

This article below provides only basic information on how to troubleshoot unicast packet flow traffic issues for the M1 Series modules.

Troubleshooting L2/L3 unicast is covered in detail in Cisco-Live presentation. Sections of this presentation covers, both platform independent, and platform specific step by step troubleshooting for unicast, among other things. Access to this presentation is available FREE. Follow the below instructions to access the presentation

1. Visit <https://www.ciscolivevirtual.com/>
2. Register for free.
3. Click on "Cisco Live Virtual" link.
4. Click on the ?Sessions? Tab on top, and select ?2011 Sessions Catalog?
5. In the search box, type ?BRKCRS-3144? and Submit search.
6. Select the session. You can either View the Session (or) download the pdf.
7. Unicast troubleshooting is covered from slides 81 through 93.

Guide Contents
Troubleshooting Overview
Troubleshooting Installs, Upgrades, and Reboots
Troubleshooting Licensing'
Troubleshooting VDCs
Troubleshooting CFS
Troubleshooting Ports
Troubleshooting vPCs
Troubleshooting VLANs
Troubleshooting STP
Troubleshooting Routing
Troubleshooting Unicast Traffic(<i>this section</i>)
Troubleshooting WCCP
Troubleshooting Memory
Troubleshooting Packet Flow Issues
Troubleshooting FCoE
Before Contacting Technical Support
Troubleshooting Tools and Methodology

Contents

- [1 Packet is Received into Interface from Wire](#)
- [2 Linksec Decryption Occurs, 1st stage Port QoS](#)
- [3 Second Stage Port QoS Occurs](#)
- [4 Layer 2 Source/Destination MAC Processing](#)
- [5 Layer 3 Engine Processing](#)
 - ◆ [5.1 Layer 3 Engine Processes Layer 3 Features](#)

◆ 5.2 Layer 3 forwarding for Routed Traffic

- 6 SFabric Processing Occurs (optional)
- 7 Layer 2 Engine Performs Source/Destination MAC Processing
- 8 Egress Port QoS is Performed
- 9 Linksec Encryption Occurs
- 10 Packet is Transmitted

Packet is Received into Interface from Wire

During this step, the packet is received into the Nexus 7000 port. When troubleshooting this step, we want to look to ensure there is transceiver interoperability, and validate whether we are seeing any errors on the interface. We do this via using the following commands

- **show interface *interface***
- **show interface *interface* transceiver**

PHX2-N7K-1# **show interface e1/1**

Ethernet1/1 is up

```
Hardware: 10000 Ethernet, address: 0024.986c.00b0 (bia 0024.986c.00b0)
Description: N7K-vdc-1 connecting to core 6506
MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA
Port mode is trunk
full-duplex, 10 Gb/s, media type is 10g
Beacon is turned off
Auto-Negotiation is turned off
Input flow-control is off, output flow-control is off
Rate mode is shared
Switchport monitor is off
Last link flapped 7week(s) 4day(s)
Last clearing of "show interface" counters never
1 minute input rate 13056 bits/sec, 9 packets/sec
1 minute output rate 4608 bits/sec, 0 packets/sec
Rx
 341190251 input packets 276211313 unicast packets 52112947 multicast packets
 12865991 broadcast packets 0 jumbo packets 0 storm suppression packets
 94295027129 bytes
Tx
 462437316 output packets 85121 multicast packets
 188251 broadcast packets 0 jumbo packets
 648159081064 bytes
0 input error 0 short frame 0 watchdog
 0 no buffer 0 runt 0 CRC 0 ecc
 0 overrun 0 underrun 0 ignored 0 bad etype drop
 0 bad proto drop 0 if down drop 0 input with dribble
 0 input discard
 0 output error 0 collision 0 deferred
 0 late collision 0 lost carrier 0 no carrier
 0 babble
 0 Rx pause 0 Tx pause
1 interface resets
```

PHX2-N7K-1# **show interface e1/1 transceiver details**

```

Ethernet1/1
 sfp is present
 name is CISCO-AVAGO      <<< If this says type is (unknown), it is not supported.
 part number is SFBR-7700SDZ
 revision is B4
 serial number is AGD12434116
 nominal bitrate is 10300 Mbits/sec
 Link length supported for 50/125um fiber is 82 m(s)
 Link length supported for 62.5/125um fiber is 26 m(s)
 cisco id is --
 cisco extended id number is 4

```

SFP Detail Diagnostics Information (internal calibration)

		Alarms		Warnings	
		High	Low	High	Low
Temperature	45.46 C	75.00 C	-5.00 C	70.00 C	0.00 C
Voltage	3.28 V	3.63 V	2.97 V	3.46 V	3.13 V
Current	6.92 mA	10.50 mA	2.50 mA	10.50 mA	2.50 mA
Tx Power	-2.75 dBm	1.69 dBm	-11.30 dBm	-1.30 dBm	-7.30 dBm
Rx Power	-2.43 dBm	1.99 dBm	-13.97 dBm	-1.00 dBm	-9.91 dBm
Transmit Fault Count = 0					

Note: ++ high-alarm; + high-warning; -- low-alarm; - low-warning

Linksec Decryption Occurs, 1st stage Port QoS

In step 2, Linksec decryption occurs as well as receive side stage 1 QoS.

