic within the buffers of the ASIC slice associated with the egress interface. If the total sum of traffic that needs to egress an interface exceeds the interface bandwidth for an extended period of time, the buffers of the ASIC slice begin to fill with packets that need to egress the interface.

When the buffers of the ASIC slice reach 90% utilization, the switch generates a syslog similar to one shown here:

```
%TAHUSD-SLOT2-4-BUFFER_THRESHOLD_EXCEEDED: Module 2 Instance 0 Pool-group buffer 90 percent threshold i
```

When the buffers of the ASIC slice become completely full, the switch drops any additional traffic that needs to egress the interface until space in the buffers becomes free. When the switch drops this traffic, the

switch increments the Output Discards counter on the egress interface.

The generated syslog and non-zero Output Discards counter are both symptoms of an oversubscribed interface. Each symptom is explored in more detail in the sub-sections here.

## Understand the BUFFER\_THRESHOLD\_EXCEEDED Syslog

An example of the BUFFER\_THRESHOLD\_EXCEEDED syslog is shown here.

```
%TAHUSD-SLOTX-4-BUFFER_THRESHOLD_EXCEEDED: Module X Instance Y Pool-group buffer Z percent threshold is
```

This syslog has three key pieces of information in it:

1. **Module X** - The slot of the line card that contains the oversubscribed interface.
2. **Instance Y** - The instance number assigned to the ASIC and slice tuple that contain the oversubscribed interface.
3. **Pool-group buffer Z** - The affected pool-group buffer threshold before the syslog is generated. This is a percentage derived by the Used Cells divided by the Total Cells as observed in the output of **show hardware internal buffer info pkt-stats** when attached to Module X.

## Understand the Output Discards Interface Counter

The Output Discards interface counter indicates the number of packets that were dropped that *must* have egressed the interface but were unable to due to the fact the ASIC slice buffer is full and unable to accept new packets. The Output Discards counter is visible in the output of **show interface** and **show interface counters errors** as shown here.

```
<#root>
```

```
switch#
```

```
show interface Ethernet1/1
```

```
Ethernet1/1 is up
admin state is up, Dedicated Interface
Hardware: 1000/10000/25000/40000/50000/100000 Ethernet, address: 7cad.4f6d.f6d8 (bia 7cad.4f6d.f6d8)
MTU 1500 bytes, BW 40000000 Kbit , DLY 10 usec
reliability 255/255, txload 232/255, rxload 1/255
Encapsulation ARPA, medium is broadcast
Port mode is trunk
full-duplex, 40 Gb/s, media type is 40G
Beacon is turned off
Auto-Negotiation is turned on FEC mode is Auto
Input flow-control is off, output flow-control is off
Auto-mdix is turned off
Rate mode is dedicated
Switchport monitor is off
EtherType is 0x8100
EEE (efficient-ethernet) : n/a
  admin fec state is auto, oper fec state is off
Last link flapped 03:16:50
Last clearing of "show interface" counters never
3 interface resets
Load-Interval #1: 30 seconds
```

```

30 seconds input rate 0 bits/sec, 0 packets/sec
30 seconds output rate 36503585488 bits/sec, 3033870 packets/sec
input rate 0 bps, 0 pps; output rate 36.50 Gbps, 3.03 Mpps
Load-Interval #2: 5 minute (300 seconds)
300 seconds input rate 32 bits/sec, 0 packets/sec
300 seconds output rate 39094683384 bits/sec, 3249159 packets/sec
input rate 32 bps, 0 pps; output rate 39.09 Gbps, 3.25 Mpps

```

```

RX
0 unicast packets 208 multicast packets 9 broadcast packets
217 input packets 50912 bytes
0 jumbo packets 0 storm suppression bytes
0 runs 0 giants 0 CRC 0 no buffer
0 input error 0 short frame 0 overrun 0 underrun 0 ignored
0 watchdog 0 bad etype drop 0 bad proto drop 0 if down drop
0 input with dribble 0 input discard
0 Rx pause

```

```

TX
38298127762 unicast packets 6118 multicast packets 0 broadcast packets
38298133880 output packets 57600384931480 bytes
0 jumbo packets
0 output error 0 collision 0 deferred 0 late collision
0 lost carrier 0 no carrier 0 babble

```

```
57443534227 output discard <<< Output discards due to oversubscription
```

```
0 Tx pause
```

```
switch#
```

```
show interface Ethernet1/1 counters errors
```

```

-----
Port          Align-Err  FCS-Err  Xmit-Err  Rcv-Err  UnderSize
-----
OutDiscards
-----
Eth1/1          0          0          0          0          0
57443534227

