



*LET'S  
BUILD  
TOMORROW  
TODAY*

# *Advanced Troubleshooting Nexus 7000 Series Switches*

Matt Esau (CCIE# 18586)

Nexus 7000 BU, Data-Center Customer-Operations

Dale Miller (CCIE# 24965)

Technical Leader, Data-Center Switching Team

BRKDCT-3234

# Session Goal

- Who For:
  - Those implementing or have existing Nexus 7/77K platforms in their network.
  - New to, or already familiar with NXOS.
  - Anyone else who just wants to hang out and learn 😊
- Session Goal
  - Troubleshooting methodology is the same, no matter the platform...
  - **Where and What** (where to look, and what to use)
  - Quick and easy data collection **Cheat Sheets**
  - **Faster root cause** = decreased impact + eliminate re-occurrence

# Recommended Sessions

Reading suggestions or equivalent knowledge



**BRKARC-3470**

Advanced - Cisco Nexus 7000/7700 Switch Architecture

**BRKARC-3471**

Cisco NX-OS Software Architecture

**BRKDCT-2121**

Virtual Device Context (VDC) Design and Implementation Considerations  
with Nexus 7000



# Other Recommendations

## More suggestions



### BRKDCT – 1890

Network Visibility using Advanced Analytics in Nexus Switches

### BRKDCT-2333

DC Network Failure Detection

### BRKDCT-3103

Advanced OTV - Configure, Verify and Troubleshoot OTV in Your Network

# Agenda

- NX-OS Troubleshooting Tools
- Control Plane Troubleshooting
- Data Plane Troubleshooting
- Q&A / Conclusion
- Appendix



Cisco Live 2015 San Diego :  
120 min

# Agenda

- NX-OS Troubleshooting Tools
- Control Plane Troubleshooting
- Data Plane Troubleshooting
- Q&A / Conclusion
- Appendix

# *Granular Show Tech-Support and CLI Filtering*

# Granular Show-Techs

- Every feature has its own ``sh tech <feature>'`
- Issues frequently involve multiple features so good practice to collect generic ``sh tech detail'` + ``sh tech <feature>'`
- Collect show techs as early as possible

```
N7K# show tech ?
aclmgr          ACL commands
aclqos          Show information for aclqos technical support
adjmgr          Display Adjmgr information
```

- 'sh tech' can be over **150MB+** and can take several minutes...

```
N7K# sh tech details
---- show tech-support details ----
Show tech details will take 4-6 minutes to complete. Please Wait ...
```

# Granular Show-Techs cont...

## Alternatives:

1. Tac-Pac: Still takes 4-6 minutes, but automatically zips file reducing size
2. Binary-Tech: **Done in seconds** and compresses into .tar – faster and reduces file size
  - CLI Issued from default VDC, and takes tech-support for ALL VDCs

```
N7K# tac-pac bootflash:tac-pac
Show tech details will take 4-6 minutes to complete. Please Wait ...
N7K# dir bootflash: | eg -i tac
24521266 Mar 14 04:27:07 2015 tac-pac
```

Compressed tech,  
but still 4-6 minutes

```
N7K# show tech-support all binary bootflash:
N7K# dir bootflash: | eg -i binary
65208320 Mar 14 03:13:02 2015 binary_show_tech_all_03_14_2015_03_12_33HRS.tar
```

Binary format so  
must be parsed by  
TAC

# CLI Filtering and Redirection

- Re-direct show commands to a file (can also append multiple cmds)

```
N7K# show ip route > bootflash:ip-route.txt
N7K# dir bootflash: | eg -i route
4873 Jan 26 21:09:29 2015 ip-route.txt

N7K# show ip route vrf all >> bootflash:ip-route.txt
N7K# dir bootflash: | eg -i route
10385 Jan 26 21:17:11 2015 ip-route.txt
```

">" Creates a new file

">>" Appends to existing file

- Best Practice** to run full command and then grepped output

```
N7K# sh ver | ?
egrep      Egrep - print lines matching a pattern
grep       Grep - print lines matching a pattern
head       Display first lines
last       Display last lines

N7K# show logging logfile | grep -b 5 -i memory | grep "Mar 22"
Mar 22 15:40:13 N7K-1 %BGP-5-MEMALERT: bgp-1 [3439] BGP memory status changed from OK
to Minor Alert
```



# What commands are available

- Quick way to find what commands are avail for what features:

```
N7K# show tech ospf | eg -i `show`
`show running-config ospf`
`show running-config rpm`
`show ip ospf internal event-history cli`
`show ip ospf vrf all`
`show ip ospf ha vrf all`
```

```
N7K# sh cli syntax | eg -i ospf
(2510) show ip ospf [ <tag> ] neighbors { { <interface> summary } | { summary [ vrf { <vrf-name>
(2511) show ip ospf [ <tag> ] interface brief [ vrf { <vrf-name> | <vrf-known-name> |
(2512) show ip ospf [ <tag> ] interface [ <interface> | vrf { <vrf-name> | <vrf-known-name>
(2513) show ip ospf [ <tag> ] virtual-links brief [ vrf { <vrf-name> | <vrf-known-name> |
(2514) show ip ospf [ <tag> ] virtual-links [ vrf { <vrf-name> | <vrf-known-name> |
(2515) show ip ospf [ <tag> ] sham-links [ brief ] [ vrf { <vrf-name> | <vrf-known-name> |
```



# *Logging Capabilities*

# Logging Capabilities: Logflash

- Logflash is an 8GB **persistent** storage location.
- Provides storage for info such as: syslog messages, debug output, core file
- Sup1 = compact flash
- Sup2/2E = USB flash drive
- Should **ALWAYS** be inserted into supervisor. Syslog alerts when it's not

```
2014 Sep 1 13:08:56 N7K-18 %USBHSD-1-ABSENT_ALERT: logflash: not present, Please
ensure logflash is inserted to save debug logs
```

- If not recognized, try formatting

```
N7K# format logflash:
=====
NOTE: Formatting logflash can take over 60s. Please be patient.
=====
```

# Logging Capabilities: Logflash cont...

- ``show logging log-file`` standard logging messages but **lost on reload**
- Logflash allows for persistent storage – **survives reloads**

```
N7K# sh clock
21:19:03.878 UTC Fri Jan 23 2015
N7K# sh ver | in uptime
Kernel uptime is 16 day(s), 2 hour(s), 45 minute(s), 59 second(s)

N7K# show file logflash://sup-active//log/messages
2010 Jan 1 14:05:54 %IDEHSD-2-MOUNT: logflash: online
2010 Jan 1 14:06:07 %MODULE-5-ACTIVE SUP OK: Supervisor 1
JAF1545BTGH)
2010 Jan 1 14:06:07 %PLATFORM-5-MOD_STATUS: Module 6 current-stat

N7K# dir logflash://sup-standby/vdc_3/log/messages
219040 Jul 16 20:51:25 2012 vdc_3/log/messages
```

System rebooted  
~Jan 2015

We can still see  
messages from Jan  
2010

Can read non-default  
vdc's and standby sup  
logs

# Logging Capabilities: Accounting Log

- ``show accounting log`` allows us to see all **'config' commands**

```
N7K# show accounting log
Fri Nov 16 16:51:20 2012:type=update:id=172.18.118.33@pts/0:user=mesau:cmd=switchto ;
configure terminal ; interface Vlan143 (SUCCESS)
Fri Nov 16 16:51:20 2012:type=update:id=172.18.118.33@pts/0:user=mesau:cmd=switchto ;
configure terminal ; interface Vlan143 ; no shutdown (SUCCESS)
```

- ``terminal log-all`` adds logging of **'show' commands** as well

```
Nexus-7k(config)# terminal log-all
Nexus-7k(config)# show accounting log all
2:user=mesau:cmd=switchto ; show version internal build-identifier (SUCCESS)
Sat Jan 26 21:41:28 2015:type=update:id=10.82.245.163@pts/2:user=mesau:cmd=switchto ;
show accounting log all | eg build- (SUCCESS)
```

- Accounting logs also have **persistent** storage on Logflash:

```
N7K# dir logflash://sup-active/vdc_1
130557 Jan 26 21:46:12 2015 accounting_log
```

# *Debug Capabilities*

# Debug Capabilities: Event-Histories

- Event-histories - are “**running debugs**” on by default (per vdc / per feature)
- No more waiting for maintenance windows to debug 😊
- No CPU Impact

```
Nexus-7k(config)# show ip pim event-history join-prune
```

```
5) Event:E_DEBUG, length:61, at 61372 usecs after Sat Mar 14 20:18:24 2015
```

```
[116] : Received Hello from 10.0.104.2 on Vlan104, length: 30
```

```
6) Event:E_DEBUG, length:113, at 613090 usecs after Sat Mar 14 20:18:22 2015
```

```
[116] : iod = 49 - Send Hello on Vlan1510 from 10.15.10.252, holdtime: 105 secs,  
genID: 0xdc8661f, dr-priority: 1
```

```
Nexus-7k(config)# sh ip eigrp event-history packet
```

```
IP-EIGRP packet events for AS 50
```

```
2015 Mar 14 18:11:50.408038 eigrp 50 [7816]: : Send ACK 10.0.102.252->10.0.102.2 Vlan102  
len:20 seq:0 ack:6624 flg:0
```

# Debug Capabilities: Debug Filter

- Debug-filter allows more granularity
- Helpful when debugging specific packets, protocols, subnets, etc.
- Able to apply multiple filters at once.

```
N7K# debug-filter pktmgr interface e4/1
N7K# debug-filter pktmgr dest-mac 0100.5e00.000d
N7K# debug pktmgr frame
```

I've set up filters for PIM pkts on a specific interface

```
N7K# show ip pim int b | count
25
```

25 PIM interfaces, but only printing debugs for e4/1

```
N7K# show debug-filter all
debug-filter pktmgr dest-mac 0100.5E00.000D
debug-filter pktmgr interface Ethernet4/1
```

```
2013 Jan 27 20:12:19.447962 netstack: In 0x0800 72 7 001f.27c8.8c00 -> 0100.5e00.000d Eth4/1
2013 Jan 27 20:12:24.996482 netstack: Out 0x0800 64 6 64a0.e744.3942 -> 0100.5e00.000d Eth4/1
2013 Jan 27 20:12:48.607940 netstack: In 0x0800 72 7 001f.27c8.8c00 -> 0100.5e00.000d Eth4/1
```

# Debug Capabilities: Debug Log-File

- Debugs also still available
- No more concern about printing debugs to terminal and CPU impact
- Can be directly logged to a logfile and viewed immediately

```
N7K# debug logfile ospf1
N7K# dir log:
    1261623      Jan 28 16:24:16 2015  messages
    18628       Jan 24 19:12:22 2015  ospf1
    16939       Jan 24 19:07:23 2015  startupdebug

N7K# show debug logfile ospf1
2015 Jan 24 19:10:12.801652 ospf: 1 [12275] (default) Nbr 10.10.10.4 FSM start: old state
INIT, event HELLORCVD
```

- `'undebug all'` command disables all existing debugs.



# *OBFL & GOLD*

## *(Module Troubleshooting)*

# OBFL (On-Board Failure Logging)

- On-Board Failure Logging provides **persistent storage**
- Primarily used for Module troubleshooting
  - Insertion & Removal times
  - Reset Reasons
  - Error counters & packet drops,
  - ISSU Info & LC CPU info/issues
- Info is tied to LC, so move LC to new chassis = still access OBFL info
- ``show logging onboard`` & ``show tech mod <x|all>``
- Let's look at a few examples:

# OBFL – Practical Example

```
N7K# show clock; show logging onboard mod 3 exception-log
2015-03-16 20:02:15
```

```
Device Id       : 70
Device Name     : Santa-Cruz-Module
Sys Error      : X-bar Interface ASIC Error
Errtype        : WARNING
Error Description : Lost Sync on slot:1 asic:0 port:16 to Fabric:1
Time           : Thu Jan 26 13:50:08 2012
                (414667 usecs 4F215A10(H) jiffies)
```

Able to store information  
from 3+ years ago

Specify time-range you  
want to see any  
“interrupts” or “counters”  
that occurred around time  
of event

```
N7K# show logging onboard mod 3 starttime 01/26/12-13:49:00 endtime 01/26/12-13:50:30
```

```
Thu Jan 26 13:50:08 2012 (464621 us)
[sc_dc3_check_chico_intr]SCZ Chico Interrupt Detected slot: 1 asic:0 port:11-->connected to -->Fabric-
Slot:1
Thu Jan 26 13:50:08 2012 (454727 us)
[sc_dc3_check_chico_intr]SCZ Chico Interrupt Detected slot: 1 asic:0 port:7-->connected to -->Fabric-
Slot:1
```

# GOLD: Generic On-Line Diagnostics

- Diagnostic test suite to detect hw, sw, or even traffic related issues.
- **Common mistake** = GOLD Failure does not always indicate HW failure & RMA
- Tests for Supervisors (active & standby), LCs, and Fabric/Spine modules.
  - Bootup
  - Health Monitoring (run while system is live)
  - On-demand
- **NOTE:** Show commands used in Default-VDC, but tests run on all hardware regardless of VDC

# GOLD: Generic On-Line Diagnostics

## ■ What tests are available ?

```
N7K# show diagnostic content module 1
```

ID	Name	Attributes	Testing Interval (hh:mm:ss)
1)	ASICRegisterCheck----->	***N*****A	00:01:00
2)	PrimaryBootROM----->	***N*****A	00:30:00
<snip>			
11)	BootupPortLoopback----->	CP*N**XE*T*	-NA-

Intervals are configurable

Legend explaining attributes included in this command (omitted here)

## ■ What does each test do?

```
N7K# show diagnostic description module 1 test PortLoopback
```

```
PortLoopback :
```

A health monitoring test that will test the packet path from the Supervisor card to the physical port in ADMIN DOWN state on Linecards.

# GOLD: Generic On-Line Diagnostics

- GOLD can take corrective action upon failure - 6.2(8) and later
  - PortLoopback, RewriteEngineLoopback, SnakeLoopback test , and StandbyFabricLoopback.

```
Nexus-7k(config)# diagnostic eem action conservative
Nexus-7k(config)# show diagnostic eem action
```

EEM action is Conservative

Useful to see syslog  
msgs, test corrective  
actions, etc

- Simulate test failures

```
Nexus-7k(config)# diagnostic test simulation mod 2 test 7 fail
Nexus-7k(config)# show log log
2015 Mar 17 03:34:08 %DIAG_PORT_LB-2-REWRITE_ENGINE_LOOPBACK_TEST_FAIL: Module:2 Test:RewriteEngine
Loopback failed 10 consecutive times due to faulty port(s):1-24 Error:Simulated Error for Forwarding
Asic ports. Error disabling ports
```

# GOLD: Generic On-Line Diagnostics

- View diagnostic result details and statistics

```
Nexus-7k(config)# show diagnostic result mod 2 statistics
```

```
6) PortLoopback
```

Port No	Packet TX	Packet RX	Packet Loss
1	15624	15624	0
----- SNIP -----			
24	15628	15625	3

**Remember...**

Diag drops / failures do not mean 100% hardware failure.

Packet Loss could be due to congestion as well...

```
Nexus-7k(config)# show diagnostic result mod 2 test 7 detail
```

```
7) RewriteEngineLoopback:
```

```
Error code -----> DIAG TEST ERR DISABLE
Total run count -----> 4223
Last test execution time ----> Tue Mar 17 03:35:08 2015
First test failure time ----> Tue Mar 17 03:25:08 2015
Last test failure time ----> Tue Mar 17 03:34:08 2015
Last test pass time -----> Tue Mar 17 03:24:08 2015
Total failure count -----> 10
Consecutive failure count ---> 10
Last failure reason -----> Simulated Test
```

# Putting it all Together

Log Msg indicating Mod  
2 failure

```
N7K# show logging log
2014 Jun 6 14:38:49 %MODULE-2-MOD_DIAG_FAIL: Module 2 reported failure on ports 2/17-2/17 (Ethernet)
```

```
N7K# show logging onboard exception-log
<snip> - some irrelevant output omitted
Device Id       : 79
Device Name     : Metropolis
Sys Error       : Metro fatal interrupt
Errtype         : CATASTROPHIC
PhyPortLayer    : Ethernet
Port(s) Affected : 17, 19, 21, 23, 25, 27, 29, 31
Time            : Fri Jun 6 14:38:46 2014
```

Gives specific ASIC on LC  
that had the problem

Found config changes being  
made just prior to issue –  
led us to be able to  
reproduce and RCA issue

```
N7K# show accounting log
Fri Jun 6 14:02:38 2014 cmd=configure terminal ; monitor session 1 ; source vlan 2909 both (SUCCESS)
Fri Jun 6 14:12:42 2014 cmd=configure terminal ; monitor session 1 (SUCCESS)
Fri Jun 6 14:12:43 2014 cmd=configure terminal ; monitor session 1 ; filter vlan 1-3967 include-
untagged (SUCCESS)
```



# GOLD: Generic On-Line Diagnostics



Test Name	Test Type	Description
ASICRegisterCheck	Health Monitoring	Checks read/write access to scratch registers on ASICs on the module. Runs on all modules. (enabled by default)
CryptoDevice	Bootup	Checks the CTS device initialization on the module. Run on the Supervisor only.
NVRAM	Health Monitoring	Checks the sanity of the NVRAM device on the module. Runs on the Supervisor only. (enabled by default)
RealTimeClock	Health Monitoring	Verifies the real time clock on the module. Runs on the Supervisor only. (enabled by default)
PrimaryBootROM	Health Monitoring	Verifies the primary BootROM state. Runs on all modules.
SecondaryBootROM	Health Monitoring	Verifies the secondary BootROM state. Runs on all modules.
CompactFlash	Health Monitoring	Verifies access to the internal and external compactflash devices. Runs on the Supervisor only. (enabled by default)
PwrMgmtBus	Health Monitoring	Verifies the redundant Power Management Control Bus. Runs on the Supervisor only. (enabled by default)

# GOLD: Generic On-Line Diagnostics



Test Name	Test Type	Description
SpineControlBus	Health Monitoring	Verifies the redundant Spine Card Control Bus. Runs on the Supervisor only. (enabled by default)
SystemMgmtBus	Health Monitoring	Verifies the redundant System Management Bus that controls the Power Supplies and the Fans. Runs on the Supervisor only. (enabled by default)
MgmtPortLoopback	Bootup	Tests loopback on the management port of the module. Runs on the Supervisor only.
EOBCPortLoopback	Bootup	Tests loopback on the EOBC interface of the module. Runs on all modules.
RewriteEngineLoopback	Health Monitoring	Performs non-disruptive loopback for all LineCard ports up to the Rewrite Engine ASIC. (enabled by default)
PortLoopback	Health Monitoring	Verifies the packet path loopback till the port physical device for admin down ports. Runs on LineCards only (enabled by default)
SpineControlBus	Health Monitoring	Verifies the redundant Spine Card Control Bus. Runs on the Supervisor only. (enabled by default)
SystemMgmtBus	Health Monitoring	Verifies the redundant System Management Bus that controls the Power Supplies and the Fans. Runs on the Supervisor only. (enabled by default)

# GOLD: Generic On-Line Diagnostics



Test Name	Test Type	Description
OBFL	Bootup	Checks the onboard flash used for failure logging (OBFL) device initialization on the module (enabled by default)
SnakeLoopback	Health Monitoring	Does SnakeTest for SoC based ASICs (F1, F2, and F2e modules)
InternalPortLoopback	Health Monitoring	Nondisruptive per-port loopback test, and hence can run on ports that are up as well (Supported on M2, F2, F2E)
StandbyFabricLoopback	Health Monitoring	Verifies packet path from the Standby supervisor to the Fabric (Supported on Sup1, Sup2, Sup2E)

# Module Troubleshooting Cheat Sheet



- tac-pac
- show tech gold
- show tech mod all
- show tech platform (hidden)

# *Scripting (EEM, Scheduler, & Python)*

# Scripting: EEM

- EEM scripts will have an “*event*” that triggers an “*action*”

Event based on SNMP  
OID (many other events  
also avail)

```
event manager applet HIGH-CPU
  event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1 get-type exact entry-op ge entry-val 60 poll-interval 1
  action 1.0 syslog msg High CPU hit $_event_pub_time
  action 2.0 cli enable
  action 3.0 cli show clock >> bootflash:high-cpu.txt
  action 4.0 cli show proc cpu sort >> bootflash:high-cpu.txt
```

Creating a syslog message  
and running show  
commands when event  
triggers

```
event manager applet clear_hw_counter
  action 1.0 cli show hardware rate-limiter > bootflash:hw_rl.out
  action 2.0 cli clear hardware rate-limit all
  action 3.0 cli show policy-map interface control-plane >> bootflash:hw_rl.out
  action 4.0 cli clear copp stat
```

Notice no “event”.