It is important to step back and evaluate the difference between stage 1 and stage 2 QoS. The difference is that some ports can be configured in shared mode, whereas some can be configured in dedicated mode, on the 10G modules. What this means, is that there is 10g of bandwidth that can be dedicated to a port or shared amongst ports (4 ports, on the m132 module).

When running in shared mode, there exists a chance for contention accessing the 10g bandwidth through the 4:1 Mux. To alleviate this, some QoS intelligence was passed down to the 4:1 Mux which aggregates the ports.

In dedicated mode, there is no QoS applied at the Mux, instead, all traffic is processed in phase 2 QoS. To summarize, in shared mode, 1st stage QoS ensures fair access to the 10g of port bandwidth. In both shared and dedicated mode, 2nd stage QoS occurs to provide ingress queuing to the system.

For the ingress QoS, we are concerned about the Receive side QoS parameters in the show queuing command.

Use the **show policy-map** command to see per queue dropped packets.

The commands to troubleshoot Linksec and Port QoS are as follows:

- **show cts interface** [*all* | *interface*]
- **show queuing interface** *interface*
- **show policy-map interface** (for per queue drop)

switch# **show cts interface all Working Example**

```

CTS Information for Interface Ethernet2/24:
CTS is enabled, mode:      CTS_MODE_DOT1X
IFC state:                 CTS_IFC_ST_CTS_OPEN_STATE
Authentication Status:    CTS_AUTHC_SUCCESS
  Peer Identity:          india1
  Peer is:                CTS Capable
  802.1X role:           CTS_ROLE_AUTH
  Last Re-Authentication:
Authorization Status:     CTS_AUTHZ_SUCCESS
  PEER SGT:               2
  Peer SGT assignment:    Trusted
  Global policy fallback access list:
SAP Status:                CTS_SAP_SUCCESS
  Configured pairwise ciphers: GCM_ENCRYPT
  Replay protection:      Enabled
  Replay protection mode: Strict
  Selected cipher:        GCM_ENCRYPT
  Current receive SPI:    sci:1b54c1fbff0000 an:0
  Current transmit SPI:   sci:1b54c1fc000000 an:0
    
```

PHX2-N7K-1# show cts interface eth 1/8 Broken Example

```

CTS Information for Interface Ethernet1/8:
CTS is enabled, mode:      CTS_MODE_MANUAL
IFC state:                 Unknown
Authentication Status:    CTS_AUTHC_INIT
  Peer Identity:
  Peer is:                 Not CTS Capable
  802.1X role:            CTS_ROLE_UNKNOWN
  Last Re-Authentication:
Authorization Status:     CTS_AUTHZ_INIT
  PEER SGT:               0
  Peer SGT assignment:    Not Trusted
SAP Status:                CTS_SAP_INIT
  Configured pairwise ciphers:
  Replay protection:
  Replay protection mode:
  Selected cipher:
  Current receive SPI:
  Current transmit SPI:
    
```

PHX2-N7K-1# show queuing int eth 1/1

```

Interface Ethernet1/1 TX Queuing strategy: Weighted Round-Robin
Port QoS is enabled
Queuing Mode in TX direction: mode-cos
Transmit queues [type = 1p7q4t]
    
```

Queue Id	Scheduling	Num of thresholds
1p7q4t-out-q-default	WRR	04
1p7q4t-out-q2	WRR	04
1p7q4t-out-q3	WRR	04
1p7q4t-out-q4	WRR	04
1p7q4t-out-q5	WRR	04
1p7q4t-out-q6	WRR	04
1p7q4t-out-q7	WRR	04
1p7q4t-out-pq1	Priority	04

Configured WRR

Cisco_Nexus_7000_Series_NX-OS_Troubleshooting_Guide_-_Troubleshooting_Unicast_Traffic

```
WRR bandwidth ratios: 25[1p7q4t-out-q-default] 15[1p7q4t-out-q2] 12[1p7q4t-out-q3]
12[1p7q4t-out-q4] 12[1p7q4t-out-q5] 12[1p7q4t-out-q6] 12[1p7q4t-out-q7]
```

WRR configuration read from HW

```
WRR bandwidth ratios: 25[1p7q4t-out-q-default] 15[1p7q4t-out-q2] 11[1p7q4t-out-q3]
11[1p7q4t-out-q4] 11[1p7q4t-out-q5] 11[1p7q4t-out-q6] 11[1p7q4t-out-q7]
```