```

```

-----
Port          Single-Col  Multi-Col  Late-Col  Exces-Col  Carri-Sen  Runts
-----
Eth1/1          0          0          0          0          0          0

```

```

-----
Port          Giants  SQETest-Err  Deferred-Tx  IntMacTx-Er  IntMacRx-Er  Symbol-Err
-----
Eth1/1          0          --          0          0          0          0

```

```

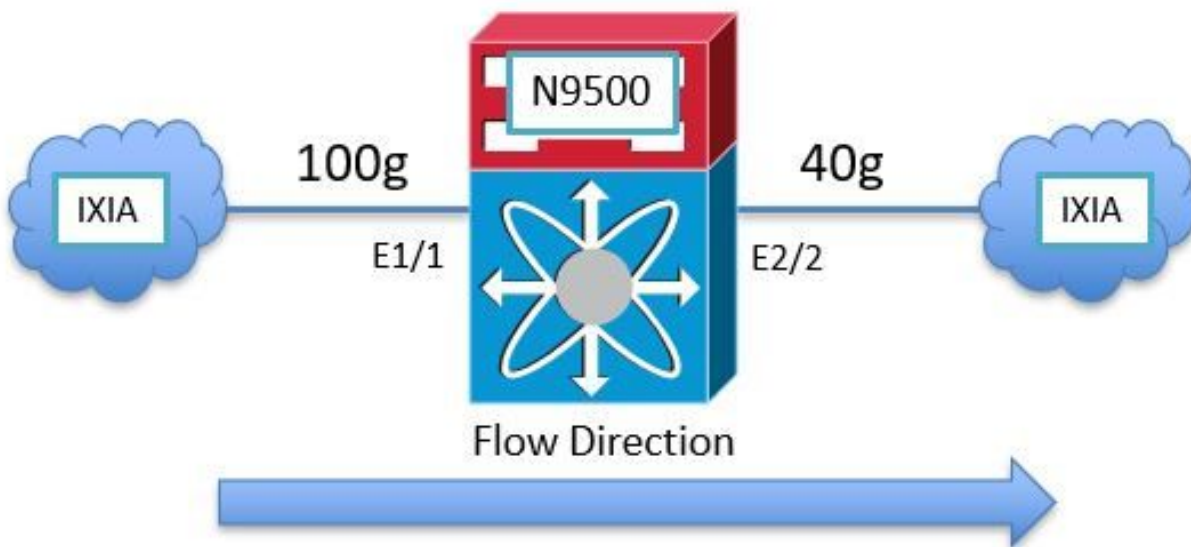
-----
Port          InDiscards
-----
Eth1/1          0

```

## Example Oversubscription Scenario



Consider a scenario, where traffic between two IXIA traffic generators traverse a Nexus 9504 switch with two N9K-X9736C-FX line cards inserted in slots 1 and 2 of the chassis. 100Gbps of traffic enters the switch through 100Gbps interface Ethernet1/1 and needs to egress 40Gbps interface Ethernet2/2. Therefore, Ethernet2/2 is oversubscribed. A topology of this scenario is shown here.