So how do we trigger?

- Can also be **scheduled** at a specific time or intervals...

# Scripting: Scheduler + EEM

- Scheduler – per vdc feature allows you to schedule specific tasks
  - Save configs, run commands, call EEM scripts etc...

Enable feature then  
Create “job name”

```
Nexus-7k(config)# feature scheduler
Nexus-7k(config)# scheduler job name check-inband
Nexus-7k(config-job)# event manager run clear_hw_counter
```

Job is to “run” EEM script  
“clear\_hw\_counter”

```
Nexus-7k(config)# scheduler schedule name check-inband
Nexus-7k(config)# time start now repeat 00:00:2
Nexus-7k(config)# job name check-inband
Nexus-7k(config)# email-addr john@gmail.com
```

Schedules above job to  
run every two minutes

```
Nexus-7k(config-applet)# sh scheduler schedule
Schedule Name      : check-inband
-----
User Name          : mesau
Schedule Type      : Run every 0 Days 0 Hrs 2 Mins
Start Time         : Wed Mar 18 04:25:53 2015
Last Execution Time : Wed Mar 18 04:35:53 2015
Last Completion Time: Wed Mar 18 04:35:53 2015
Execution count    : 6
```

Scheduler timestamp matches  
file created by EEM script

```
Nexus-7k(config-applet)# dir bootflash:
33756    Feb 26 06:24:24 2014  bootflash
15014    Mar 18 04:35:54 2015  hw_rl.out
```

# Scripting: Scheduler + Python

- Scheduler can also call .py scripts (starting in 6.2.8a)

Scripts must be stored  
in /scripts directory

```
Nexus-7k(config)# dir bootflash:scripts
      130      Mar 30 20:17:47 2015  switch_info_file2.py
      4136     Mar 30 19:59:56 urib-mrib-rpf-check.py
```

```
Nexus-7k(config)# scheduler job name rpf-check
Nexus-7k(config-job)# source urib-mrib-rpf-check.py
```

Calls specified python  
script.

```
Nexus-7k(config-schedule)# show scheduler logfile
```

```
Job Name      : rpf-check                      Job Status: Success (0)
Schedule Name : rpf-check-schedule             User Name  : mesau
Completion time: Mon Mar 30 20:55:42 2015
```

```
----- Job Output -----
`source urib-mrib-rpf-check.py`
```

You can see the  
results of the called  
.py script

```
(200.5.1.200/32, 235.50.1.100/32), uptime: 2d18h, ip mrib pim
Incoming interface: Ethernet4/1, RPF nbr: 172.16.1.42
Outgoing interface list: (count: 1)
  Ethernet2/23, uptime: 2d18h, mrib
```

Of course, you can  
just have .py redirect  
CLI output to file as  
well ☺

```
-----SNIP-----
```



# Scripting: Reference Info



- **EEM:**

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus7000/sw/system-management/guide/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_System\\_Management\\_Configuration\\_Guide/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_System\\_Management\\_Configuration\\_Guide\\_appendix\\_010101.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus7000/sw/system-management/guide/b_Cisco_Nexus_7000_Series_NX-OS_System_Management_Configuration_Guide/b_Cisco_Nexus_7000_Series_NX-OS_System_Management_Configuration_Guide_appendix_010101.html)

- **Scheduler:**

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus7000/sw/system-management/guide/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_System\\_Management\\_Configuration\\_Guide/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_System\\_Management\\_Configuration\\_Guide\\_chapter\\_01110.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus7000/sw/system-management/guide/b_Cisco_Nexus_7000_Series_NX-OS_System_Management_Configuration_Guide/b_Cisco_Nexus_7000_Series_NX-OS_System_Management_Configuration_Guide_chapter_01110.html)

- **Python:**

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6\\_x/nx-os/fundamentals/configuration/guide/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_Fundamentals\\_Configuration\\_Guide\\_Release\\_6-x/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_Fundamentals\\_Configuration\\_Guide\\_Release\\_6-x\\_chapter\\_01011.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6_x/nx-os/fundamentals/configuration/guide/b_Cisco_Nexus_7000_Series_NX-OS_Fundamentals_Configuration_Guide_Release_6-x/b_Cisco_Nexus_7000_Series_NX-OS_Fundamentals_Configuration_Guide_Release_6-x_chapter_01011.html)

# *Packet Capture Tools (SPAN, ELAM, Ethalyzer)*

# Packet Capture Tools: SPAN Options

- Allows us to capture any packet Rx/Tx the switch
  - Even allows us to capture specific exception packets being dropped
  - Copy of original packet
  - Requires external sniffer
- 
- **Note:** Be aware of CSCtg90112 when using **SPAN with Multicast and M1 cards**

# Packet Capture Tools: SPAN Options cont...

## Local SPAN

```
monitor session 2 type local
source interface port-channel1995 both
source interface Ethernet 4/1 both
source interface Ethernet 7/1-7/5 both
source vlan 500-510 both
destination interface Ethernet2/23
```



Sniffer Device

## ERSPAN (Encapsulated Remote)

```
monitor session 1 type erspan-source
erspan-id 41
vrf default
destination ip 100.3.20.101
source interface port-channel11 both
source interface Ethernet4/1 both
```

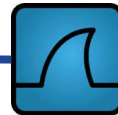
## Exception SPAN (Packet Drops)

```
monitor session 2
source exception all
destination interface Ethernet2/23
```

```
N7K# show hardware ip verify
```

IPv4 IDS Checks	Status	Packets Failed
checksum	Enabled	3671
protocol	Enabled	183
address source multicast	Enabled	851

Now we can Identify source of dropped packets



Sniffer Device

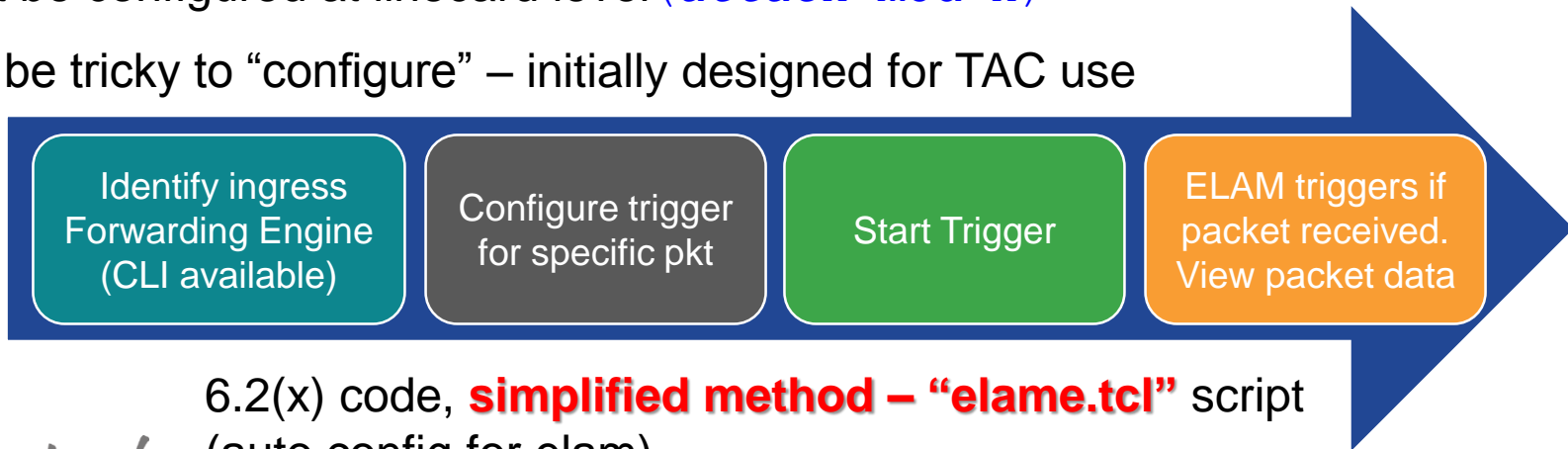
Layer 3 Network



Sniffer Device

# Packet Capture Tools: ELAM

- “Internal” packet capture, **No external sniffer required.**
- Captures **‘single’ packet** at forwarding engine ASIC level of linecard
- Not ideal for intermittent packet drop issues (use SPAN)
- Must be configured at linecard level (`attach mod x`)
- Can be tricky to “configure” – initially designed for TAC use



6.2(x) code, **simplified method – “elame.tcl”** script  
(auto config for elam)

# ELAM Example

```
module-4(eureka-elam)# status
```

```
Slot: 4, Instance: 1
```

Status "Triggered" = we rx packet

```
EU-DBUS: Triggered
```

```
trigger dbus dbi ingress ipv4 if source-ipv4-address 172.16.1.42 destination-ipv4-address 200.5.0.100
```

Must configure a specific "trigger" to capture

```
module-4(eureka-elam)# show dbus
```

```
cos = 0x0
```

```
source_index = 0x00180
```

```
dmac = 64.a0.e7.44.39.42
```

```
smac = 00.1f.27.c8.8c.00
```

```
ip_ttl = 0xff
```

```
vlan = 10
```

```
ip_source = 172.016.001.042
```

```
ip_destination = 200.005.000.100
```

DBUS = packet info 'before' forwarding decision

Src\_idx maps to ingress front panel port

Original pkt info, mac, TTL, lps, vlan, etc

```
module-4(eureka-elam)# show rbus
```

```
cos = 0x0
```

```
dest_index = 0x0017e
```

```
vlan = 20
```

```
data(rit/dmac/recir) = 00.00.00.50.01.00
```

```
data(rit/smac/recir) = 64.a0.e7.44.39.42
```

```
ttl = 0xfe
```

RBUS = packet info 'after' forwarding decision

Dest\_idx maps to egress front panel port

Info post-rewrite, TTL decrement, smac re-written, etc

# ELAM Example



Module #

```
N7K# slot 1 show hardware internal dev-port-map
```

```
-----  
CARD_TYPE:      48 port 10G  
>Front Panel ports:48  
-----
```

```
Device name          Dev role          Abbr num_inst:
```

```
-----  
> Clipper MAC          DEV_ETHERNET_MAC      MAC_0  12  
> Clipper FWD          DEV_LAYER_2_LOOKUP    L2LKP  12  
> Clipper XBAR         DEV_QUEUEING          QUEUE  12  
> Sacramento Xbar ASIC DEV_SWITCH_FABRIC     SWICHF 1  
> PHY                  DEV_PHY               PHYS   12  
> Clipper L3 Driver     DEV_LAYER_3_LOOKUP    L3LKP  12  
-----
```

Internal ASIC Numbering  
And abbreviation

Front Panel  
Port #

```
+-----+  
+-----+FRONT PANEL PORT TO ASIC INSTANCE MAP+-----+  
+-----+
```

FP port	PHYS	MAC_0	L2LKP	L3LKP	QUEUE	SWICHF
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	1	1	1	1	1	0
6	1	1	1	1	1	0
7	1	1	1	1	1	0
8	1	1	1	1	1	0
9	2	2	2	2	2	0

Match FP port to  
LKP (L2/L3 Lookup) instances

# ELAM Example (6.2x with ELAMe)

- **Simpler method** to use ELAM in 6.2(x) code, - elame.tcl script

```
N7K# source sys/elame.tcl 172.16.1.42, 200.5.0.100
```

Specify src and/or dst trigger

```
.. packet captured at FE: 4/1/1
```

Packet was received

```
... capture instance 4/1/1 (slot/port/inst
```

```
+++ IPv4 packet: 118 bytes from MAC 001f.27c8.8c00 / IP 172.16.1.42  
to MAC 64a0.e744.3942 / IP 200.5.0.10 TTL 255
```

Mac+IP Src and Dst info

```
+++ protocol ICMP type 0x08 code 0x00
```

```
+++ packet received on interface Eth4/1 vlan 0 (source index 0x00180)
```

Rx and Tx front-panel ports

```
... rbus: ccc 0x4 cap1 0x0 cap2 0x1 flood 0x0 dest_vlan 0 dest_index 0x0017e  
fwd 0x0
```

```
+++ packet is sent to Eth2/23 on vlan 0
```

```
... destination index is NOT from L2 table lookup
```

```
... lamira OFE: rdt 0x1 dest_index 0x0017e flood 0x0 l2fwd 0x0 ofe_drop 0x0
```

```
... done
```



# Packet Capture Tools: Ethalyzer

- Built-in sniffer that captures packets **Tx/Rx CPU**
- No longer required to have external sniffer
- View capture immediately via CLI, or write to a file for export view.

```
N7K# ethalyzer local interface inband ?
capture-filter      Filter on ethalyzer capture
detail             Display detailed protocol information
display            Display packets even when writing to a file
display-filter      Display filter on frames captured
limit-captured-frames Maximum number of frames to be captured (default is 10)
raw                Hex/Ascii dump the packet with possibly one line summary
write              Filename to save capture to
```

```
N7K# ethalyzer local interface inband capture-filter "tcp port 5000" 50 write bootflash:test.cap
N7K# dir test.cap
26224      Jan12 18:40:08 2015  test.cap
```

# Packet Capture Tools: Ethalyzer cont...

- Detailed packet capture example:

```
N7K# ethalyzer local interface inband detail
```

```
Capturing on inband
```

```
Frame 1 (106 bytes on wire, 74 bytes captured)
```

```
Arrival Time: Feb 10, 2013 23:00:24.253088000
```

```
<snip>
```

```
Ethernet II, Src: 00:26:51:ce:0f:44 (00:26:51:ce:0f:44), Dst: 01:00:5e:00:00:0a (01:00:5e:00:00:0a)
```

```
Destination: 01:00:5e:00:00:0a (01:00:5e:00:00:0a)
```

```
Address: 01:00:5e:00:00:0a (01:00:5e:00:00:0a)
```

```
.... 1 .... = IG bit: Group address (multicast/broadcast)
```

```
.... 0 .... = LG bit: Globally unique address (factory default)
```

```
Source: 00:26:51:ce:0f:44 (00:26:51:ce:0f:44)
```

```
Address: 00:26:51:ce:0f:44 (00:26:51:ce:0f:44)
```

```
.... 0 .... = IG bit: Individual address (unicast)
```

```
.... 0 .... = LG bit: Globally unique address (factory default)
```

```
Type: IP (0x0800)
```

```
Internet Protocol, Src: 10.10.18.6 (10.10.18.6), Dst: 224.0.0.10 (224.0.0.10)
```

```
-----SNIP-----
```

# Packet Capture Tools: Cheat Sheet



Packet Capture Tool	When Useful	Comments
<b>SPAN</b>	<ul style="list-style-type: none"><li>• External Sniffer is available.</li><li>• Capture many packets (look at traffic rate)</li></ul>	<ul style="list-style-type: none"><li>• Local SPAN (external sniffer local to switch)</li><li>• ERSPAN (centralized sniffer, no need for directly connected sniffer)</li><li>• Exception SPAN: Captures dropped packets and identifies offending source/dst IPs</li></ul>
<b>ELAM</b> <b>ELAMe - script in 6.2(x)</b>	<ul style="list-style-type: none"><li>• Don't have external sniffer</li><li>• Quick verification of packet reception</li><li>• Looking for a specific packet</li></ul>	<ul style="list-style-type: none"><li>• Original ELAM can be tricky to configure</li><li>• source sys/elame.tcl – 6.2(x) makes is customer friendly</li></ul>
<b>Ethalyzer</b>	<ul style="list-style-type: none"><li>• No external sniffer</li><li>• Capture packets Rx/Tx CPU</li><li>• Can be used with ACL to capture data-plane packets</li></ul>	<ul style="list-style-type: none"><li>• Run in default vdc</li><li>• Can be viewed immediately or directed to file</li><li>• Capture and display filters</li></ul>