Configured queue-limit ratios

```
queue-limit ratios: 78[1p7q4t-out-q-default] 1[1p7q4t-out-q2] 1[1p7q4t-out-q3]
*1[1p7q4t-out-q4] *1[1p7q4t-out-q5] *1[1p7q4t-out-q6] *1[1p7q4t-out-q7] 16[1p7q4t-out-pq1]
* means unused queue with mandatory minimum queue-limit
```

queue-limit ratios configuration read from HW

```
queue-limit ratios: 78[1p7q4t-out-q-default] 1[1p7q4t-out-q2] 1[1p7q4t-out-q3]
*1[1p7q4t-out-q4] *1[1p7q4t-out-q5] *1[1p7q4t-out-q6] *1[1p7q4t-out-q7] 16[1p7q4t-out-pq1]
* means unused queue with mandatory minimum queue-limit
```

Thresholds:

COS	Queue	Threshold Type	Min	Max
0	1p7q4t-out-q-default	DT	100	100
1	1p7q4t-out-q-default	DT	100	100
2	1p7q4t-out-q-default	DT	100	100
3	1p7q4t-out-q-default	DT	100	100
4	1p7q4t-out-q-default	DT	100	100
5	1p7q4t-out-pq1	DT	100	100
6	1p7q4t-out-pq1	DT	100	100
7	1p7q4t-out-pq1	DT	100	100

Interface Ethernet1/1 RX Queuing strategy: Weighted Round-Robin

Queuing Mode in RX direction: mode-cos

Receive queues [type = 8q2t]

Port Cos not configured

Queue Id	Scheduling	Num of thresholds
8q2t-in-q-default	WRR	02
8q2t-in-q2	WRR	02
8q2t-in-q3	WRR	02
8q2t-in-q4	WRR	02
8q2t-in-q5	WRR	02
8q2t-in-q6	WRR	02
8q2t-in-q7	WRR	02
8q2t-in-q1	WRR	02

Configured WRR

```
WRR bandwidth ratios: 20[8q2t-in-q-default] 0[8q2t-in-q2] 0[8q2t-in-q3]
0[8q2t-in-q4] 0[8q2t-in-q5] 0[8q2t-in-q6] 0[8q2t-in-q7] 80[8q2t-in-q1]
```

WRR configuration read from HW

```
WRR bandwidth ratios: 20[8q2t-in-q-default] 0[8q2t-in-q2] 0[8q2t-in-q3]
0[8q2t-in-q4] 0[8q2t-in-q5] 0[8q2t-in-q6] 0[8q2t-in-q7] 80[8q2t-in-q1]
```

No queue-limit ratios user configuration

queue-limit ratios configuration read from HW

```
queue-limit ratios: 100[8q2t-in-q-default] 100[8q2t-in-q2] 100[8q2t-in-q3]
100[8q2t-in-q4] 100[8q2t-in-q5] 100[8q2t-in-q6] 100[8q2t-in-q7] 100[8q2t-in-q1]
```

Thresholds:

COS	Queue	Threshold Type	Min	Max
0	8q2t-in-q-default	DT	100	100
1	8q2t-in-q-default	DT	100	100
2	8q2t-in-q-default	DT	100	100
3	8q2t-in-q-default	DT	100	100

4	8q2t-in-q-default	DT	100	100
5	8q2t-in-q1	DT	100	100
6	8q2t-in-q1	DT	100	100
7	8q2t-in-q1	DT	100	100

PHX2-N7K-1# show policy-map interface eth 1/2

```
Global statistics status :   enabled
Ethernet1/2
Service-policy (queuing) input:   default-in-policy
  policy statistics status:   enabled

Class-map (queuing):   in-q1 (match-any)
  queue-limit percent 50
  bandwidth percent 80
  queue dropped pkts : 0

Class-map (queuing):   in-q-default (match-any)
  queue-limit percent 50
  bandwidth percent 20
  queue dropped pkts : 0

Service-policy (queuing) output:   default-out-policy
  policy statistics status:   enabled

Class-map (queuing):   out-pq1 (match-any)
  priority level 1
  queue-limit percent 16
  queue dropped pkts : 0

Class-map (queuing):   out-q2 (match-any)
  queue-limit percent 1
  queue dropped pkts : 0

Class-map (queuing):   out-q3 (match-any)
  queue-limit percent 1
  queue dropped pkts : 0

Class-map (queuing):   out-q-default (match-any)
  queue-limit percent 82
  bandwidth remaining percent 25
  queue dropped pkts : 0
```

Second Stage Port QoS Occurs

For the ingress QoS, we are concerned about the Receive side QoS parameters in the show queuing command.

Use the **show policy-map** command to view queue drops .

The commands to troubleshoot Port QoS are

- **show queuing interface *interface***
- **show policy-map interface**

Layer 2 Source/Destination MAC Processing

In this step, the ASIC submits the packet headers to the Layer 2 engine for lookup, and the Layer 2 engine performs source/destination MAC processing.