Since the Nexus 9000 Cloud Scale ASIC uses a shared-memory egress buffer architecture, you must check the buffer of the egress interface Ethernet2/2 to see the congestion. In this example, the line card inserted in slot 2 is the egress line card, so you must use the **attach module 2** command before view the internal hardware buffer with the **show hardware internal tah buffer counters** command. Notice the non-zero "Occupancy drops" counter for the Unit 0, Slice 0 pool-group and associated pools, which indicate the number of packets dropped because the pool-group buffer is fully occupied.

```
<#root>
```

```
switch#
```

```
attach module 2
```

```
module-2#
```

```
show hardware internal tah buffer counters
```

```
Unit: 0 Slice: 0
```

```
=====
```

```
|-----|
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|-----|
```

```
Occupancy drops                    51152554987
```

```
0          0          0          0          0          |
```



```

|
|           Pool 0      Pool 1      Pool 2      Pool 3      Pool 4      Pool 5      Pool
|-----|-----|-----|-----|-----|-----|-----|
Dynamic Threshold (cells)      93554      93554      93554      93554      93554      93554      9355
Occupancy drops                  0          0          0          0          0          0          0
AQM drops                        0          0          0          0          0          0          0
|-----|-----|-----|-----|-----|-----|-----|
|
|                                     Output MC Pool counters
|           Pool 0      Pool 1      Pool 2      Pool 3      Pool 4      Pool 5      Pool
|-----|-----|-----|-----|-----|-----|-----|
Dynamic Threshold (cells)      93554      93554      93554      93554      93554      93554      9355
Dynamic Threshold (desc)      93554      93554      93554      93554      93554      93554      9355
Dynamic Threshold (inq thr)    64035      64035      64035      64035      64035      64035      6403
Occupancy drops                  0          0          0          0          0          0          0
|-----+-----+-----+-----+-----+-----+-----+|
|                                     Additional counters
|-----+-----+-----+-----+-----+-----+-----+|
MEM cell drop reason           :      0
MEM descriptor drop reason     :      0
OPG cell drop reason           :      0
OPG descriptor drop reason     :      0
OPG CPU cell drop reason       :      0
OPG CPU descriptor drop reason :      0
OPG SPAN cell drop reason      :      0
OPG SPAN descriptor drop reason:      0
OPOOL cell drop reason         :      0
OPOOL descriptor drop reason   :      0
UC OQUEUE cell drop reason     :      0
MC OQUEUE cell drop reason     :      0
OQUEUE descriptor drop reason  :      0
MC OPOOL cell drop reason      :      0
FWD DROP                       :      8
SOD                            :      0
BMM BP                         :      0
No Drop                        :      0
Packets received               : 45981341
TRUNC MTU                      :      0
TRUNK BMM BP                   :      0
VOQFC messages sent           :      0
SOD messages sent              :      0
SPAN descptor drop            :      0

```

Each ASIC unit/slice tuple is represented through a unique identified called an "instance". The output of the **show hardware internal buffer info pkt-stats** command displays detailed information about the congested pool-group (abbreviated as "PG") for each instance. The command also shows the historical peak/maximum number of cells in the buffer that have been used. Finally, the command shows an instantaneous snapshot of the Cloud Scale ASIC port identifiers of ports with traffic that is buffered. An example of this command is shown here.

```
<#root>
```

```
switch#
```

```
attach module 2
```



```
|ASIC Port      Q7      Q6      Q5      Q4      Q3      Q2      Q1      Q0      |
|-----+-----+-----+-----+-----+-----+-----+-----+|
```

```
[12] <<< A

      UC->      0      0      0      0      0      0      0      59988  |
MC cells->    0      0      0      0      0      0      0      0      |
MC desc->    0      0      0      0      0      0      0      0      |
```

Also see the **peak** variation of the command. Use this command to associate the syslog with a potential spike in a particular pool-group, pool, or port.

```
<#root>
```

```
switch# show hardware internal buffer info pkt-stats peak
```

```
slot 1
```

```
=====
```

```
Instance 0
```

```
=====
```

```
|-----+-----+-----+-----+-----+|
|                               Pool-Group Peak counters                               |
|-----+-----+-----+-----+-----+|
Drop PG      :      0
No-drop PG   :      0
```

```
|-----+-----+-----+-----+-----+|
|                               Pool Peak counters                               |
|-----+-----+-----+-----+-----+|
MC Pool 0    :      0
MC Pool 1    :      0
MC Pool 2    :      0
MC Pool 3    :      0
MC Pool 4    :      0
MC Pool 5    :      0
MC Pool 6    :      0
MC Pool 7    :      0

UC Pool 0    :      0
UC Pool 1    :      0
UC Pool 2    :      0
UC Pool 3    :      0
UC Pool 4    :      0
UC Pool 5    :      0
UC Pool 6    :      0
UC Pool 7    :      0
```

```
|-----+-----+-----+-----+-----+|
|                               Port Peak counters                               |
| classes mapped to count_0: 0 1 2 3 4 5 6 7 |
| classes mapped to count_1: None          |
|-----+-----+-----+-----+-----+|
```

```

[0]                                     <<< ASIC Port. This can be checked via "show hardware i
      count_0       :           0
      count_1       :           0
[1]
      count_0       :           0
      count_1       :           0