# Packet Capture Tools: Reference Info



- Ethalyzer:  
[http://www.cisco.com/c/en/us/products/collateral/switches/nexus-3000-series-switches/white\\_paper\\_c11-673817.html](http://www.cisco.com/c/en/us/products/collateral/switches/nexus-3000-series-switches/white_paper_c11-673817.html)
- SPAN:  
[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6\\_x/nx-os/system\\_management/configuration/guide/sm\\_nx\\_os\\_cg/sm\\_14span.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6_x/nx-os/system_management/configuration/guide/sm_nx_os_cg/sm_14span.html)
- [http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6\\_x/nx-os/system\\_management/configuration/guide/sm\\_nx\\_os\\_cg/sm\\_erspan.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6_x/nx-os/system_management/configuration/guide/sm_nx_os_cg/sm_erspan.html)
- ELAM:  
[https://supportforums.cisco.com/document/9880486/understanding-elam#Is\\_there\\_a\\_more\\_customer\\_friendly\\_way\\_to\\_utilize\\_ELAM](https://supportforums.cisco.com/document/9880486/understanding-elam#Is_there_a_more_customer_friendly_way_to_utilize_ELAM)

# Agenda

- NX-OS Troubleshooting Tools
- Control-Plane Troubleshooting
- Data Plane Troubleshooting
- Q&A / Conclusion
- Appendix

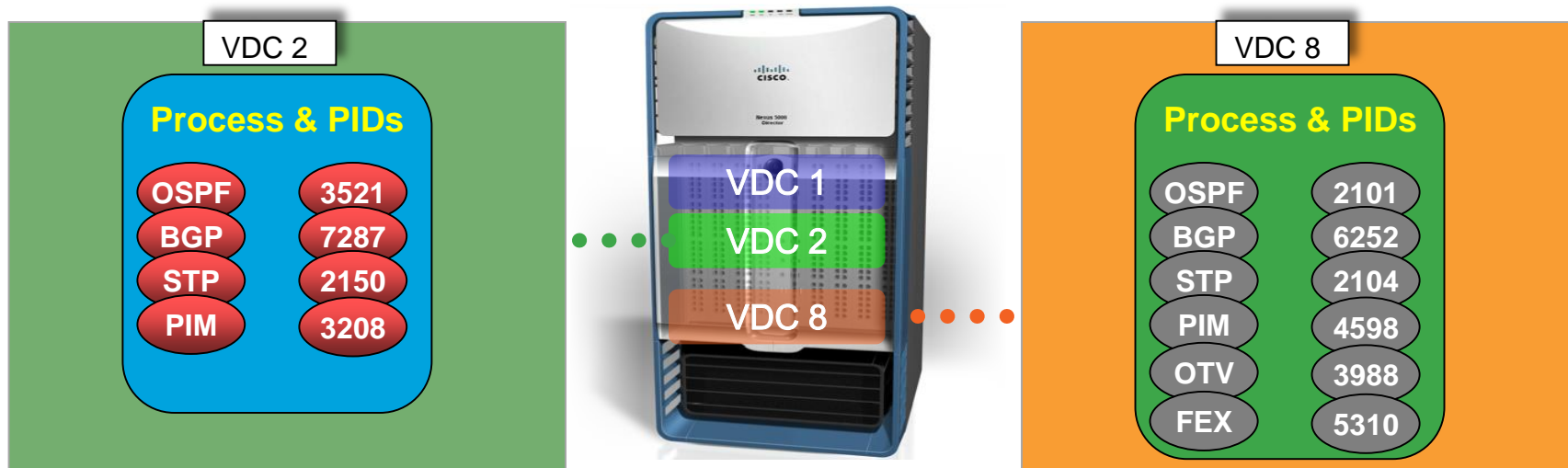
# *Control Plane Troubleshooting*

# Control-Plane Troubleshooting: Architecture

- N7K supports VDCs (Virtual Device Context)
- Control-Plane processes are separated on a per VDC basis
- VDCs still **share CPU and Inband** (path to CPU) resources
- Pre-emptive multitasking to prevent single process CPUHOGs
- Sup2/Sup2E add **“cpu-shares”** allowing priority to specific VDCs

	Sup1	Sup2	Sup2E
CPU	Dual-Core Xeon	Quad-Core Xeon	2 x Quad-Core Xeon
VDCs	4	4+1	8+1
CPU-Shares	Not Supported	Supported	Supported

# Control-Plane Troubleshooting: Architecture



- Processes in one VDC are independent of other VDCs

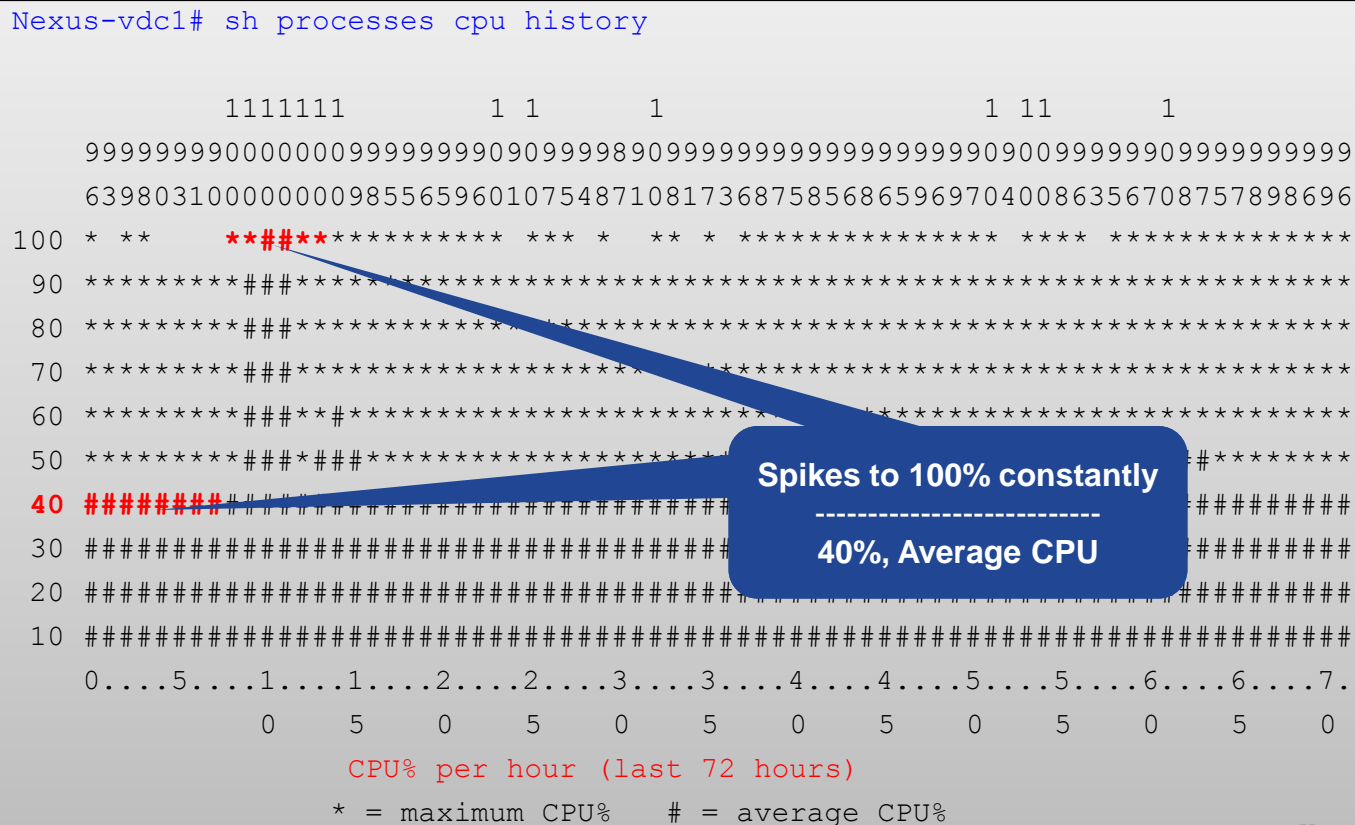


# Control-Plane Troubleshooting: Processes CPU

- Always understand CPU history and **baseline** for your system

- CPU history can be observed for past **60 sec, 60 min, and 72 hrs**

**Cisco** *live!*



# Control-Plane Troubleshooting: Processes CPU

Process in question  
OTV PID = 8273

```
N7K# sh processes cpu sort
```

CPU utilization for five seconds: 67%/3%; one minute: 62%; five minutes: 60%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
8273	1088950	2566908	2565	60.13%	60.31	30.5%	-	otv

Processes sorted  
high to low

```
N7K# show process cpu detailed 8273
```

CPU utilization for five seconds: 67%/3%; one minute: 63%; five minutes: 59%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
8273	200	20	10	60.63%	58.31%	0.21%	-	otv
8330	278710	716662	0	0.00%	0.00%	0.00%	-	otv:active-time
8333	158660	406757	0	48.21%	42.61%	0.01%	-	otv:otv-ip-udp

PIDs have sub-  
processes as well  
8273 → 8333

```
N7K# # show otv internal event-history events
```

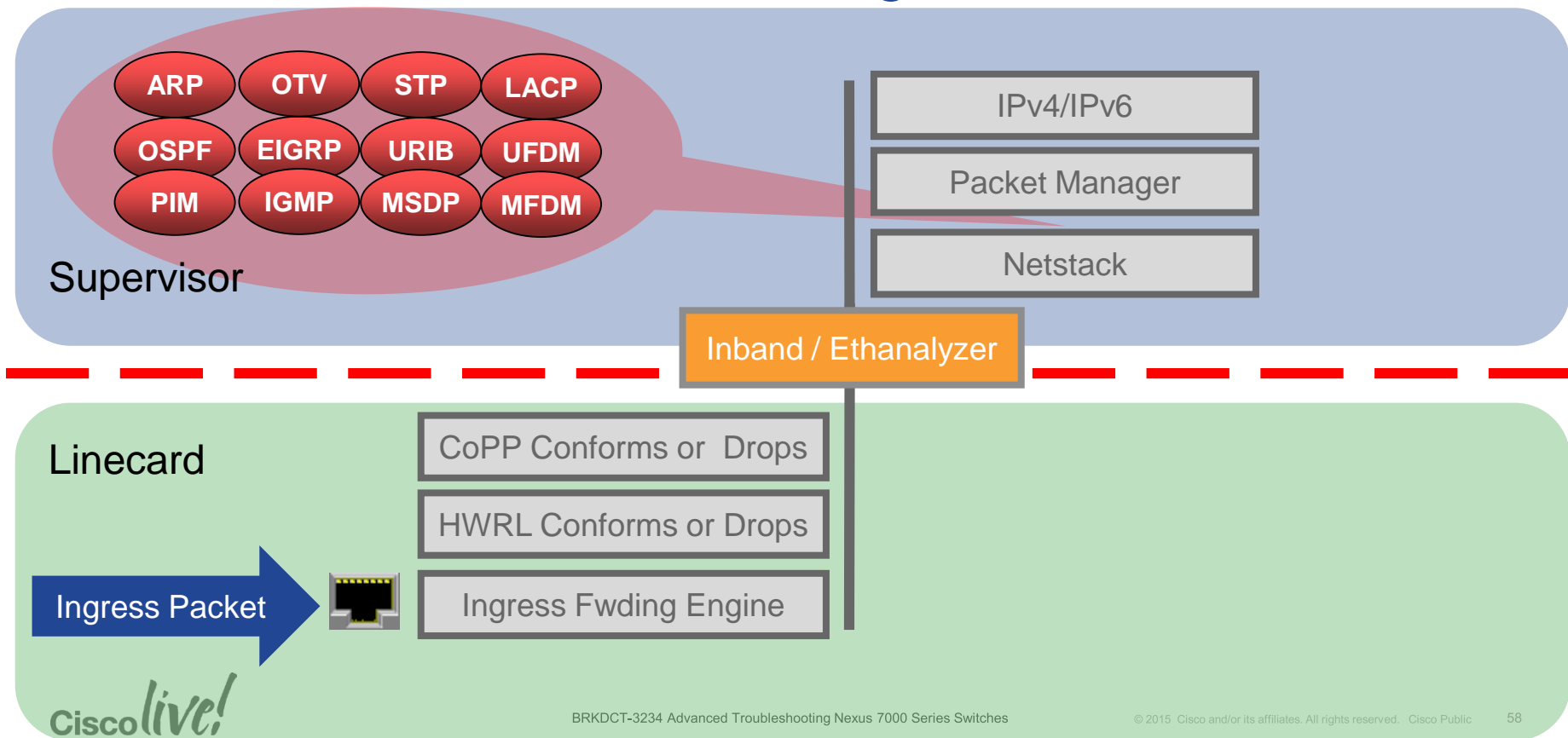
```
2015 Apr 4 03:36:57.236187 otv [8273]: [8333]: otv_ha_send_vd_sdb_batched_updates:
2015 Apr 4 03:36:27.203105 otv [8273]: [8333]: otv_ha_send_vd_sdb_batched_updates:
2015 Apr 4 03:35:57.163145 otv [8273]: [8333]: otv_ha_send_vd_sdb_batched_updates:
2015 Apr 4 03:35:27.133090 otv [8273]: [8333]: otv_ha_send_vd_sdb_batched_updates:
```

Event history gives  
further indication of  
what's occurring.

# Control-Plane Troubleshooting: Traffic

- Common traffic types driving CPU
  - ARP destined to CPU
  - Fragmented traffic
  - Glean
  - Broadcast / Link-Local mcast
- Identifying offending traffic
  - Control-Plane Policing (CoPP) & Hardware Rate-Limiters (HWRL)
    - Both implemented on **`per-forwarding engine`** basis and CLI available in **default vdc**
  - Inband counters
  - Ethalyzer

# Control-Plane Troubleshooting: Traffic Path



# Control-Plane Troubleshooting: HWRL Traffic

- Hardware Rate-Limiters are on by default & CLI available in **default vdc**
- Implemented on a '**per-FE**' basis

```
N7K# show hardware rate-limiter module 9
```

**Module: 9**

R-L Class	Config	Allowed	Dropped	Total
<hr/>				
<b>L3 mtu</b>	500	3012591	<b>3030853</b>	6043444
<snip>				
receive	30000	3011594	0	3011594
L2 port-sec	500	0	0	0
L2 mcast-snoop	10000	0	0	0

Type of Traffic

Configured Rate = pps

- Clear these counters via CLI: '`clear hardware rate-limiter <options>`'

# Control-Plane Troubleshooting: CoPP Traffic

- CoPP: more customizable and granular than HWRL
- 4 default profiles (dense, lenient, moderate, strict), but also **customizable**
  - Cannot modify default profiles. Copy default profile and then customize

```
N7K# copp copy profile moderate suffix mesau
N7K# (config)# show run copp
class-map type control-plane match-any copp-class-critical-mesau
  match access-group name copp-acl-bgp-mesau
  match access-group name copp-acl-rip-mesau
----- SNIP -----
class-map type control-plane match-any copp-class-monitoring-mesau
class-map type control-plane match-any copp-class-multicast-host-mesau
class-map type control-plane match-any copp-class-multicast-router-mesau
```

- Customize existing classes and policers, or create your own
  - **Match** (acl, protocol, etc) → **Classify** (class-map) → **Police** (policer)

# Control-Plane Troubleshooting: CoPP Traffic cont...

- Pay attention to violate packets AND conform packets

```
N7K# show policy-map interface control-plane | eg -i "drop|class|mod|violate|conform" | exc "violated 0"
  class-map copp-mcast-data-mesau (match-any)
    violate action: drop
  module 1:
    conformed 23085272 bytes,
  module 2:
    conformed 10681168 bytes,
  class-map class-default (match-any)
    violate action: drop
```

Conformed packets can be just as important as dropped packets, depending on the class (ex: TTL expiring classes)

- Configure syslog capable thresholds

```
Nexus-7k(config-pmap-c)# logging drop threshold [drop-count [level syslog-level]]
%COPP-5-COPP_DROPS5: CoPP drops exceed threshold in class: copp-system-class-critical, check show policy-map interface control-plane for more info
```

# Control-Plane Troubleshooting: Inband Traffic

- Key statistics when looking at the traffic hitting In-Band, destined for CPU

```
N7K# show hardware internal cpu-mac inband stats
```

```
<snip>
```

```
Packet rate limit ..... 32000 pps
```

```
Rx packet rate (current/max) 1173 / 2107 pps
```

```
Tx packet rate (current/max) 4558 / 1765 pps
```

```
...
```

```
Rate limit reached counter .. 0
```

```
...
```

```
Rx no buffers ..... 0
```

Max rx pps  
&  
# of times reached pps rx  
limit for Inband

```
N7K# show hardware internal cpu-mac inband events
```

```
6) Event:RX_PPS_MAX, length:4, at 387644 usecs after Thu Mar 26 14:22:00 2015  
new maximum = 2107
```

```
29) Event:RX_PPS_MAX, length:4, at 357606 usecs after Sat Mar 11 14:22:00 2015  
new maximum = 1615
```

Indicates when any new  
maximum Rx pkts occurred

To clear counters and events use:  
"test cpu-mac inband clear-counters"



# Control-Plane Troubleshooting: Inband Traffic

- We can look at various commands to identify offending interfaces...

```
Nexus-vdc1# show system internal pktmgr interface
Ethernet2/23, ordinal: 201 Hash_type: 1
SUP-traffic statistics: (sent/received)
  Packets: 842445 / 11147
  Bytes: 71194935 / 2784182
  Instant packet rate: 0 pps / 8 pps
  Packet rate limiter (Out/In): 0 pps / 0 pps
  Average packet rates(1min/5min/15min/EWMA):
  Packet statistics:
    Tx: Unicast 523, Multicast 841905, Broadcast 17
    Rx: Unicast 521, Multicast 10615, Broadcast 11
```

Interface and VDC  
Granularity for CPU bound  
pkts

```
Nexus-vdc1# show system internal pktmgr internal vdc inband | exc "In VDC"
Interface          Src Index      VDC ID      Packet rcvd
-----
Ethernet8/1        0x30           2           42557
Ethernet8/30       0x4d           2          1552046
Ethernet8/48       0x5f           2          6103038
Ethernet2/23       0x17e          2           13352
Ethernet4/1        0x180          2          642228
```

Quick Snapshot of all  
interfaces and packets  
destined for CPU

# Control-Plane Troubleshooting: Cheat-Sheet



- ethalyzer local interface inband ...
- show tech / tac pac
- show tech <feature> <<<< feature driving cpu from show proc cpu
- show tech netstack
- show proc cpu history
- show proc cpu sort | exc 0.00
- show policy-map interface control-plane | eg "drop|class-map|module" | ex "violated 0| conform"
- show hardware rate-limiter
- show hardware internal cpu-mac inband stats
- show hardware internal cpu-mac inband events
- show system internal pktmgr interface
- show system internal pktmgr internal vdc inband