To validate forwarding of the Layer 2 engine, we should first look at the centralized mac table aggregated on the supervisor to validate whether the mac addresses are correlated as we expect them, and assigned to the ports where we expect the Mac's to reside.

Based off of this, we can then validate the hardware programming on the ingress linecard to validate that our mac address table is properly programmed into the hardware based Layer 2 engine on the linecard.


We first will look at the mac address table, then we can ensure programming is properly occurring in the hardware table.

The commands used to accomplish this are as follows:

- **show mac address-table**
- **show hardware mac address-table module interface interface**

To drill down on a specific MAC address, we can use the grep function with these commands to validate the mac is associated with a particular port, and that the hardware programming reflects that.

- **show mac address-table | grep macaddress**
- **show hardware mac address-table module interface interface | grep macaddress**

 **Note:** When evaluating the Hardware mac table, if the Index is set to 0x00400, or the GM bit is set to ?1?, that traffic will be routed. For example, you will see the index set to 0x00400 and GM bit set to 1 for traffic destined to the mac address local to the device

PHX2-N7K-1# **show mac address-table**

Legend:

* - primary entry, G - Gateway MAC, (R) - Routed MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link

VLAN	MAC Address	Type	age	Secure	NTFY	Ports
G -	0023.ac67.dd41	static	-	False	False	sup-eth1 (R)
G 5	0023.ac67.dd41	static	-	False	False	sup-eth1 (R)
* 5	0000.0c07.ac01	dynamic	0	False	False	Eth1/1
* 5	000c.2943.a67e	dynamic	180	False	False	Eth1/1
* 5	000c.294b.c5ca	dynamic	0	False	False	Eth1/1
* 5	000d.ece2.0640	dynamic	180	False	False	Eth1/1
* 5	0013.5f32.aa80	dynamic	0	False	False	Eth1/1
* 5	0018.8b45.41b7	dynamic	0	False	False	Eth1/1
* 5	0019.bb2f.4871	dynamic	0	False	False	Eth1/1
* 5	0019.bbe5.f3b8	dynamic	1230	False	False	Eth1/1
* 5	001a.4b33.ccdc	dynamic	1080	False	False	Eth1/1
* 5	001a.4ba8.6a9c	dynamic	1680	False	False	Eth1/1
* 5	001b.210a.87f9	dynamic	600	False	False	Eth1/1
* 5	001b.d46f.70e0	dynamic	60	False	False	Eth1/1
* 5	001c.c4e5.ac9a	dynamic	150	False	False	Eth1/1
* 5	0023.ac64.6f7c	dynamic	1230	False	False	Eth1/1
* 5	0024.986d.21c8	dynamic	270	False	False	Eth1/1

PHX2-N7K-1# **show hardware mac address-table 1 int eth1/1**

Valid	PI	BD	MAC	Index	Stat	SW	Modi	Age	Tmr	GM	Sec
TR	NT	RM	RMA	Cap	Fld	Always					Learn
	ic		fied	Byte	Sel		ure	AP	FY		TURE
1	1	2	000c.294b.c5ca	0x00422	0	3	0	67	1	0	
0	0	0	0	0	0	0	0	0	0	0	
1	1	2	0050.567e.58e6	0x00422	0	3	0	68	1	0	
0	0	0	0	0	0	0	0	0	0	0	
1	1	2	0050.56aa.6067	0x00422	0	3	0	67	1	0	
0	0	0	0	0	0	0	0	0	0	0	
1	1	2	00c0.b72e.cfa0	0x00422	0	3	0	67	1	0	
0	0	0	0	0	0	0	0	0	0	0	
1	1	2	0018.8b45.41b7	0x00422	0	3	0	68	1	0	
0	0	0	0	0	0	0	0	0	0	0	
1	1	2	0013.5f32.aa80	0x00422	0	3	0	68	1	0	
0	0	0	0	0	0	0	0	0	0	0	
1	1	2	0050.56aa.75ca	0x00422	0	3	0	64	1	0	
0	0	0	0	0	0	0	0	0	0	0	
1	1	2	00a0.9811.a233	0x00422	0	3	0	39	1	0	
0	0	0	0	0	0	0	0	0	0	0	

PHX2-N7K-1# show mac address-table | grep 000c.294b.c5ca

```
* 5          000c.294b.c5ca    dynamic    150          False  False  Eth1/1
```

PHX2-N7K-1# show hardware mac address-table 1 int eth 1/1 | grep 000c.294b.c5ca

```
1          1          2          000c.294b.c5ca  0x00422    0          3          0          67          1          0
0          0          0          0          0          0          0          0          0          0          0
```

Layer 3 Engine Processing

After the Layer 2 engine is finished, it sends the header to the Layer 3 engine. The layer 3 engine applies layer 3 intelligent features, to all packets, and layer 3 forwarding, to routed packets. As such, this section will divide the troubleshooting into two components, the layer 3 features applied to all packets, and the layer 3 forwarding.