```

The **show interface hardware-mappings** command can be used to translate the Cloud Scale ASIC port identifier to a front-panel port. In the aforementioned example, ASIC port 12 (represented by the SPort column in the output of show interface hardware-mappings) associated with ASIC Unit 0 on Slice/Instance 0 has 59,988 occupied cells of 416 bytes each. An example of the show interface hardware-mappings command is shown here, which maps this interface to front-panel port Ethernet2/2.

```
<#root>
```

```
switch#
```

```
show interface hardware-mappings
```

```
Legends:
```

```

SMod  - Source Mod. 0 is N/A
Unit  - Unit on which port resides. N/A for port channels
HPort - Hardware Port Number or Hardware Trunk Id:
HName - Hardware port name. None means N/A
FPort - Fabric facing port number. 255 means N/A
NPort - Front panel port number
VPort - Virtual Port Number. -1 means N/A
Slice - Slice Number. N/A for BCM systems
SPort - Port Number wrt Slice. N/A for BCM systems
SrcId - Source Id Number. N/A for BCM systems
MacIdx - Mac index. N/A for BCM systems
MacSubPort - Mac sub port. N/A for BCM systems

```

```
-----
```

```
Name
```

```
    Ifindex  Smod
```

```
Unit
```

```
HPortFPort NPort VPort
```

```
Slice
```

```
SPort
```

```
    SrcId MacId MacSP VIF  Block BlkSrcID
```

```
-----
```

```
Eth2/2
```

```
    1a080200 5
```

```
0
```

```
12  255  4  -1
```

```
0
```

12

24 3 0 149 0 24

We can further correlate the oversubscription of interface Ethernet2/2 with QoS queueing drops with the **show queuing interface** command. An example of this is shown here.

<#root>

switch#

show queuing interface Ethernet2/2

Egress Queuing for Ethernet2/2 [System]

QoS-Group#	Bandwidth%	PrioLevel	Min	Shape Max	Units	QLimit
7	-	1	-	-	-	9(D)
6	0	-	-	-	-	9(D)
5	0	-	-	-	-	9(D)
4	0	-	-	-	-	9(D)
3	0	-	-	-	-	9(D)
2	0	-	-	-	-	9(D)
1	0	-	-	-	-	9(D)
0	100	-	-	-	-	9(D)

QOS GROUP 0	
Unicast	Multicast
Tx Pkts   35593332351	18407162
Tx Byts   53532371857088	27684371648

WRED/AFD & Tail Drop Pkts

53390604466

27573307

WRED/AFD & Tail Drop Byts

80299469116864

110293228

|  
| Q Depth Byts |

24961664

| 0 |  
|

WD & Tail Drop Pkts

|  
53390604466

|  
27573307

QOS GROUP 1		
	Unicast	Multicast
Tx Pkts	0	0
Tx Byts	0	0
WRED/AFD & Tail Drop Pkts	0	0
WRED/AFD & Tail Drop Byts	0	0
Q Depth Byts	0	0
WD & Tail Drop Pkts	0	0

QOS GROUP 2		
	Unicast	Multicast
Tx Pkts	0	0
Tx Byts	0	0
WRED/AFD & Tail Drop Pkts	0	0
WRED/AFD & Tail Drop Byts	0	0
Q Depth Byts	0	0
WD & Tail Drop Pkts	0	0

QOS GROUP 3		
	Unicast	Multicast
Tx Pkts	0	0
Tx Byts	0	0
WRED/AFD & Tail Drop Pkts	0	0
WRED/AFD & Tail Drop Byts	0	0
Q Depth Byts	0	0
WD & Tail Drop Pkts	0	0

QOS GROUP 4		
	Unicast	Multicast
Tx Pkts	0	0
Tx Byts	0	0
WRED/AFD & Tail Drop Pkts	0	0
WRED/AFD & Tail Drop Byts	0	0
Q Depth Byts	0	0
WD & Tail Drop Pkts	0	0



QOS GROUP 5			
	Unicast	Multicast	
Tx Pkts	0	0	
Tx Byts	0	0	
WRED/AFD & Tail Drop Pkts	0	0	
WRED/AFD & Tail Drop Byts	0	0	
Q Depth Byts	0	0	
WD & Tail Drop Pkts	0	0	
QOS GROUP 6			
	Unicast	Multicast	
Tx Pkts	0	0	
Tx Byts	0	0	
WRED/AFD & Tail Drop Pkts	0	0	
WRED/AFD & Tail Drop Byts	0	0	
Q Depth Byts	0	0	
WD & Tail Drop Pkts	0	0	
QOS GROUP 7			
	Unicast	Multicast	
Tx Pkts	0	0	
Tx Byts	0	0	
WRED/AFD & Tail Drop Pkts	0	0	
WRED/AFD & Tail Drop Byts	0	0	
Q Depth Byts	0	0	
WD & Tail Drop Pkts	0	0	
CONTROL QOS GROUP			
	Unicast	Multicast	
Tx Pkts	5704	0	
Tx Byts	725030	0	
Tail Drop Pkts	0	0	
Tail Drop Byts	0	0	
SPAN QOS GROUP			
	Unicast	Multicast	
Tx Pkts	0	0	
Tx Byts	0	0	