# Agenda

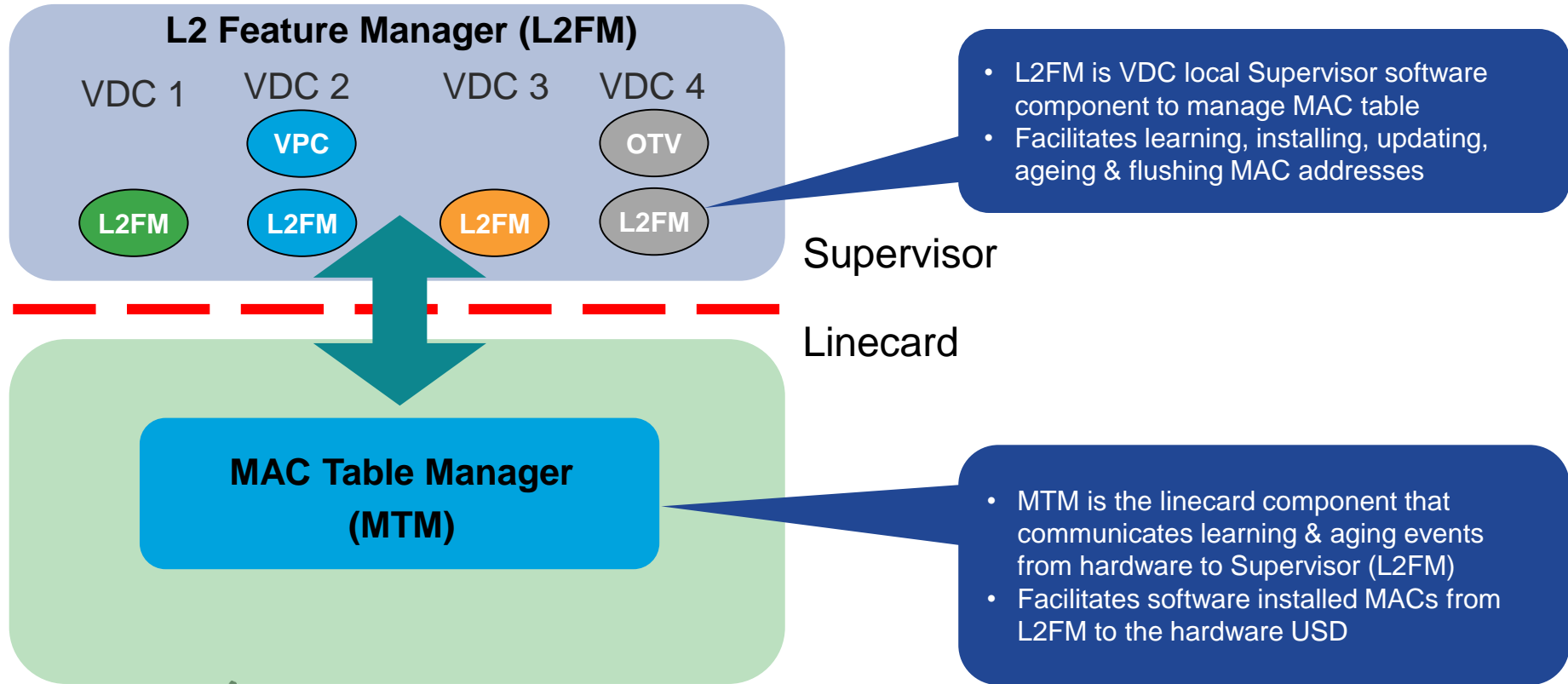
- NX-OS Troubleshooting Tools
- Control-Plane Troubleshooting
- Data-Plane Troubleshooting
- Q&A / Conclusion
- Appendix

# *Data-Plane Troubleshooting: L2 Programming Verification*

# Layer 2 Data Plane Troubleshooting: Common Symptoms

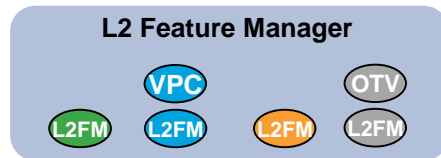
- Unicast flooding
  - Bandwidth spikes
  - Unexpected traffic on interfaces
  - Congestion drops
  - Mixed Chassis **(The smallest TCAM sets theoretical MAC limits per VDC)**
- Black-holing traffic
  - L2 Traffic destined out wrong interface

# Layer 2 Data Plane Troubleshooting:



# Layer 2 Data Plane Troubleshooting: Verifying SW

- L2FM is the Supervisor's perspective of the MAC table
- L2FM can pass this information to other features such as OTV



```
N7K# show mac address-table vlan 500 address 0011.2233.4455
Note: MAC table entries displayed are getting read from software.
Use the 'hardware-age' keyword to get information related to 'Age'
```

This command is looking at Supervisor perspective

Legend:

\* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC  
age - seconds since last seen, + - primary entry using vPC Peer-Link,  
(T) - True, (F) - False, ~~~ - use 'hardware-age' keyword to retrieve

age info

VLAN	MAC Address	Type	age	Secure	NTFY	Ports/SWID.SSID.LID
* 500	0011.2233.4455	dynamic	~~~	F	F	Po1995

How was this MAC learned?

```
N7K# show system internal l2fm event-history debugs | include 0011.2233.4455
[snip]
[104] l2fm_macdb_insert(5817): slot 1 fe 1 mac 0011.2233.4455 vlan 500 flags 0x107 hints 0 E8 NL
lc : 0x160007ca
```

Front Panel Port represented in ifIndex

# Layer 2 Data Plane Troubleshooting: Verifying SW

- Debug MACDB tracks current and **historical state of MAC address** learned, updated or flushed

## L2 Feature Manager



```
N7K# show system internal l2fm l2dbg macdb address 0011.2233.4455
```

### Legend

-----  
Db: 0-MACDB, 1-GWMACDB, 2-SMACDB, 3-RMDB, 4-SECMACDB  
Src: 0-UNKNOWN, 1-L2FM, 2-PEER, 3-LC, 4-HSRP  
5-GLBP, 6-VRRP, 7-STP, 8-DOTX, 9-PSEC 10-CLI 11-PVLAN  
12-ETHPM, 13-ALW LRN, 14-Non PI MOD, 15-MCT DOWN, 16 - SDB  
17-OTV, 18-Debounce Timer, 19-AM, 20-PCM\_DOWN, 21 - MCT\_UP  
22-L2VPN, 23-EFP, 24-DRV  
Slot:0 based for LCS 19-MCEC 20-OTV/ORIB

Use the Legend to determine where the mac was installed from

VLAN: 500 MAC: 0011.2233.4455

“how” and “where” mac was learned

Time	If/swid	Db	Op	Src	Slot	FE
Mon Apr 13 01:21:00 2015	0x160007ca	0	INSERT	3	1	1
Mon Apr 13 01:45:39 2015	0x160007ca	0	FLUSH	10	0	15
Mon Apr 13 01:45:39 2015	0x160007ca	0	DELETE	0	0	15
Mon Apr 13 01:45:40 2015	0x160007ca	0	INSERT	3	1	1

Installed from LC 2 FE 2  
(zero based)

```
N7K# show system internal pixm info | grep 0x160007ca
```

PC_TYPE	PORT	LTL	RES_ID	LTL_FLAG	CB_FLAG	MEMB_CNT
Normal	Po1995	0x041e	0x160007ca	0x00000000	0x00000002	1



# Layer 2 Data Plane Troubleshooting: Verifying HW

- Everything you need to know about HW programming is in MTM
- MTM informs L2FM of a mac learn updating software.

MAC Table Manager

Linecard

```
N7K# show hardware mac address-table 2 vlan 500 address 0011.2233.4455
```

FE	Valid	PI	BD	MAC	Index	Stat	SW	Modi	Age	Tmr	GM	Sec	TR	NT	RM	RMA	Cap	Fld	Always
						ic		fied	Byte	Sel		ure	AP	FY		TURE			Learn
0	1	0	70	0011.2233.4455	0x0041e	0	0x003	0	6	1	0	0	0	0	0	0	0	0	0
1	1	1	70	0011.2233.4455	0x0041e	0	0x003	0	0	1	0	0	0	0	0	0	0	0	0

Forwarding  
Engine

"Primary Input"

MAC Address

LTL Index -  
Maps to  
port

Entry Age

```
N7K# show system internal pixm info ltl 0x041e
```

```
Normal Po1995 0x041e 0x160007ca 0x00000000 0x00000002 1
```

# Layer 2 Data Plane Troubleshooting: Verifying HW

- PIXM (Port Index Management) manages all internal indices used to forward packets
  - LTL, IF-Index, BD, CBL, etc...

```
N7K# sh system internal pixm info ltl 0x41e
```

PC_TYPE	PORT	LTL	RES_ID	LTL_FLAG	CB_FLAG	MEMB_CNT
Normal	Po1995	0x041e	0x160007ca	0x00000000	0x00000002	1

Member rbh rbh cnt  
Eth2/24 0x000000ff 0x08

CBL Check States: Ingress: Enabled; Egress: Enabled

VLAN| BD | BD-St | CBL St & Direction:

1	0x0	EXCLUDE_IF_FROM BD	BLOCKING (Both)
500	0x46	INCLUDE_IF_IN BD	FORWARDING (Both)
502	0x48	INCLUDE_IF_IN BD	FORWARDING (Both)

Member info

Type	LTL
PORT_CHANNEL	Po1995
MCAST_GROUP	0x7fe8
MCAST_GROUP	0x7bdc
FLOOD_W_FPOE	0x8046
FLOOD_W_FPOE	0x8048

Port <-> LTL mapping &  
If-Index information

Color Blocking Logic is the HW  
version of STP State

PIXM includes interfaces into the  
Flood List for BD if in a *forwarding*  
state

# Layer 2 Data Plane Troubleshooting: Verifying HW

- VLAN is unique to the VDC
- BD (Bridge Domain) is unique to the system
- PIXM has VLAN to BD mapping

```
N7K# show system internal pixm info vlan-bd-db
BD info for VDC: 6
```

VLAN	BD	BD LTL	BD flag
1	65	0x8041	0x00
500	70	0x8046	0x00
501	71	0x8047	0x00
502	72	0x8048	0x00

The LTL used to flood to the VLAN

- PIXM also maintains the BD Flood Index (all ports in the VLAN)

```
N7K# show system internal pixm info ltl 0x8048
LTL      res_id      ltl_flag      cb_flag      MI[0]
0x8048    0x00000000    0x00000000    0x00000000    0x00cb
```

Member info

IFIDX	LTL
Po1995	0x041e
Po211	0x041a

Flood to VLAN includes these ports

# Layer 2 Data Plane Troubleshooting: Verifying HW

- All in one command to display discrepant, missing, or extra MAC addresses between the supervisor and the module.

```
N7K# show forwarding consistency l2 2
```

Note: MAC table entries displayed are getting read from software.  
Use the 'hardware-age' keyword to get information related to 'Age'

Legend:

\* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC  
age - seconds since last seen, + - primary entry using vPC Peer-Link,  
(T) - True, (F) - False , ~~~ - use 'hardware-age' keyword to retrieve age info

## Missing entries in the MAC Table

VLAN	MAC Address	Type	age	Secure	NTFY	Ports/SWID.SSID.LID
G	-	0022.557a.7446	static	-	F	F sup-eth1 (R) (Eth2/23)

## Extra and Discrepant entries in the MAC Table

VLAN	MAC Address	Type	age	Secure	NTFY	Ports
-----	-----	-----	-----	-----	-----	-----

# Layer 2 Data-Plane Troubleshooting: Cheat-Sheet



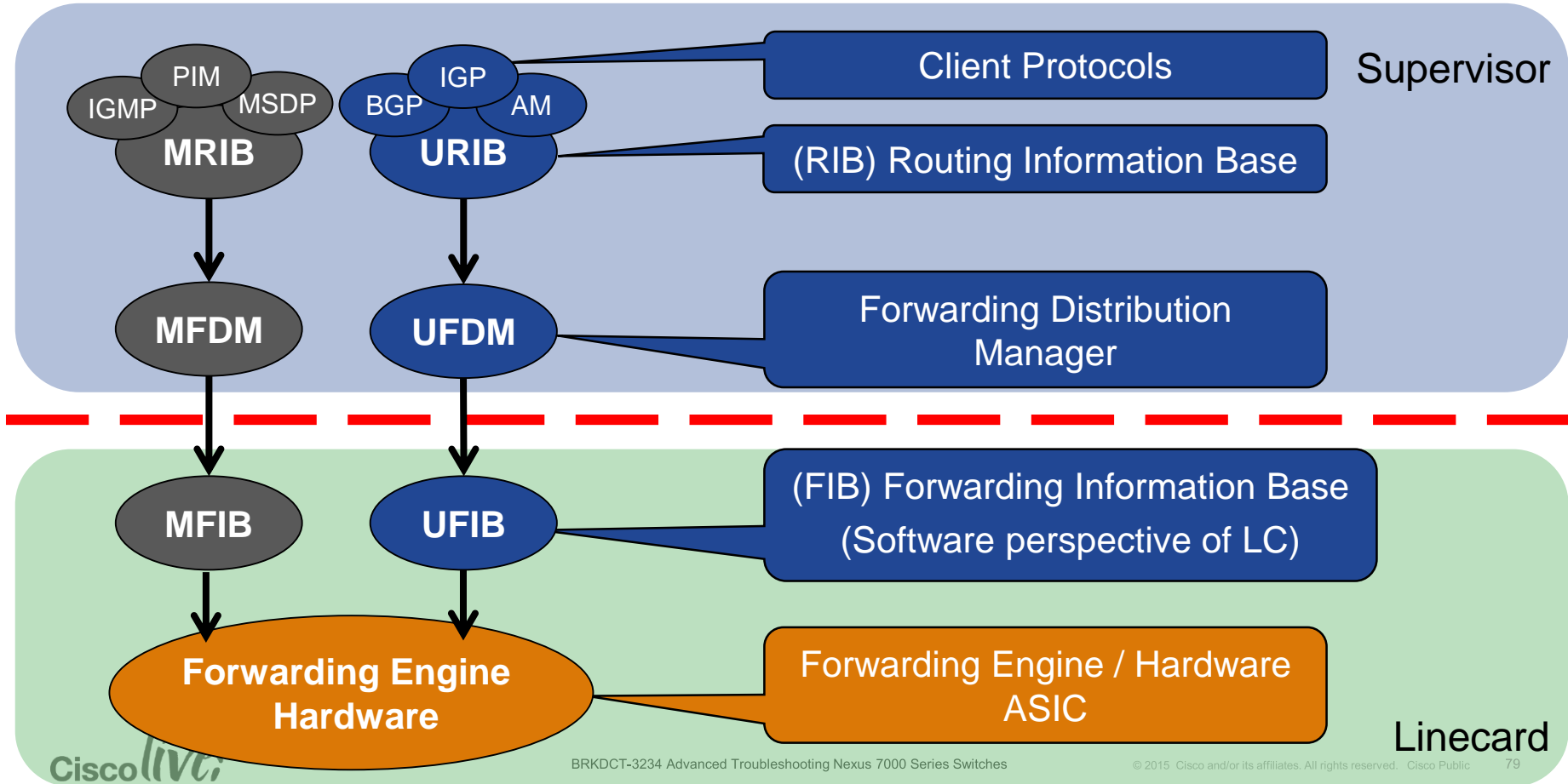
- Show mac address-table
- Show system internal l2fm l2dbg macdb address [mac address]
- show hardware mac address-table [LC#] [mac address]
- show forwarding consistency l2 <mod>
- show tech-support L2FM detail (Always go for the detailed output)
  - Includes MTM information
  - Includes all MACDB L2 debug outputs
- Show tech-support forwarding l2 unicast
- Show tech-support vlan

# *Data-Plane Troubleshooting: L3 Programming Verification*

# Layer 3 Data Plane Troubleshooting: Common Symptoms

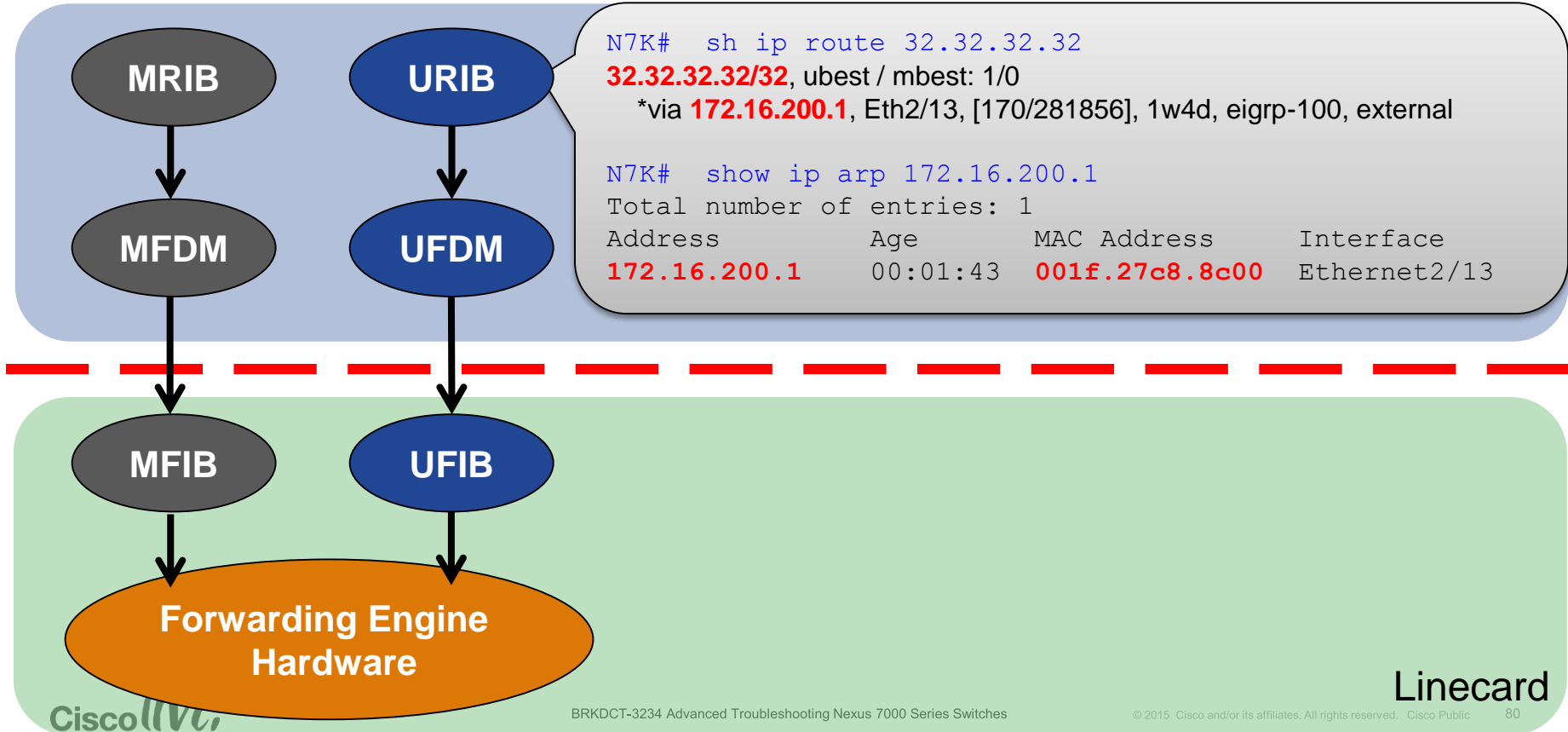
- Complete Blackholing traffic (for L3 flows)
  - **Not Intermittent traffic loss**
  - Usually a pattern
    - Same next-hop (unicast)
    - Same mroute or OIF (multicast)
    - Group of physical interfaces all mapping to the same forwarding engine (FE)