The layer 3 features which are applied to all packets include the below features :

- 1) ACL
- 2) QoS
- 3) Netflow
- 4) Hardware IPS

Following the evaluation of the features, we will evaluate the layer 3 forwarding troubleshooting.

Layer 3 Engine Processes Layer 3 Features

We'll drill through each of the layer 3 features below, looking at one feature at a time.

The first feature we will look at is ACL. To troubleshoot ACL, we want to evaluate the configuration, and any relevant hit counters. We then can identify if the hardware on the linecard is programming the ACL.

It is important to note, that if you wish to see per ACL counters, you must enable ?statistics per-entry? in the ACL.

The commands to troubleshoot ACL are as follows:

- **show access-lists** *name*
- **show hardware access-lists module** *module*
- **show hardware access-lists resource-utilization module** *module*

PHX2-N7K-1# show access-lists sample-86

```
IP access list sample-86
  statistics per-entry
  10 permit ospf any any [match=0]
  20 permit pim any any [match=0]
  30 permit tcp any any [match=0]
  40 permit ip any any [match=0]
```

PHX2-N7K-1# show hardware access-list mod 1

```

                VLAN 86 :
                =====
No ingress policies
No Netflow profiles in ingress direction

Policies in egress direction:
  Policy type          Policy Id      Policy name
-----
  RACL                  4              sample-86
No Netflow profiles in egress direction

Tcam 1 resource usage:
-----
Label_b = 0x800
Bank 0
-----
  IPv4 Class
    Policies: RACL(sample-86) [Merged]
    1 tcam entries

1 14 protocol cam entries
0 mac etype/proto cam entries
0 lous
0 tcp flags table entries
0 adjacency entries

                VDC-1 Ethernet1/1 :
                =====
Policies in ingress direction:
  Policy type          Policy Id      Policy name
-----
  QoS                    1
No Netflow profiles in ingress direction

Policies in egress direction:
  Policy type          Policy Id      Policy name
-----
  QoS                    2
No Netflow profiles in egress direction
```

...

```

VDC-1 CoPP :
=====
Policies in ingress direction:
  Policy type          Policy Id      Policy name
-----
  QoS                  3

No Netflow profiles in ingress direction

Tcam 1 resource usage:
-----
Label_b = 0x1
Bank 1
-----
  IPv4 Class
    Policies: QoS()
    100 tcam entries
  IPv6 Class
    Policies: QoS()
    73 tcam entries

3 14 protocol cam entries
0 mac etype/proto cam entries
2 lous
0 tcp flags table entries
0 adjacency entries

No egress policies
No Netflow profiles in egress direction

```

PHX2-N7K-1# show hardware access-list resource utilization mod 1

```

ACL Hardware Resource Utilization (Module 1)
-----
          Used      Free      Percent
          Utilization
-----
Tcam 0, Bank 0          1      16383      0.00
Tcam 0, Bank 1          2      16382      0.01
Tcam 1, Bank 0          6      16378      0.03
Tcam 1, Bank 1        176      16208      1.07

```

The next feature we will look at is the QoS troubleshooting for the Nexus. Note, we will have QoS applied, potentially, on both ingress and egress. So we should interrogate both the ingress and egress QoS.

The commands to troubleshoot QoS are

- **show policy-map interface** *interface*

PHX2-N7K-1# show policy-map interface eth 1/2

```

Global statistics status :   enabled
Ethernet1/2

Service-policy (queuing) input:   default-in-policy
policy statistics status:   enabled

```

```
Class-map (queuing):  in-q1 (match-any)
  queue-limit percent 50
  bandwidth percent 80
  queue dropped pkts : 0

Class-map (queuing):  in-q-default (match-any)
  queue-limit percent 50
  bandwidth percent 20
  queue dropped pkts : 0
```

```
Service-policy (qos) output:  test-police-86
policy statistics status:  enabled
```

```
Class-map (qos):  test-police-86 (match-all)
Match: dscp 18
police cir 100 mbps bc 200 ms
```

```
Service-policy (queuing) output:  default-out-policy
policy statistics status:  enabled
```

```
Class-map (queuing):  out-pq1 (match-any)
  priority level 1
  queue-limit percent 16
  queue dropped pkts : 0
```

```
Class-map (queuing):  out-q2 (match-any)
  queue-limit percent 1
  queue dropped pkts : 0
```

```
Class-map (queuing):  out-q3 (match-any)
  queue-limit percent 1
  queue dropped pkts : 0
```

```
Class-map (queuing):  out-q-default (match-any)
  queue-limit percent 82
  bandwidth remaining percent 25
  queue dropped pkts : 0
```

Netflow processing also has portions which occur in hardware. For netflow, we collect statistics in hardware on the linecards. We then, can export them via software.