#### Per Slice Egress SPAN Statistics

SPAN Copies Tail Drop Pkts	0
SPAN Input Queue Drop Pkts	0
SPAN Copies/Transit Tail Drop Pkts	0
SPAN Input Desc Drop Pkts	0

Finally, you can verify that egress interface Ethernet2/2 has a non-zero output discard counter with the **show interface** command. An example of this is shown here.

<#root>

switch#

show interface Ethernet2/2

```
Ethernet2/2 is up
admin state is up, Dedicated Interface
Hardware: 1000/10000/25000/40000/50000/100000 Ethernet, address: 7cad.4f6d.f6d8 (bia 7cad.4f6d.f6d8)
MTU 1500 bytes, BW 40000000 Kbit , DLY 10 usec
reliability 255/255, txload 232/255, rxload 1/255
Encapsulation ARPA, medium is broadcast
Port mode is trunk
full-duplex, 40 Gb/s, media type is 40G
Beacon is turned off
Auto-Negotiation is turned on FEC mode is Auto
Input flow-control is off, output flow-control is off
Auto-mdix is turned off
Rate mode is dedicated
Switchport monitor is off
EtherType is 0x8100
EEE (efficient-ethernet) : n/a
  admin fec state is auto, oper fec state is off
Last link flapped 03:16:50
Last clearing of "show interface" counters never
3 interface resets
Load-Interval #1: 30 seconds
  30 seconds input rate 0 bits/sec, 0 packets/sec
  30 seconds output rate 36503585488 bits/sec, 3033870 packets/sec
  input rate 0 bps, 0 pps; output rate 36.50 Gbps, 3.03 Mpps
Load-Interval #2: 5 minute (300 seconds)
  300 seconds input rate 32 bits/sec, 0 packets/sec
  300 seconds output rate 39094683384 bits/sec, 3249159 packets/sec
  input rate 32 bps, 0 pps; output rate 39.09 Gbps, 3.25 Mpps
RX
  0 unicast packets  208 multicast packets  9 broadcast packets
  217 input packets  50912 bytes
  0 jumbo packets  0 storm suppression bytes
  0 runts  0 giants  0 CRC  0 no buffer
  0 input error  0 short frame  0 overrun  0 underrun  0 ignored
  0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
  0 input with dribble  0 input discard
  0 Rx pause
TX
  38298127762 unicast packets  6118 multicast packets  0 broadcast packets
  38298133880 output packets  57600384931480 bytes
  0 jumbo packets
  0 output error  0 collision  0 deferred  0 late collision
  0 lost carrier  0 no carrier  0 babble

57443534227 output discard

<<< Output discards due to oversubscription

  0 Tx pause
```

## Next Steps

If you observe output discards on a Nexus 9000 series switch with a Cloud Scale ASIC, you can resolve the

issue with one or more of the methods here:

- If the interface that experiences output discards is a single interface and is not bundled into a port-channel, then you can upgrade the bandwidth of the interface to help alleviate congestion. For example, if a congested egress interface is a 10Gbps interface, then if you upgrade to a 25Gbps, 40Gbps interface, or 100Gbps interface it can help resolve the issue.
  - Based on the transceiver form factor of the egress interface, you can upgrade the transceiver (you can migrate from a 10Gbps SFP+ inserted in a CVR-QSFP-SFP10G inside a QSFP port to a native 40Gbps QSFP transceiver).
  - This can also be done if you migrate the congested egress interface configuration from a 10Gbps port to a 25Gbps, 40Gbps, or 100Gbps port.
- If the interface that experiences output discards is a single interface and is not bundled into a port-channel, then you can configure the congested interface to be a member of a port-channel alongside another interface of the same bandwidth and alleviate congestion.
- If the interface that experiences output discards is a port-channel interface, then if you add members to the port-channel it can increase the bandwidth of the overall port-channel and improve load-balanced hashing for multiple large traffic flow.
- Validate whether congested traffic flows between hosts in your network involve interfaces that step down in speed (for example, traffic that ingresses a switch through a 40Gbps interface and egresses a switch through a 10Gbps interface). This can be a bottleneck that causes network congestion. To eliminate this bottleneck, upgrade the lower-speed interface (for example, 10Gbps) to a higher-speed interface (for example, 25Gbps, 40Gbps, and so on) and this alleviates network congestion.
- If you cannot increase available bandwidth on the congested egress interface, validate [End-to-End QoS](#) and apply appropriate queuing actions for your network.
- If micro-bursts are a potential cause for intermittent congestion, reference the [Monitoring Micro-Bursts](#) section of this document for information on how to configure micro-burst monitoring.