# Layer 3 Data Plane Troubleshooting: Components

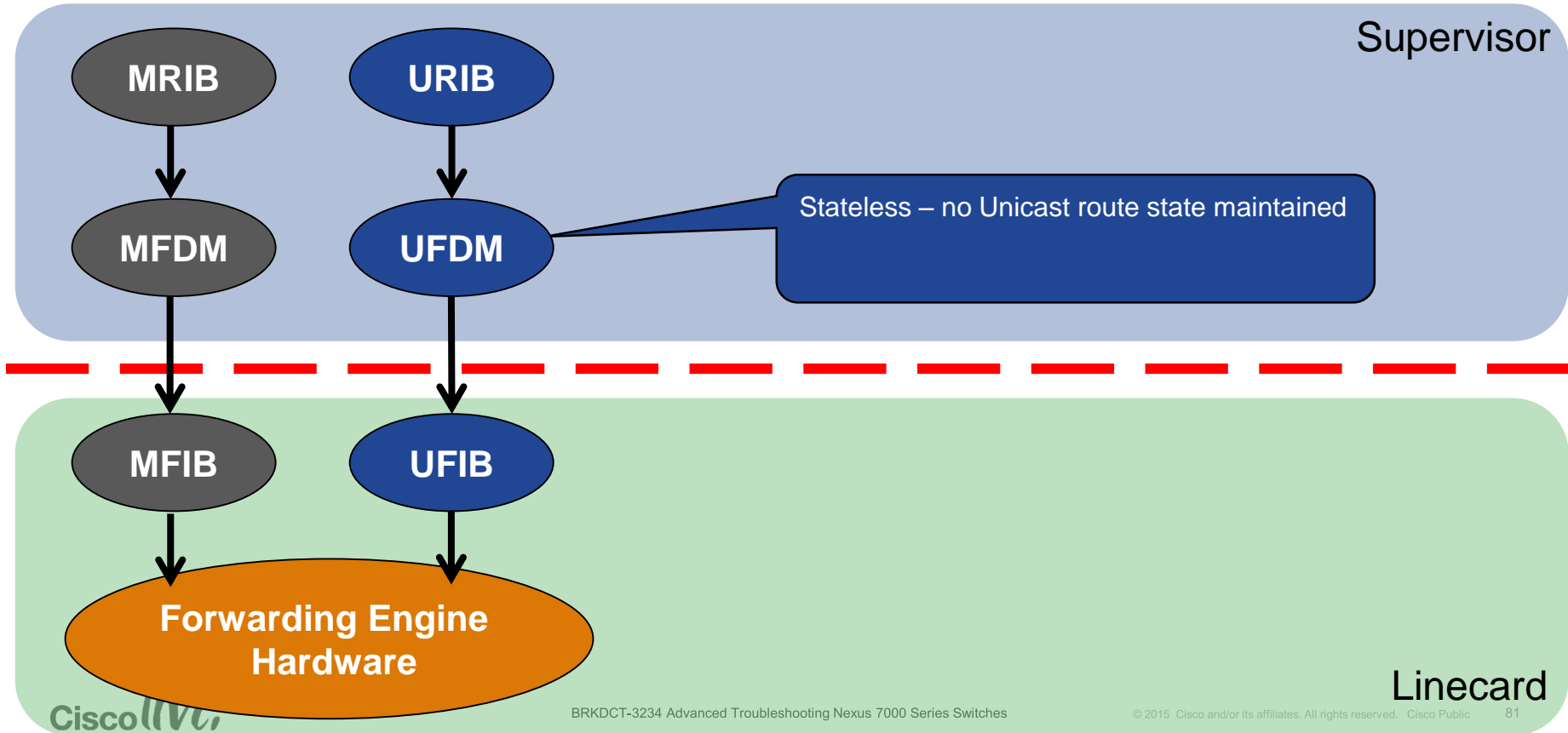




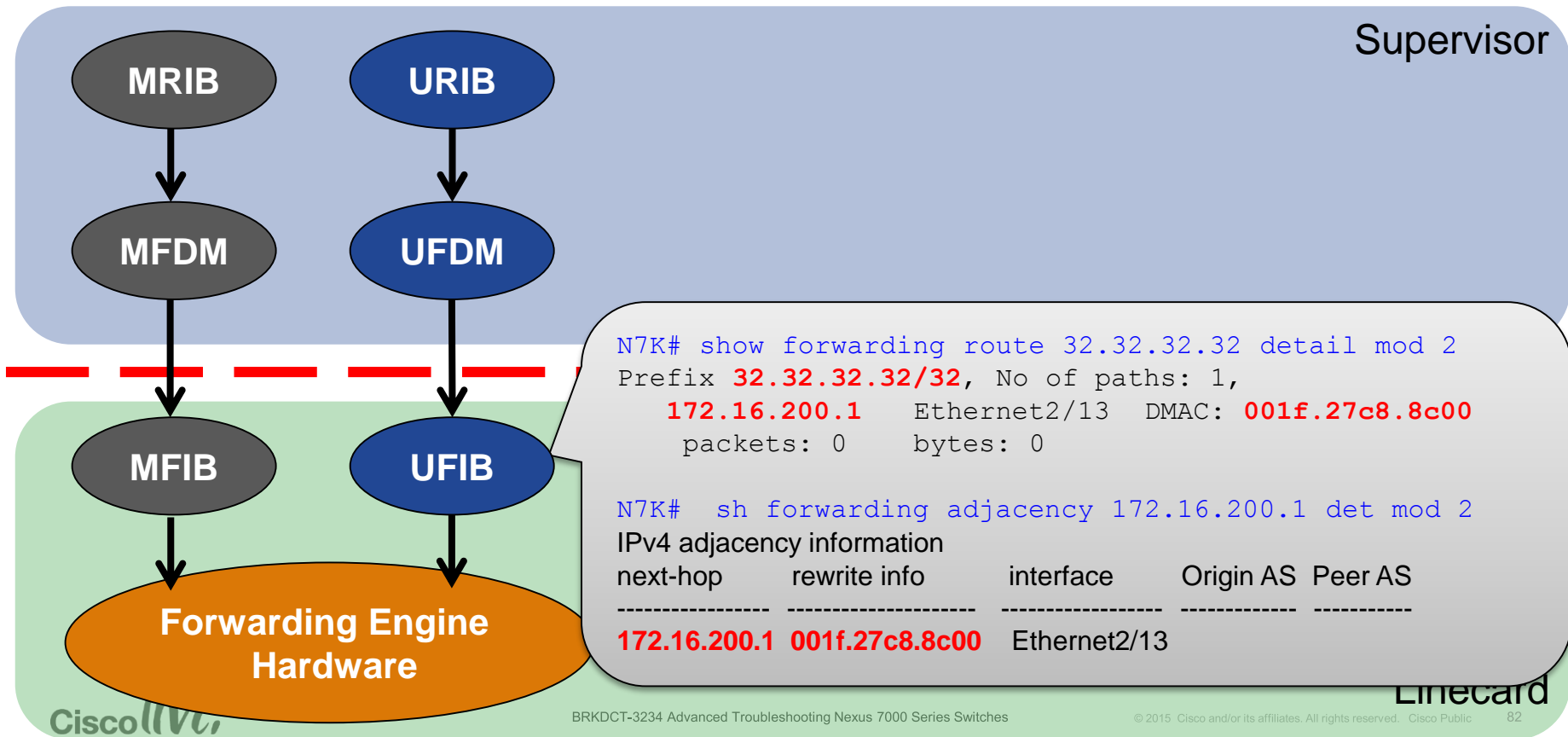
# Layer 3 Data Plane Troubleshooting: Unicast



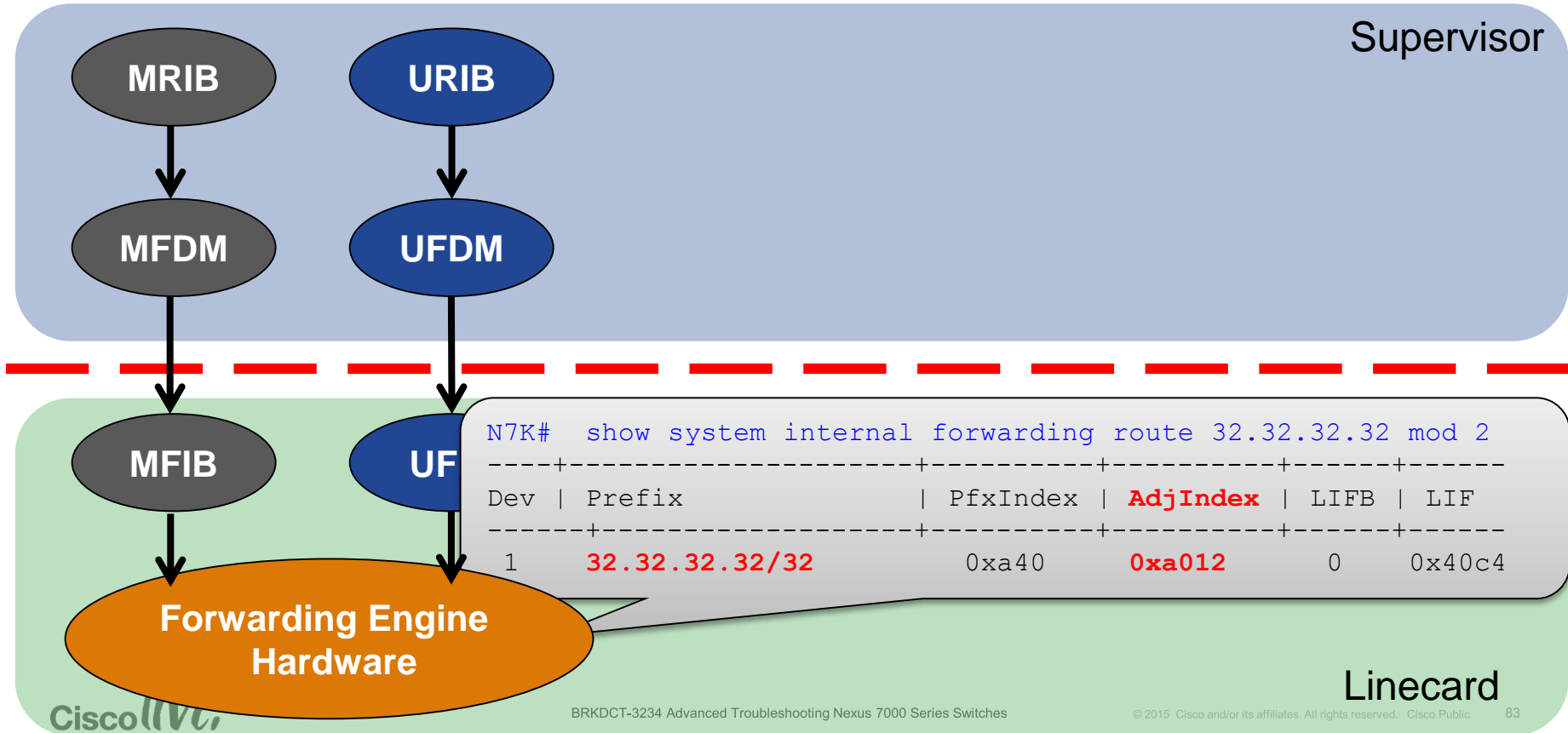
# Layer 3 Data Plane Troubleshooting: Unicast



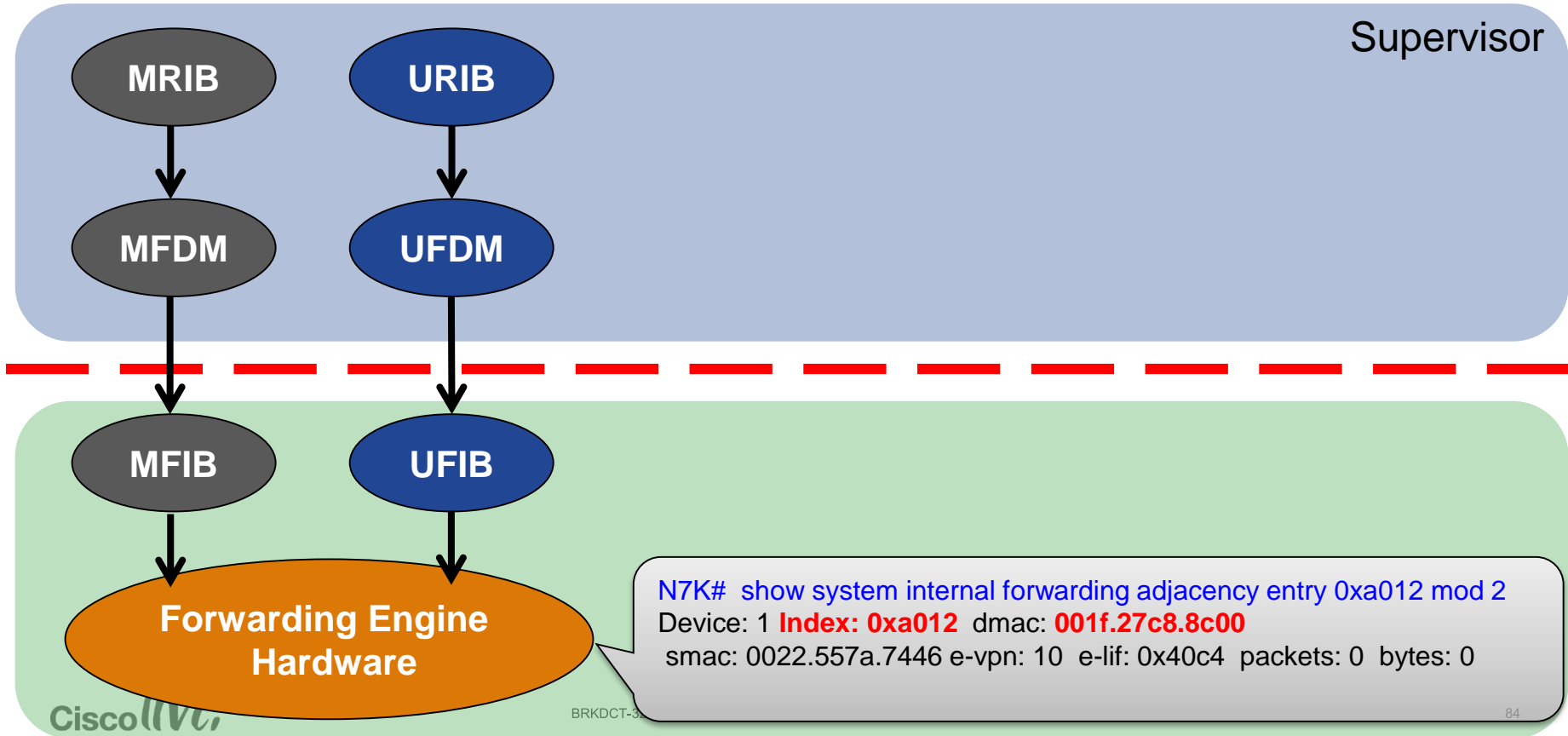
# Layer 3 Data Plane Troubleshooting: Unicast