The commands to troubleshoot Netflow are

- **show flow interface**
- **show flow record**
- **show flow monitor**
- **show hardware flow ip module *module***

PHX2-N7K-1# **show flow interface vlan 86**

```
Interface Vlan86:
  Monitor: sample-86
  Direction: Input
  Monitor: sample-86
  Direction: Output
```

PHX2-N7K-1# **show flow record**

```
Flow record netflow-original:
  Description: Traditional IPv4 input NetFlow with origin ASs
  No. of users: 2
```

Template ID: 256

Fields:

```

match ipv4 source address
match ipv4 destination address
match ip protocol
match ip tos
match transport source-port
match transport destination-port
match interface input
match interface output
match flow direction
collect routing source as
collect routing destination as
collect routing next-hop address ipv4
collect transport tcp flags
collect counter bytes
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last

```

PHX2-N7K-1# show flow monitor

```

Flow Monitor Solarwinds1:
  Use count: 0
  Flow Record: netflow-original
  Flow Exporter: Solarwinds
Flow Monitor sample-86:
  Use count: 2
  Flow Record: netflow-original


```

switch(config)# show hardware flow ip module 8

D - Direction; L4 Info - Protocol:Source Port:Destination Port
 IF - Interface: ()ethernet, (S)vi, (V)lan, (P)ortchannel, (T)unnel
 TCP Flags: Ack, Flush, Push, Reset, Syn, Urgent

D	IF	SrcAddr	DstAddr	L4 Info	PktCnt	TCP Flags
I	8/26	007.002.000.002	007.001.000.002	000:00000:00000	0000421885
I	8/25	007.001.000.002	007.002.000.002	000:00000:00000	0000421900
O	8/25	007.002.000.002	007.001.000.002	000:00000:00000	0000422213
O	8/26	007.001.000.002	007.002.000.002	000:00000:00000	0000422228

Cisco NX-OS supports a hardware based intrusion detection system that checks for ip packet verification. These checks handle well known, and unusable traffic types which can be witnessed during malicious activity, such as if the source is a broadcast address, or if the destination is the 0.0.0.0 address. You can validate if any of these checks are dropping packets.

 **Note:** It has been shown in the field, that frequently it is advantageous to disable IP fragment verification. This is done via the below command

- **no hardware ip verify fragment**

To validate if you are seeing any drops because of ip packet verification, you can use the below command.

- **show hardware forwarding ip verify**
- **show hardware rate-limit module *module***

PHX2-N7K-1# show hardware forwarding ip verify

IPv4 and v6 IDS Checks	Status	Packets Failed
address source broadcast	Enabled	0
address source multicast	Enabled	0
address destination zero	Enabled	0
address identical	Enabled	0
address source reserved	Enabled	8
address class-e	Disabled	--
checksum	Enabled	0
protocol	Enabled	0
fragment	Enabled	0
length minimum	Enabled	0
length consistent	Enabled	0
length maximum max-frag	Enabled	0
length maximum udp	Disabled	--
length maximum max-tcp	Enabled	0
tcp flags	Disabled	--
tcp tiny-frag	Enabled	0
version	Enabled	0
IPv6 IDS Checks	Status	Packets Failed
length consistent	Enabled	0
length maximum max-frag	Enabled	0
length maximum udp	Disabled	--
length maximum max-tcp	Enabled	0
tcp tiny-frag	Enabled	0
version	Enabled	0

PHX2-N7K-1# show hardware rate-limit module 1

Units for Config: packets per second

Allowed, Dropped & Total: aggregated since last clear counters

Rate Limiter Class	Parameters
layer-3 mtu	Config : 500 Allowed : 0 Dropped : 0 Total : 0
layer-3 ttl	Config : 500 Allowed : 0 Dropped : 0 Total : 0
layer-3 control	Config : 10000 Allowed : 0 Dropped : 0 Total : 0
layer-3 glean	Config : 100 Allowed : 0 Dropped : 0 Total : 0
layer-3 multicast directly-connected	Config : 3000 Allowed : 0 Dropped : 0 Total : 0

```

layer-3 multicast local-groups          Config      : 3000
                                         Allowed     : 0
                                         Dropped    : 0
                                         Total       : 0

layer-3 multicast rpf-leak             Config      : 500
                                         Allowed     : 0
                                         Dropped    : 0
                                         Total       : 0

layer-2 storm-control                  Config      : Disabled

access-list-log                         Config      : 100
                                         Allowed     : 0
                                         Dropped    : 0
                                         Total       : 0

copy                                    Config      : 30000
                                         Allowed     : 0
                                         Dropped    : 0
                                         Total       : 0

receive                                 Config      : 30000
                                         Allowed     : 20450903
                                         Dropped    : 0
                                         Total       : 20450903

layer-2 port-security                  Config      : Disabled


layer-2 mcast-snooping                  Config      : 10000
                                         Allowed     : 906
                                         Dropped    : 0
                                         Total       : 906

```

Layer 3 forwarding for Routed Traffic

The Layer 3 engine will only perform Layer 3 forwarding for traffic that is routed through the router. This is traffic which has been sent to the MAC address of a valid routed interface, local to the router.