## Additional Information

This section of the document contains additional information about the next steps to take when you encounter the `BUFFER_THRESHOLD_EXCEEDED` syslog, network congestion/oversubscription scenarios, and increment output discard interface counters.

### `BUFFER_THRESHOLD_EXCEEDED` Syslog Configuration Options

You can modify the system buffer status polling interval, which controls how often the system polls the current utilization of ASIC slice buffers. This is done with the **hardware profile buffer info poll-interval** global configuration command. The default configuration value is 5,000 milliseconds. This configuration can be modified globally or on a per-module basis. An example of this configuration command is shown here, where it is modified to a value of 1,000 milliseconds.

```
<#root>
```

```
switch#
```

```
configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
switch(config)#
```

```
hardware profile buffer info poll-interval timer 1000
```

```
switch(config)#
```

```
end
```

```
switch#  
  
show running-config | include hardware.profile.buffer  
  
hardware profile buffer info poll-interval timer 1000  
switch#
```

You can modify the port egress buffer usage threshold value, which controls when the system generates the `BUFFER_THRESHOLD_EXCEEDED` syslog indicates that indicates ASIC slice buffer utilization has exceeded the configured threshold. This is done with the **hardware profile buffer info port-threshold** global configuration command. The default configuration value is 90%. This configuration can be modified globally or on a per-module basis. An example of this configuration command is shown here, where it is modified to a value of 80%.

```
<#root>  
  
switch#  
  
configure terminal  
  
Enter configuration commands, one per line. End with CNTL/Z.  
switch(config)#  
  
hardware profile buffer info port-threshold threshold 80  
  
switch(config)#  
  
end  
  
switch#  
  
show running-config | include hardware.profile.buffer  
  
hardware profile buffer info port-threshold threshold 80  
switch#
```

You can modify the minimum interval in between `BUFFER_THRESHOLD_EXCEEDED` syslogs generated by the switch. You can also outright disable the `BUFFER_THRESHOLD_EXCEEDED` syslog. This is done with the **hardware profile buffer info syslog-interval timer** global configuration command. The default configuration value is 120 seconds. The syslog can be disabled entirely by setting the value to 0 seconds. An example of this configuration command is shown here, where the syslog is disabled entirely.

```
<#root>  
  
switch#  
  
configure terminal  
  
Enter configuration commands, one per line. End with CNTL/Z.  
switch(config)#  
  
hardware profile buffer info syslog-interval timer 0  
  
switch(config)#  
  
end  
  
switch#
```

```
show running-config | include hardware.profile.buffer
```

```
hardware profile buffer info syslog-interval timer 0  
switch#
```

## Logs to Collect for Network Congestion Scenarios

You can collect the logs shown here from a switch affected by a network congestion scenario to identify a congested egress interface in addition to the commands listed in this document.

1. The output of the **show tech-support details** command.
2. The output of the **show tech-support usd-all** command.
3. The output of the **show tech-support ipqos all** command.
4. When you work with a Nexus 9500 series switch with Cisco Cloud Scale line cards inserted, the output of the **show system internal interface counters peak module {x}** command, where {x} is the slot number of the module hosting the congested egress interface.

## Monitoring Micro-Bursts

When congestion or oversubscription happens in very short intervals (a micro-burst), additional information is needed to get an accurate depiction of how the oversubscription affects the switch.

Cisco Nexus 9000 Series switches equipped with the Cisco Cloud Scale ASIC can monitor traffic for micro-bursts that can cause temporary network congestion and traffic loss in your environment. For more information about micro-bursts and how to configure this feature, consult the documents shown here:

- ["Micro-Burst Monitoring" chapter of the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 10.1\(x\)](#)
- ["Micro-Burst Monitoring" chapter of the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.3\(x\)](#)
- ["Micro-Burst Monitoring" chapter of the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2\(x\)](#)
- ["Micro-Burst Monitoring" chapter of the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 7.x](#)

## Related Information

- [Cisco Technical Support & Downloads](#)