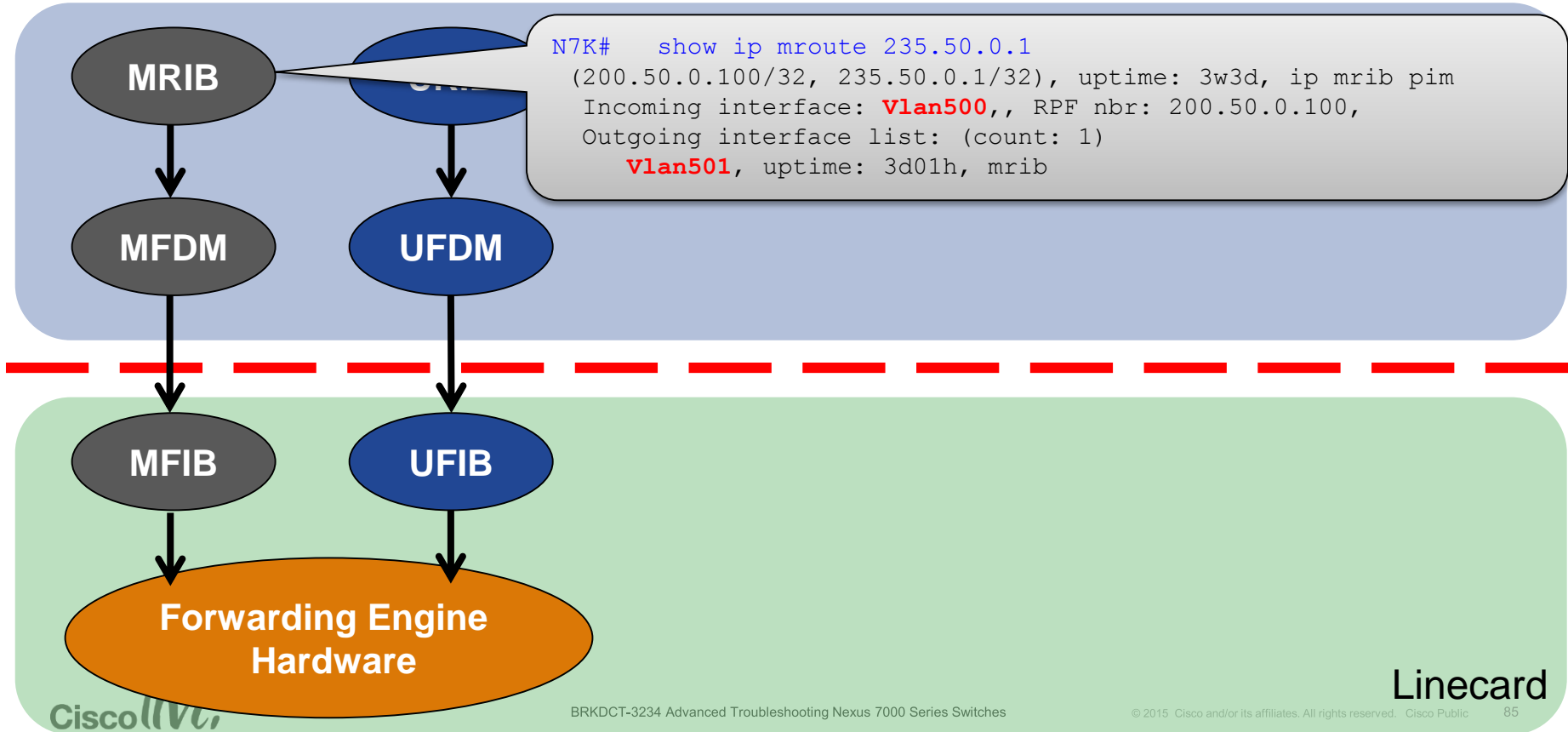
# Layer 3 Data Plane Troubleshooting: Unicast



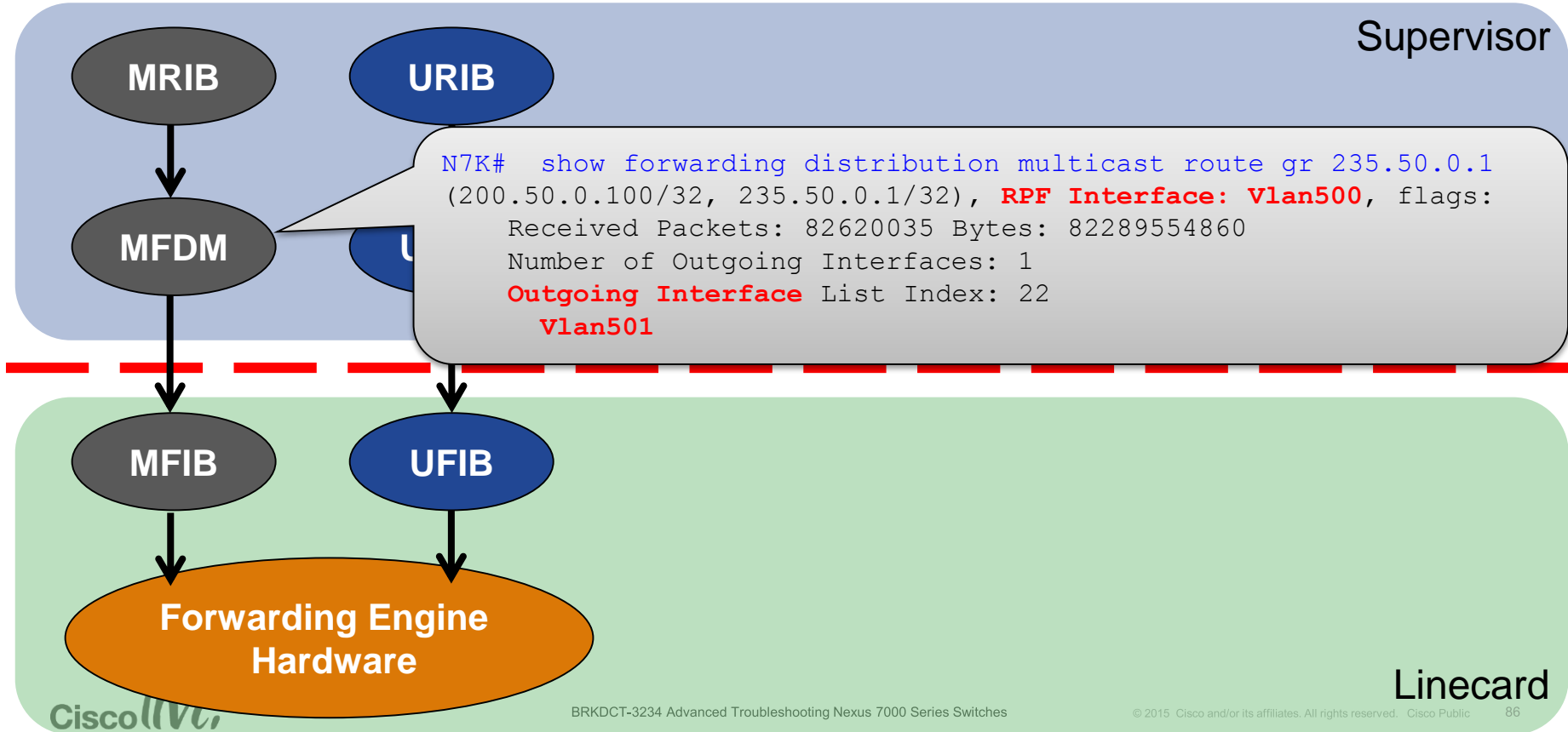
# Layer 3 Data Plane Troubleshooting: Unicast



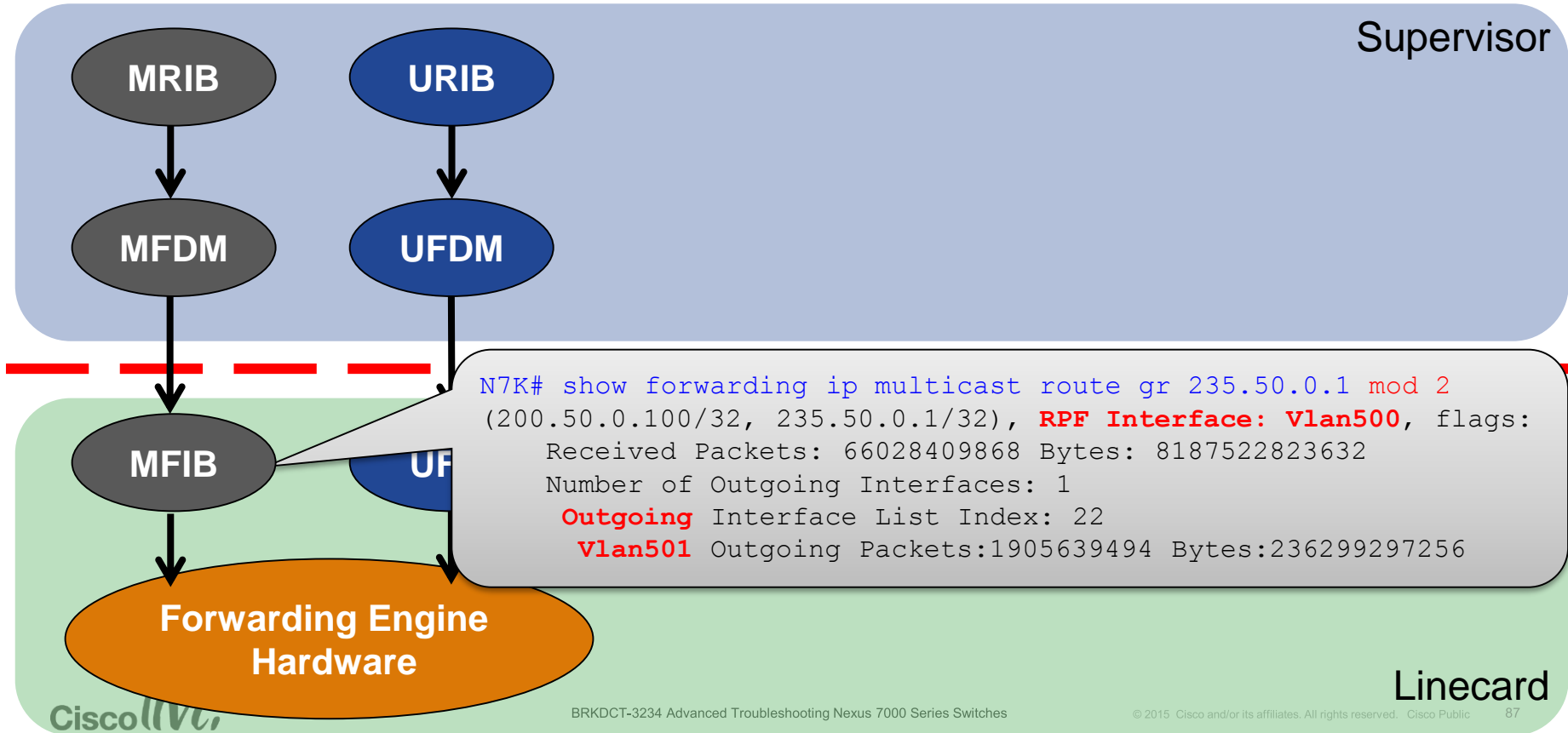
# Layer 3 Data Plane Troubleshooting: Multicast



# Layer 3 Data Plane Troubleshooting: Multicast

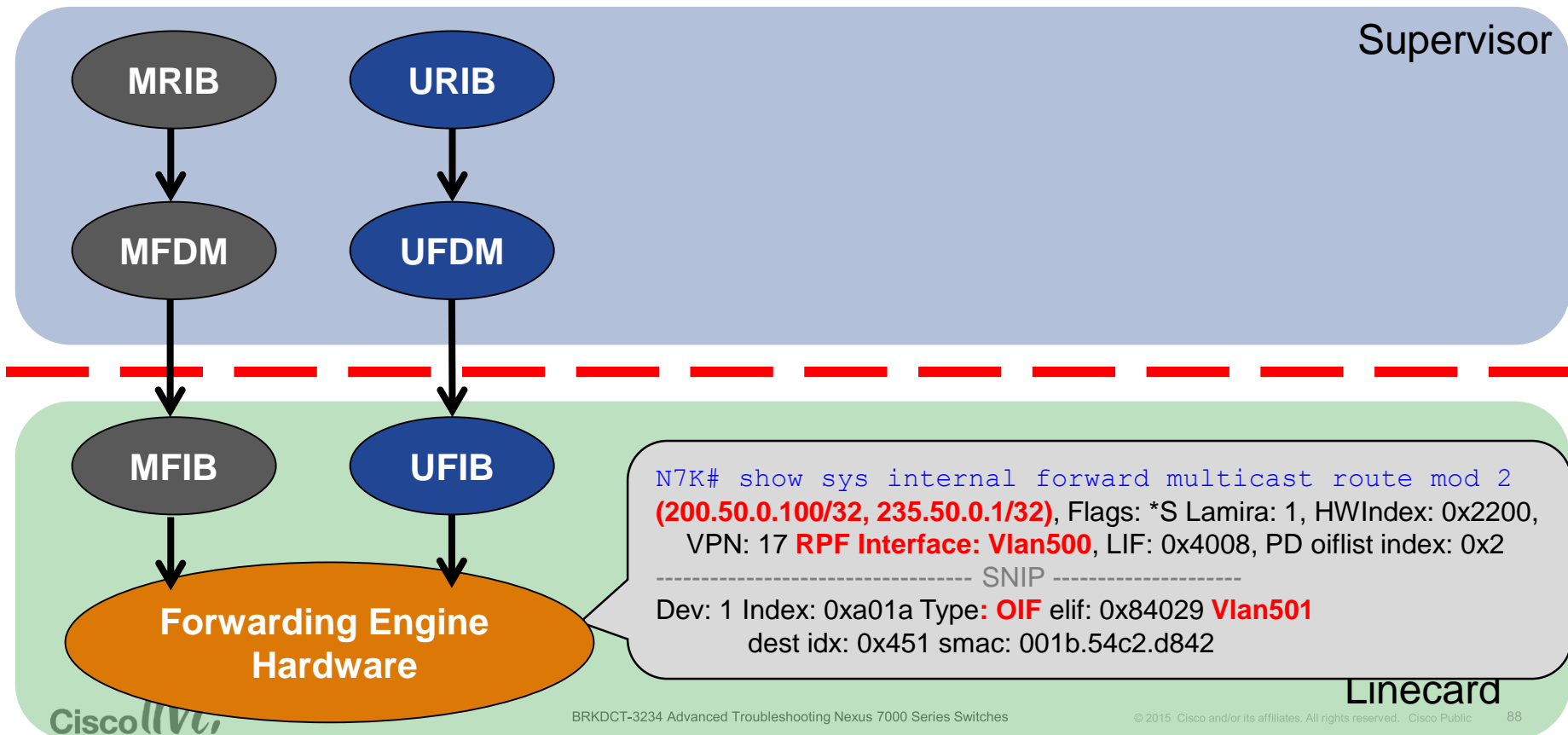


# Layer 3 Data Plane Troubleshooting: Multicast





# Layer 3 Data Plane Troubleshooting: Multicast



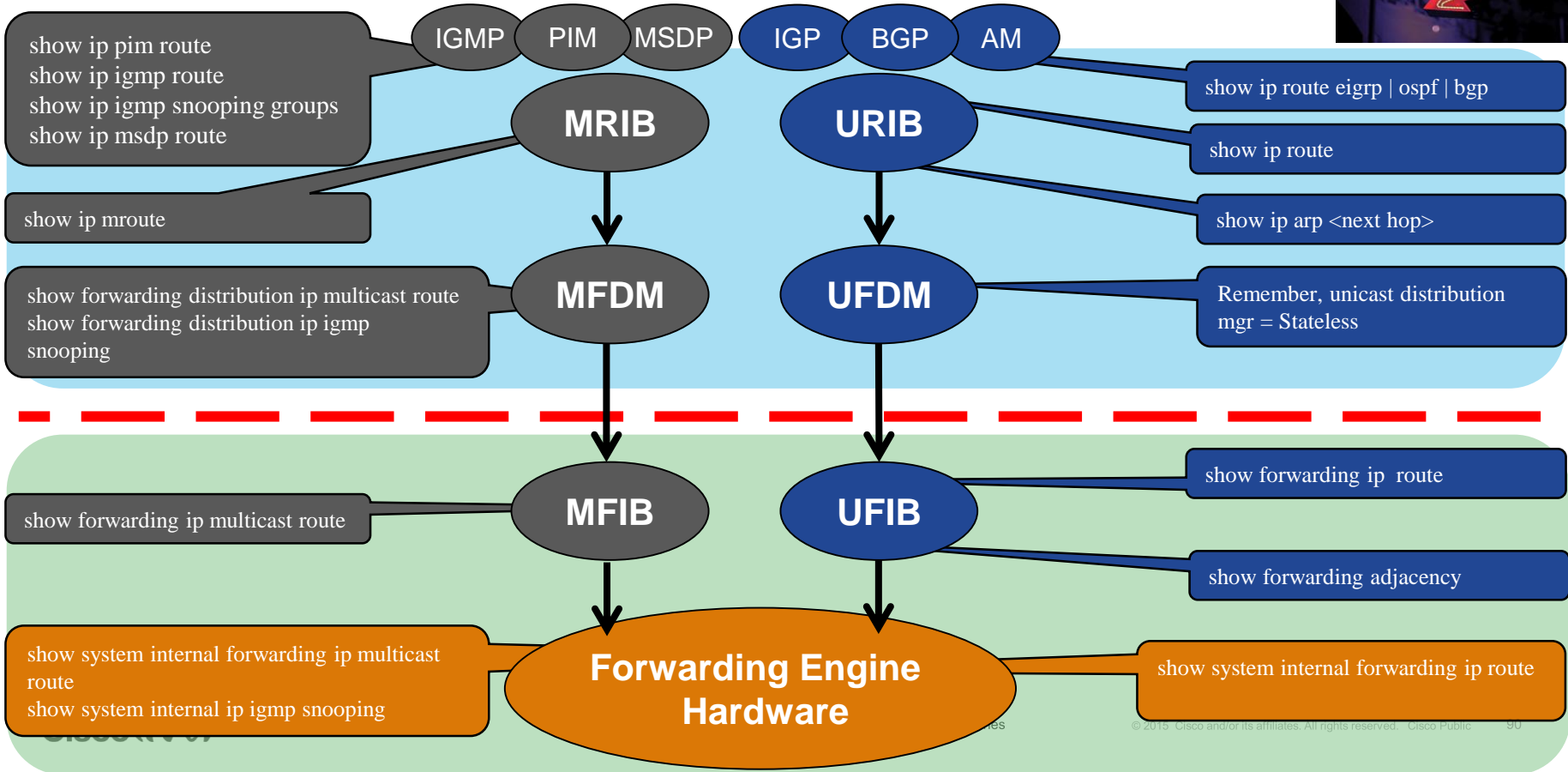


Available Friday 4:00 PM!!!