To troubleshoot the routed traffic, we need to perform the following tasks: 1. Ensure that the control plane routing is correct. 2. Ensure that the hardware forwarding entries on the ingress module have the corresponding information.

 **Note:** All routing of traffic is performed on the forwarding engine of the ingress module.

For our example, we will troubleshoot a route 86.86.87.0/24, which is set to a next hop of 86.86.86.1, and set to route out of VLAN 86 (an SVI).

We first will look at the route, ensure it is set to the correct next hop (86.86.86.1), and set to route out of VLAN 86. We will then want to ensure that we have a corresponding ARP entry associated with this next hop, and validate that the adjacency is in the adjacency table.

As we can see below, 86.86.87.0/24 is set to route to 86.86.86.1, out VLAN 86. This next hop is associated with MAC address 0011.aabb.cccd. We will use this information to investigate the hardware, next.

The commands used to accomplish this are as follows:

- **show ip route** (prefix)

- **show ip arp** (nexthop)
- **show ip adjacency**

PHX2-N7K-1# **show ip route 86.86.87.0/24**

```
IP Route Table for VRF "default"
86.86.87.0/24, 1 ucast next-hops, 0 mcast next-hops
  *via 86.86.86.1, Vlan86, [1/0], 01:19:24, static
```

PHX2-N7K-1# **show ip arp 86.86.86.1**

```
IP ARP Table
Total number of entries: 1
Address      Age      MAC Address  Interface
86.86.86.1   -       0011.aabb.ccdd Vlan86
```

PHX2-N7K-1# **show ip adjacency**

```
IP Adjacency Table for VRF default
Total number of entries: 1
Address      Age      MAC Address  Pref Source  Interface
86.86.86.1   00:00:37 0011.aabb.ccdd 1   Static      Vlan86
```

The above example shows the control plane. Now that we know how things are supposed to work, we can interrogate the hardware to ensure the hardware entries have propagated properly to the Layer 3 hardware engine. We can see that the IP FIB has properly associated 86.86.87.0/24 to the next hop of 86.86.86.1. We can also see, in the hardware entry, that this is routed out VLAN 86, that the RPF is valid if we have enabled RPF Checking), and that the route entry is correctly associated with the MAC address of 0011.aabb.ccdd.

This demonstrates that the routing in the forwarding plane is programmed correctly and that the forwarding will follow the information contained in the routing protocols.

The commands used to accomplish this are as follows:

- **show ip fib route *prefix module module***
- **show system internal forwarding route *prefix detail module module***

PHX2-N7K-1# **sho ip fib route 86.86.87.0/24 mod 1**

```
IPv4 routes for table default/base
-----+-----+-----
Prefix      | Next-hop      | Interface
-----+-----+-----
86.86.87.0/24  86.86.86.1    Vlan86
```

PHX2-N7K-1# **show system internal forwarding route 86.86.87.1/24 detail mod 1**

```
RPF Flags legend:
  S - Directly attached route (S_Star)
  V - RPF valid
  M - SMAC IP check enabled
  G - SGT valid
  E - RPF External table valid
86.86.87.0/24      , Vlan86
```

```

Dev: 1 , Idx: 0x19001 , RPF Flags: V      , DGT: 0 , VPN: 1
RPF_Intf_5:  Vlan86      (0x55      )
AdjIdx: 0x43005, LIFB: 0      , LIF: Vlan86      (0x55      ), DI: 0x0
DMAC: 0011.aabb.ccdd SMAC: 0023.ac67.dd41

```

SFabric Processing Occurs (optional)

This step occurs if the packet needs to traverse the fabric.

In this step, we need to interrogate if the fabrics are functioning properly, and if their utilization is at an acceptable level. We can view the fabric status and utilization using the following commands:

- **show hardware fabric-utilization**
- **show module fabric**

switch(config)# **show hardware fabric-utilization**

```

-----
Slot  Direction  Utilization
-----
 2   ingress    3%
 2   egress     3%
 6   ingress    1%
 6   egress     1%

```

PHX2-N7K-1# **show module fabric**

Xbar	Ports	Module-Type	Model	Status
1	0	Fabric Module 1	N7K-C7010-FAB-1	ok
2	0	Fabric Module 1	N7K-C7010-FAB-1	ok
3	0	Fabric Module 1	N7K-C7010-FAB-1	ok

Xbar	Sw	Hw
1	NA	1.0
2	NA	1.0
3	NA	1.0

Xbar	MAC-Address(es)	Serial-Num
1	NA	JAF1252AHRB
2	NA	JAF1251CABF
3	NA	JAF1252AHBL

* this terminal session

Layer 2 Engine Performs Source/Destination MAC Processing

To validate forwarding of the Layer 2 engine, we should first look at the centralized MAC table aggregated on the supervisor to validate whether the MAC addresses are correlated as we expect them, and assigned to the ports where we expect the MAC's to reside.