Matt Esau

555-867-5309

# Layer 3 Data Plane Troubleshooting: Cheat-Sheet



# Layer 3 Data-Plane Troubleshooting: Cheat-Sheet



## Multicast

- Show tech-support ipv4 multicast  
(show techs included)
  - Show tech-support msdp
  - Show tech-support mfwd
  - Show tech-support routing ip multicast
  - Show tech-support ip pim
  - Show tech-support routing ip unicast
- Show tech-support m2rib
- Show tech-support m2fib
- Show tech-support forwarding I2 multicast
- Show tech-support forwarding I3 multicast

## Unicast

- Show tech-support (client protocol) ex. OSPF
- Show tech-support arp
- Show tech-support adjmgr
- Show tech-support u2rib
- Show tech-support u2fib
- Show tech-support forwarding I3 unicast detail

# *Data-Plane Troubleshooting: Intermittent Packet Loss*

# Data-Plane Troubleshooting Intermittent Packet Loss: CoPP and HWRLs



- Control Plane Policing (CoPP) and Hardware Rate-Limiters (HWRL) are **common, but overlooked** areas for packet drops.
- Data Packets can be punted to CPU as well
- Both CoPP and HWRL counters are on a **“per-FE”** basis.
- Don’t forget Ethalyzer can help to identify CPU bound packets

# Data-Plane Troubleshooting Intermittent Packet Loss: Linecard Drop Counters

## Key terms when discussing N7K Linecard Drop Counters

- **Credited traffic:** refers to unicast traffic that is centrally arbitrated by the system.
  - Ingress card will drop packet if no credit is available at egress port.
- **Uncredited traffic:** refers to multi-destination traffic (mcast, bcast, unk ucast).
  - NOT centrally arbitrated, so ingress cards will continue to send, resulting in egress drops
- F-Series is pure ingress-buffered architecture, using VOQ
  - Ingress discards for Credited, output drops for Uncredited traffic (not through VOQ)
- M-Series is a hybrid ingress & egress buffered I/O module architecture, using VOQ plus ingress/egress port buffers
  - VOQ / Ingress discards for Credited, Output Drops for Credited and Uncredited traffic

# M1 & M2 Packet Counters



**M2 10G / 40G / 100G**





# Data-Plane Troubleshooting Intermittent Packet Loss: M-Series Per-Queue Counters

- Per-Interface and Per-Queue counters for Ucast & Mcast...

```
N7K# show policy-map interface ethernet x/y | eg -i class|drop
Class-map (queuing):  in-q1 (match-any)
  queue dropped pkts : 0
Class-map (queuing):  in-q-default (match-any)
  queue dropped pkts : 0
Class-map (queuing):  out-pq1 (match-any)
  queue dropped pkts : 0
Class-map (queuing):  out-q2 (match-any)
```

Per-queue drops help identify which CoS traffic is affected

```
N7K# show system internal qos queuing stats interface x/y
```

## Receive queues

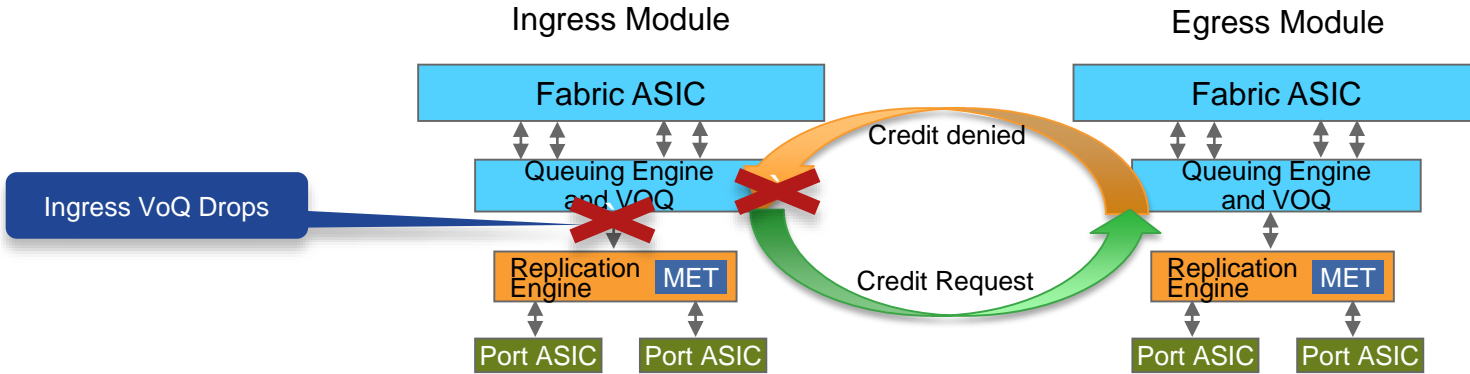
```
-----
Queue 2q4t-in-q-default
  Total packets      0
  WRED drops         0
  Taildrop drops     0
```

## Transmit queues

```
-----
Queue 1p3q4t-out-q-default
  Total packets      0
  WRED drops         0
  Taildrop drops     0
```

Shows more detailed queue counter/drop info

# Data-Plane Troubleshooting Intermittent Packet Loss: M-Series Unicast Ingress VoQ Drops



- Unicast drops at **Ingress VoQ**

```
N7K# slot 3 show hardware internal statistics device gengine congestion port 2
83 BA hard drop count for QOS3 0000000074393315 1-4 I1
87 BA WRED drop count for QOS3 0000000002390602 1-4 I1
```

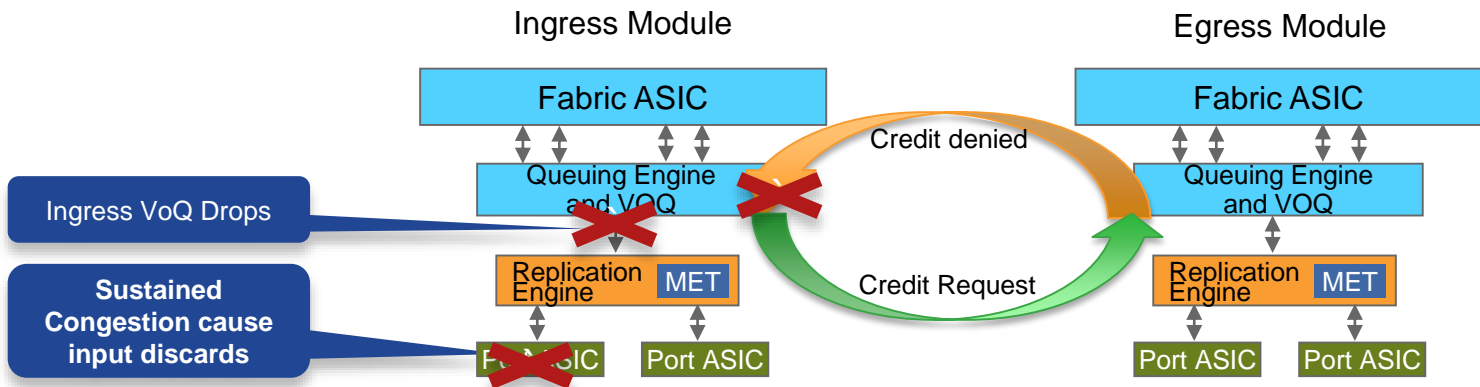
M1

```
N7K# slot 2 show hardware skytrain queuing drops
```

Source Intf	Traffic Type	Drop Reason	Count
eth 2/23	Unicast	<b>VOQ tail-drop</b>	<b>94980</b>

M2

# Data-Plane Troubleshooting Intermittent Packet Loss: M-Series Unicast Ingress VoQ Drops

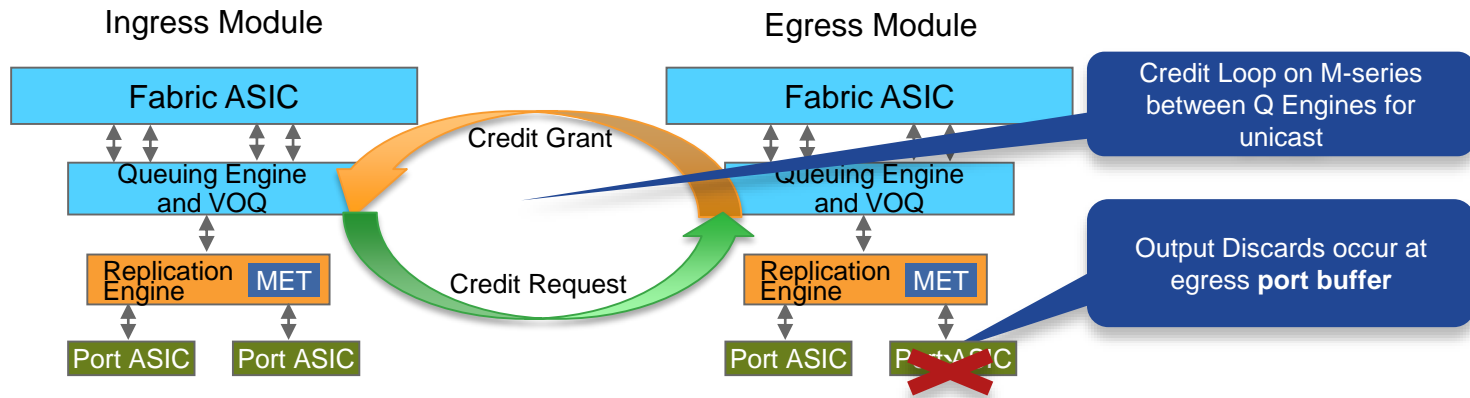


- Unicast also utilizes **Ingress Buffers**, and can drop at the port ASIC

```
N7K# show int e 2/23 | eg -i discard
0 input with dribble 14921 input discard
```

```
N7K# show policy-map interface ethernet 2/23 input | eg -i class|drop
Class-map (queuing):  in-q1 (match-any)
  queue dropped pkts : 0
Class-map (queuing):  in-q-default (match-any)
  queue dropped pkts : 15327
```

# Data-Plane Troubleshooting Intermittent Packet Loss: M-Series Egress Port Drops (Ucast & Mcast)

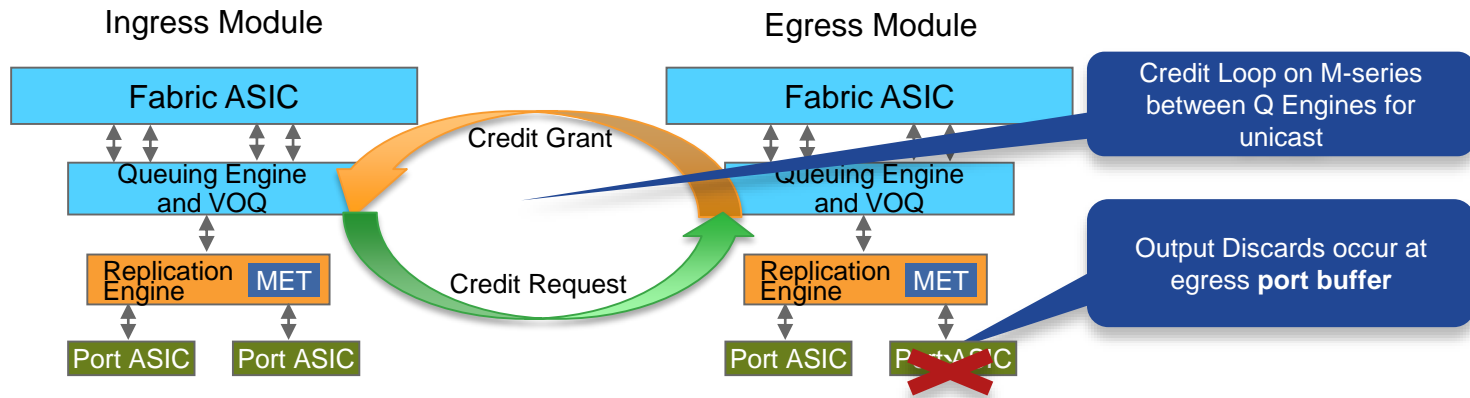


- Oversubscription of the **egress port** is identifiable as **OutDiscards** for Ucast & Mcast

```
N7K# show interface e8/48 counters errors
```

Port	Align-Err	FCS-Err	Xmit-Err	Rcv-Err	UnderSize	OutDiscards
Eth8/48	0	0	0	0	0	310000884

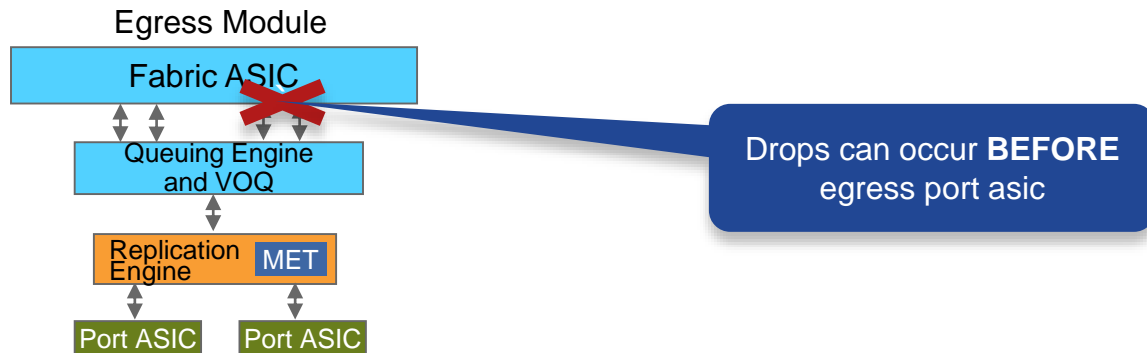
# Data-Plane Troubleshooting Intermittent Packet Loss: M-Series Egress Port Drops (Ucast & Mcast)



- Also viewable via **policy-map counters**

```
N7K# show policy-map interface ethernet 8/48 output
<snip>
Class-map (queuing):  out-q-default (match-any)
  queue-limit percent 82
  bandwidth remaining percent 25
  queue dropped pkts : 326893332
```

# Data-Plane Troubleshooting Intermittent Packet Loss: M-Series Multicast Egress Drops



- If Mcast traffic is great enough, **may not see drops in policy-map or as outdiscards.**

```
N7K# slot 8 show hardware internal statistics device qengine congestion port 32 | grep MB
320 MB FI0 packet drop count 0000000005115220 1-48 I1
321 MB FI1 packet drop count 0000000000212321 1-48 I1
322 MB FI2 packet drop count 0000000005541641 1-48 I1
323 MB FI3 packet drop count 0000000010050874 1-48 I1
```

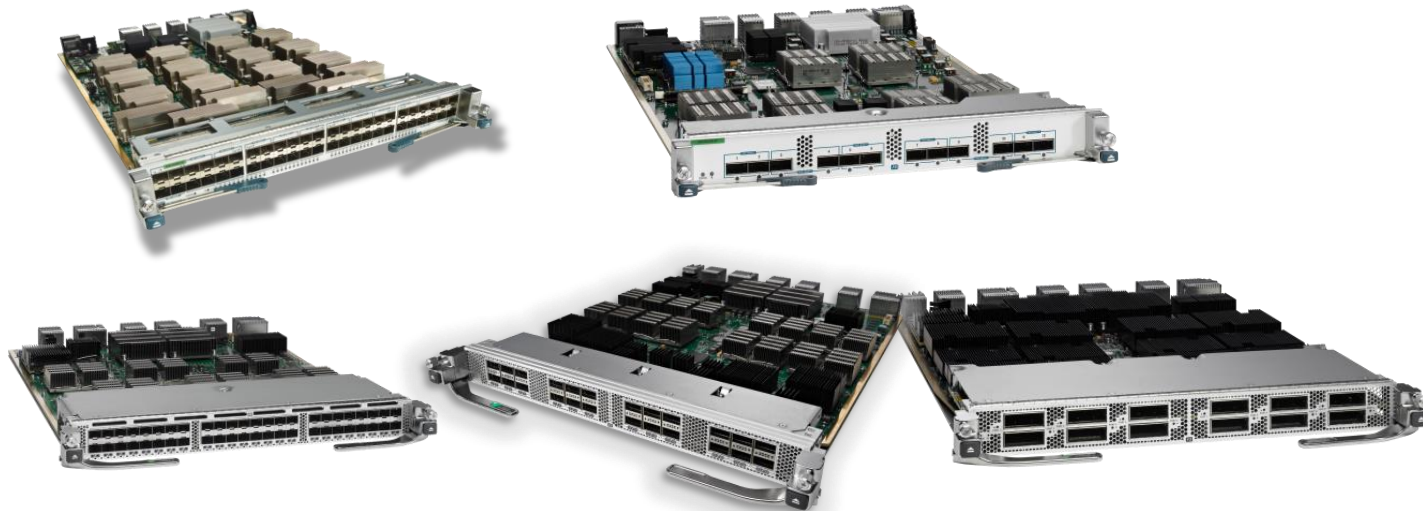
MB = Multicast Buffer  
FI = Fabric Interface

# M1+M2 Drop Counter Cheat Sheet:



CLI	Module	Description
<code>show interface counters errors</code>	Both	Basic in/outdiscards, CRC, FCS, etc errors
<code>show policy-map interface</code>	Both	Per-queue drops help identify which CoS traffic is affected
<code>show system internal qos queuing stats interface</code>	Both	Per-queue WRED and Taildrop counters
<code>slot &lt;x&gt; show hardware internal statistics device qengine congestion</code>	M1	Ingress Unicast VoQ drops due to no credits available b/c of egress congestion
<code>slot &lt;x&gt; show hardware skytrain queuing drops</code>	M2	Ingress Unicast VoQ drops due to no credits available b/c of egress congestion
<code>slot &lt;x&gt; show hardware internal statistics device qengine congestion   grep MB</code>	M1	Egress Multicast drops due to upper asic congestion prior to port-asic.
<code>clear statistics module-all device all</code>	All	Clear all hardware counters registers for monitoring purposes
<code>show tech module</code> <code>show tech module all</code>	All	Collect all hardware statistics for a particular module or all modules in the switch

# F2/F2E/F3 Packet Counters



**F2/F2E 10G**

**F3 10G / 40G / 100G**



# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Per-Queue Counters

- Per-Interface and Per-Queue counters for Ucast & Mcast...
- Ingress discards for credited traffic, output drops for uncredited (**Different than M-Series**)

```
N7K# show policy-map interface ethernet x/y | eg -i class|drop
```

```
Class-map (queuing): 2q4t-8e-in-q1 (match-any)
  queue dropped pkts : 0
Class-map (queuing): 2q4t-8e-in-q-default (match-any)
  queue dropped pkts : 0
Class-map (queuing): 1p3q1t-8e-out-pq1 (match-any)
  queue dropped pkts : 0
```

Per-queue drops help identify which CoS traffic is affected

```
N7K# show system internal qos queuing stats interface x/y
```