Based on this, we can then validate the hardware programming on the egress module to validate that our MAC address table is properly programmed into the hardware based Layer 2 engine on the module.

The commands used to accomplish this are as follows:

- **show mac address-table**
- **show hardware mac address-table module interface interface**

The output from these commands are documented in steps 3-4 above.

Egress Port QoS is Performed

As the packet nears entry from the Cisco Nexus 7000, one of the final steps is the application of Port QoS. The port level QoS will be able to buffer the traffic in times of congestion. To interrogate the egress QoS, we look at following QoS commands, paying attention to the Transmit QoS:

- **show queuing interface interface**
- **show policy-map interface**

PHX2-N7K-1# **show queuing int eth 1/1**

Interface Ethernet1/1 TX Queuing strategy: Weighted Round-Robin
Port QoS is enabled

Queuing Mode in TX direction: mode-cos

Transmit queues [type = 1p7q4t]

Queue Id	Scheduling	Num of thresholds
1p7q4t-out-q-default	WRR	04
1p7q4t-out-q2	WRR	04
1p7q4t-out-q3	WRR	04
1p7q4t-out-q4	WRR	04
1p7q4t-out-q5	WRR	04
1p7q4t-out-q6	WRR	04
1p7q4t-out-q7	WRR	04
1p7q4t-out-pq1	Priority	04

Configured WRR

WRR bandwidth ratios: 25[1p7q4t-out-q-default] 15[1p7q4t-out-q2] 12[1p7q4t-out-q3] 12[1p7q4t-out-q4]

WRR configuration read from HW

WRR bandwidth ratios: 25[1p7q4t-out-q-default] 15[1p7q4t-out-q2] 11[1p7q4t-out-q3] 11[1p7q4t-out-q4]

Configured queue-limit ratios

queue-limit ratios: 78[1p7q4t-out-q-default] 1[1p7q4t-out-q2] 1[1p7q4t-out-q3]

*1[1p7q4t-out-q4] *1[1p7q4t-out-q5] *1[1p7q4t-out-q6] *1[1p7q4t-out-q7]

16[1p7q4t-out-pq1]

* means unused queue with mandatory minimum queue-limit

queue-limit ratios configuration read from HW

queue-limit ratios: 78[1p7q4t-out-q-default] 1[1p7q4t-out-q2] 1[1p7q4t-out-q3]

*1[1p7q4t-out-q4] *1[1p7q4t-out-q5] *1[1p7q4t-out-q6] *1[1p7q4t-out-q7]

16[1p7q4t-out-pq1]

* means unused queue with mandatory minimum queue-limit

Thresholds:

COS	Queue	Threshold Type	Min	Max
0	1p7q4t-out-q-default	DT	100	100
1	1p7q4t-out-q-default	DT	100	100
2	1p7q4t-out-q-default	DT	100	100
3	1p7q4t-out-q-default	DT	100	100

4	1p7q4t-out-q-default	DT	100	100
5	1p7q4t-out-pq1	DT	100	100
6	1p7q4t-out-pq1	DT	100	100
7	1p7q4t-out-pq1	DT	100	100

...

PHX2-N7K-1# show policy-map interface eth 1/2

```
Global statistics status :   enabled
Ethernet1/2
Service-policy (queuing) input:   default-in-policy
  policy statistics status:   enabled

Class-map (queuing):   in-q1 (match-any)
  queue-limit percent 50
  bandwidth percent 80
  queue dropped pkts : 0

Class-map (queuing):   in-q-default (match-any)
  queue-limit percent 50
  bandwidth percent 20
  queue dropped pkts : 0

Service-policy (queuing) output:   default-out-policy
  policy statistics status:   enabled

Class-map (queuing):   out-pq1 (match-any)
  priority level 1
  queue-limit percent 16
  queue dropped pkts : 0

Class-map (queuing):   out-q2 (match-any)
  queue-limit percent 1
  queue dropped pkts : 0

Class-map (queuing):   out-q3 (match-any)
  queue-limit percent 1
  queue dropped pkts : 0

Class-map (queuing):   out-q-default (match-any)
  queue-limit percent 82
  bandwidth remaining percent 25
  queue dropped pkts : 0
```

Linksec Encryption Occurs

The command used to troubleshoot Linksec encryption is

- **show cts interface {all | interface}**

The output from this command is documented in step 2 above.

Packet is Transmitted

The final step in the process is the transmission of the frame out of the physical egress port. Troubleshooting of the physical port, is the same as in step 1, and includes the following commands:

- **show interface** *interface*
- **show interface** *interface* **transceiver**

The output from these commands are documented in step 1 above.