## **Receive queues**

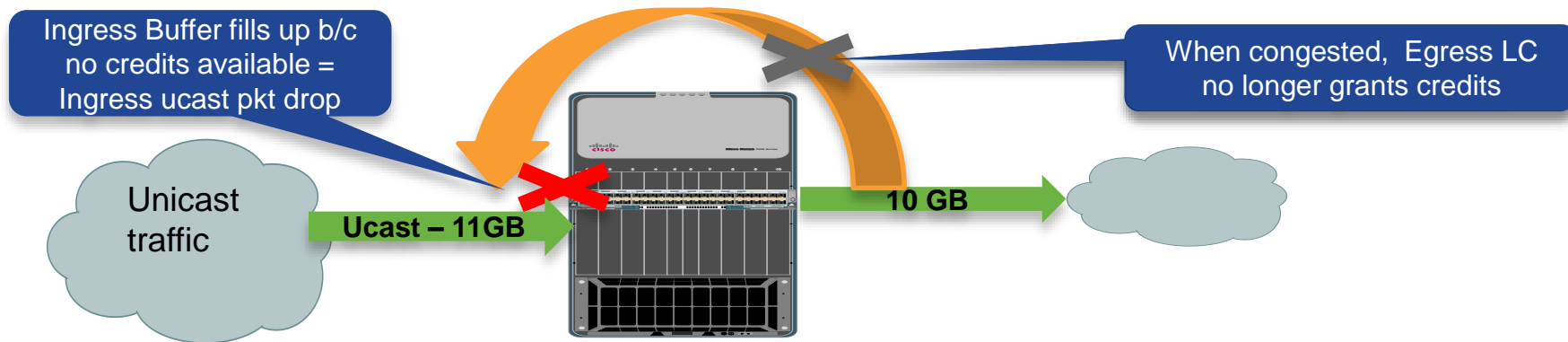
```
-----
Queue 2q4t-in-q-default
  Total packets          0
  WRED drops             0
  Taildrop drops         0
```

## **Transmit queues**

```
-----
Queue 1p3q4t-out-q-default
  Total packets          0
  WRED drops             0
  Taildrop drops        0
```

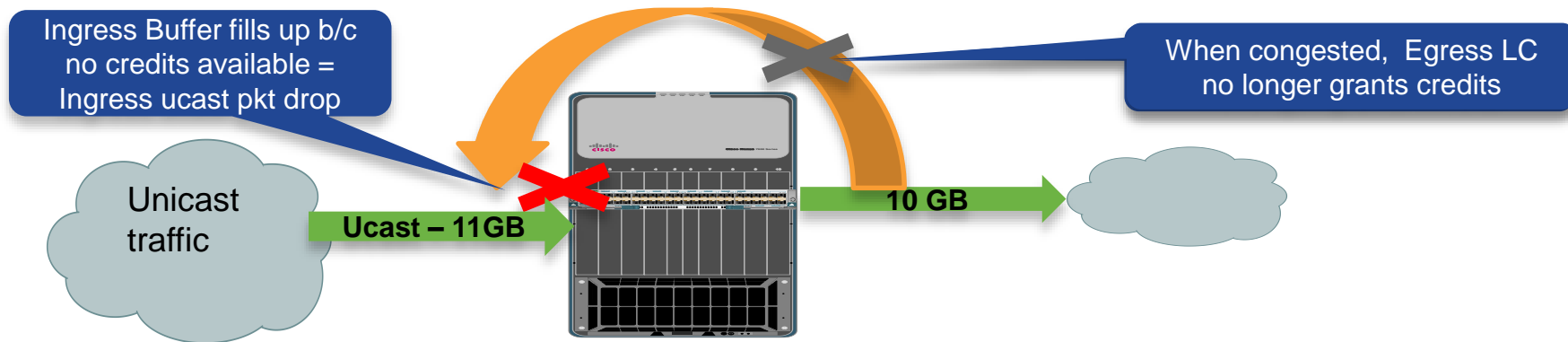
Shows more detailed queue counter/drop info

# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Ucast Congestion



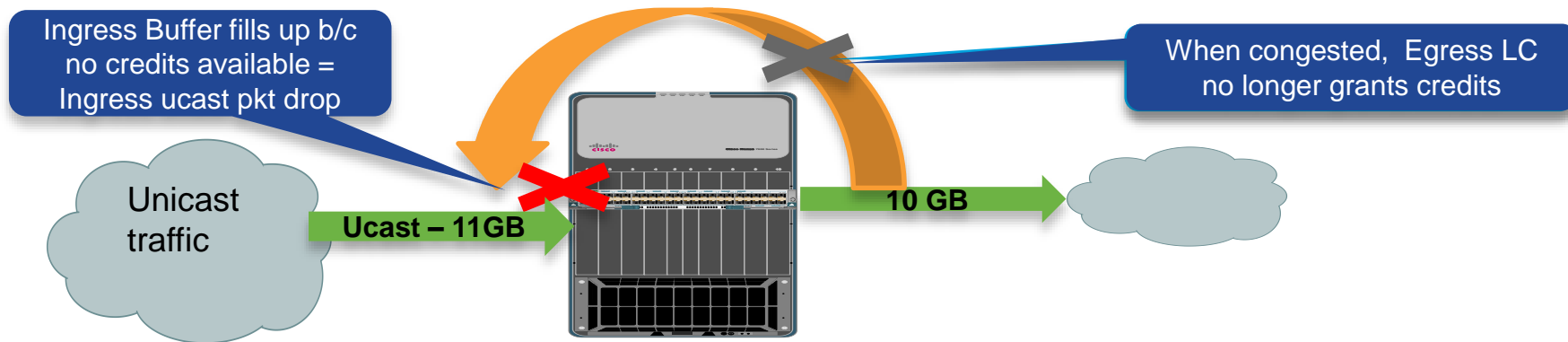
- **Unicast (credited)** traffic egress port congestion = **no credits given**
- **Ingress buffers** will fill up when **no credits** are given causing packet drops.

# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Ucast Congestion



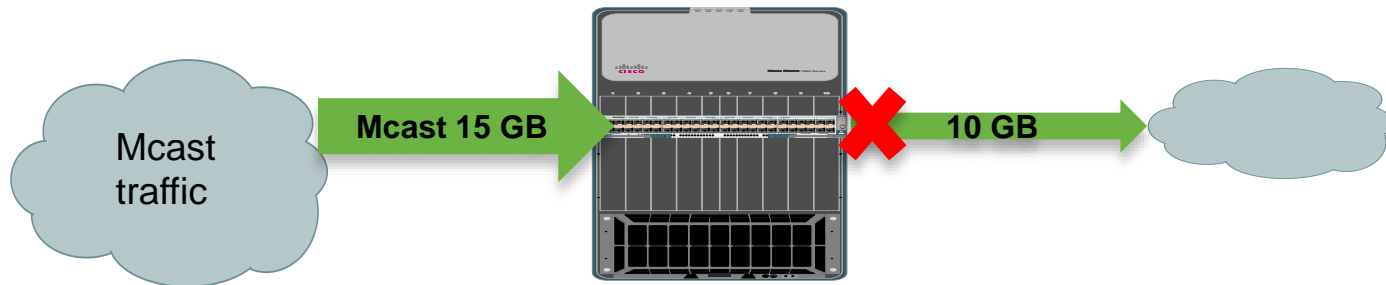
```
N7K# show int e 10/30 | eg -i discard
0 input with dribble 974183 input discard
```

# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Ucast Congestion



```
N7K# show policy-map interface ethernet 10/30 input | eg -i que|drop
Service-policy (queuing) input:  default-4q-8e-in-policy
Class-map (queuing):  2q4t-8e-in-q1 (match-any)
    queue dropped pkts : 0
Class-map (queuing):  2q4t-8e-in-q-default (match-any)
    queue dropped pkts : 974183
```

# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Multicast Congestion

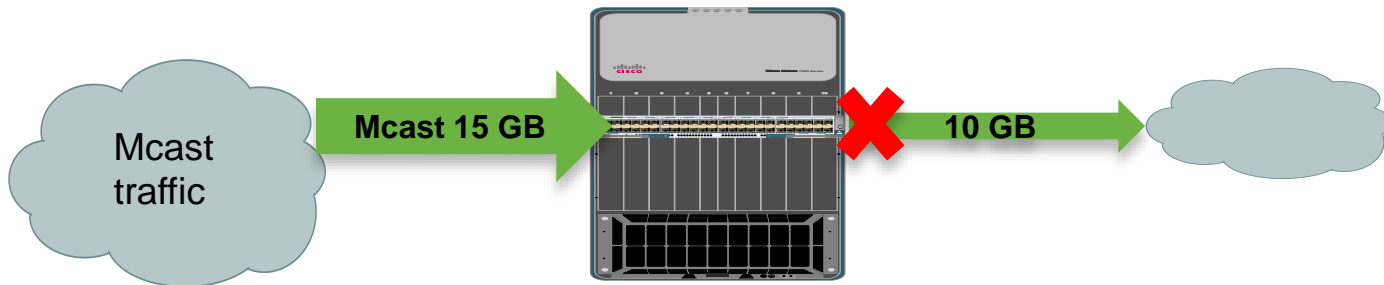


- Oversubscription by **multicast / uncredited** packets, dropped on **egress** buffer
- Per-interface / Per-queue counters available

```
N7K# show interface e10/1 counters errors
```

Port	Align-Err	FCS-Err	Xmit-Err	Rcv-Err	UnderSize	OutDiscards
Eth8/48	0	0	0	0	0	29853742

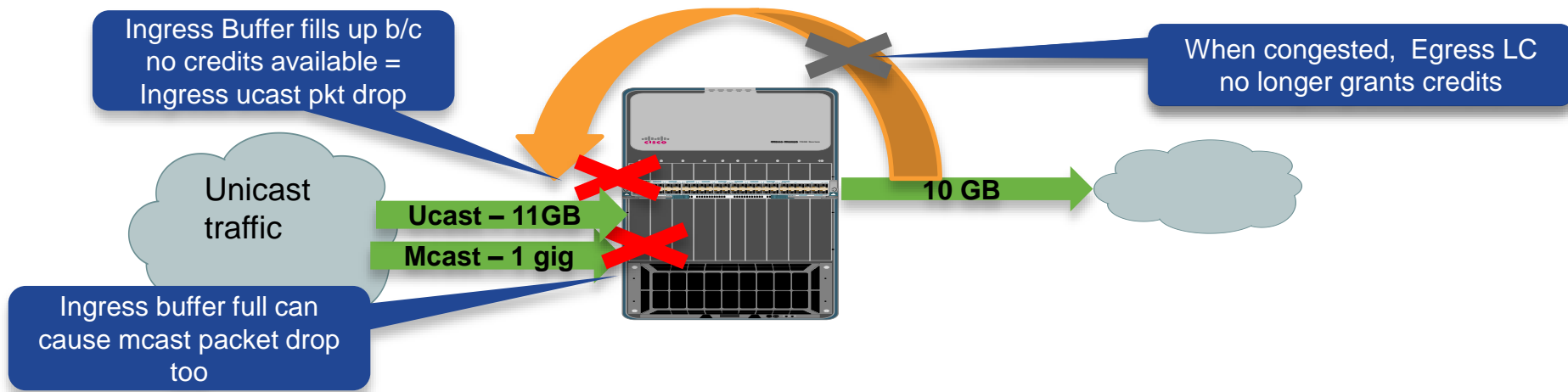
# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Multicast Congestion



- Oversubscription by **multicast / uncredited** packets, dropped on **egress** buffer
- Per-interface / Per-queue counters available

```
N7K# show policy-map interface e10/1 output | eg -i drop|queu
Service-policy (queuing) output: default-4q-8e-out-policy
<snip>
Class-map (queuing): 1p3q1t-8e-out-q-default (match-any)
queue dropped pkts : 29851887
```

# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Ucast Congestion affects Mcast



- **Unicast** (credited) traffic egress port congestion = no credits given
- **Ingress buffers** will fill up when **no credits** are given causing packet drops.
- Mcast Packets are also now dropped on ingress instead of egress
- May cause red herring if you don't see egress drops

# Data-Plane Troubleshooting Intermittent Packet Loss: F-Series Ucast Congestion affects Mcast

- No drops at egress direction may be misleading. Don't stop there...

```
N7K# show policy-map interface e10/1 output | eg -i drop
      queue dropped pkts : 0
      queue dropped pkts : 0
```

- We also need to look at **ingress counters**, considering ucast congestion scenario

```
N7K# show int e 10/30 | eg -i discard
      0 input with dribble 101593 input discard
```

```
N7K# show policy-map interface ethernet 10/30 input | eg -i que|drop
Service-policy (queuing) input: default-4q-8e-in-policy
Class-map (queuing): 2q4t-8e-in-q1 (match-any)
      queue dropped pkts : 0
Class-map (queuing): 2q4t-8e-in-q-default (match-any)
      queue dropped pkts : 107543
```

- F2 May be difficult to identify congested egress port (**improved detection on F2E & F3**)



# Data-Plane Troubleshooting Intermittent Packet Loss: F2E/F3 Optimizations

Where F2 and F2e/F3 Differ...

# Data-Plane Troubleshooting Intermittent Packet Loss: F2 Egress Port Identification

- F2 HW limitation cannot define VOQ drops
- Need to see VOQ congestion as it is happening
  - VOQ on ingress LC experiencing input discards

- FE Instance is zero based
- 4 port per FE
- Sh hard internal dev-port map

```
N7K# slot 10 show hardware internal gengine inst 7 voq-status non-empty
```

VQI:CCOS	BYTE_CNT	PKT_CNT	TAIL	HEAD	THR
95:3	6154	3077	6804	14168	1

VOQ of Egress Port experiencing Congestion

```
N7K# slot 10 show hardware internal gengine vqi-map
```

VQI	SUP	SLOT	LDI	EQI	FPOE	NUM	XBAR	IN	ASIC	ASIC	SV	FEA_
NUM	VQI	NUM	NUM	NUM	BASE	DLS	MASK	ORD	TYPE	IDX	ID	TURE
95	no	9	1	0	8	1	0x155	0	CLP	0	0	0x81

Slot = LC # (Zero Based)  
LDI = Port (need to map out)

# Data-Plane Troubleshooting Intermittent Packet Loss: F2 Egress Port Identification

- LDI to Physical port mapping **NOT 1 to 1**

	LDI	Port	LDI	Port	LDI	Port	LDI	Port
	FE 0-2		FE 3-5		FE 6-8		FE 9-11	
0	0	2	12	14	24	26	36	38
	<b>1</b>	<b>1</b>	13	13	25	25	37	37
	2	3	14	15	26	27	38	39
	3	4	15	16	27	28	39	40
1	4	6	16	18	28	30	40	42
	5	5	17	17	29	29	41	41
	6	7	18	19	30	31	42	43
	7	8	19	20	31	32	43	44
2	8	10	20	22	32	34	44	46
	9	9	21	21	33	33	45	45
	10	11	22	23	34	35	46	47
	11	12	23	26	35	36	47	48

LDI 1 maps to front panel port 1

Slot NUM is zero based

```
N7K# slot 10 sh hardware internal qengine vqi-map
VQI    SUP    SLOT    LDI
NUM    VQI    NUM    NUM
----    --    ----    --
122    no     9     1
```

**Physical Port = Eth 10/1**

# Data-Plane Troubleshooting Intermittent Packet Loss: F2E Egress Port Identification

- F2 HW limitation cannot define VOQ drops
- F2E/F3 Provides ability to look deeper into congestion counters
  - Help to **identify egress buffer congestion backing up ingress port.**
  - Only available in Admin VDC

```
Nexus-7k-Default-VDC# show hardware queuing drops egress
```

## VQ Drops

Output Interface	VQ Drops	VQ Congestion	Src Mod	Src Inst	Input Interface	
Eth1/1	0000000000393210	0000000000000000	1	7	Eth1/29-32	-----

VOQ drops are due to unicast /credited traffic

## Egress Buffer Drops

Output Interface	EB Drops
Eth3/21-24	0000000000001200

EB Drops are due to mcast / uncredited traffic

# F2 / F2e / F3 Drop Counter Cheat Sheet:



CLI	Module	Description
<code>show interface counters errors</code>	All	Basic in/out discards, CRC, FCS, etc errors
<code>show policy-map interface</code>	All	Per-queue drops help identify which CoS traffic is affected
<code>show system internal qos queuing stats interface</code>	All	Per-queue WRED and Taildrop counters
<code>show hardware queuing drops egress module</code>	F2e/F3	Ingress Unicast VoQ drops due to no credits available b/c of egress congestion
<code>clear statistics module-all device all</code>	All	Clear all hardware counters registers for monitoring purposes
<code>show tech module</code> <code>show tech module all</code>	All	Collect all hardware statistics for a particular module or all modules in the switch

# *Summary*

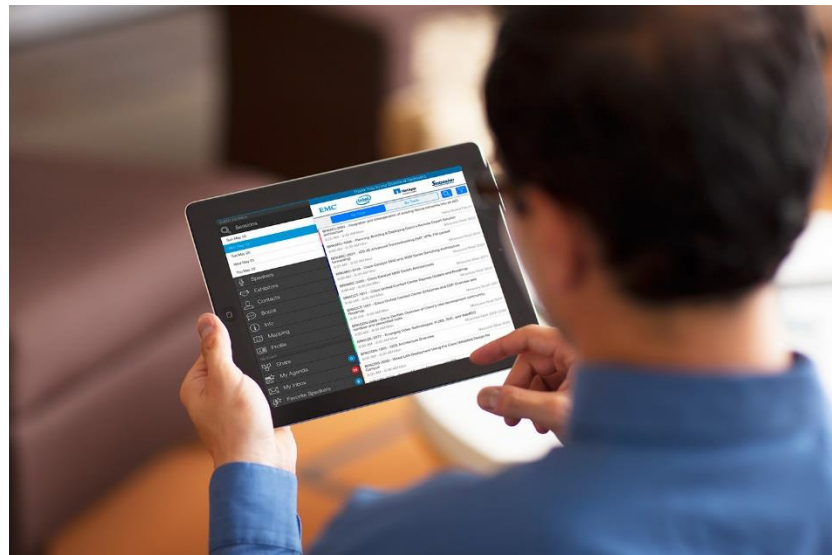
# Participate in the “My Favorite Speaker” Contest

## Promote Your Favorite Speaker and You Could Be a Winner

- Promote your favorite speaker through Twitter and you could win \$200 of Cisco Press products (@CiscoPress)
- Send a tweet and include
  - Your favorite speaker’s Twitter handle
  - Two hashtags: #CLUS #MyFavoriteSpeaker
- You can submit an entry for more than one of your “favorite” speakers
- Don’t forget to follow @CiscoLive and @CiscoPress
- View the official rules at <http://bit.ly/CLUSwin>

# Complete Your Online Session Evaluation

- Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a \$750 Amazon gift card.
- Complete your session surveys through the Cisco Live mobile app or your computer on Cisco Live Connect.



Don't forget: Cisco Live sessions will be available for viewing on-demand after the event at [CiscoLive.com/Online](https://cislive.com/online)



# Continue Your Education

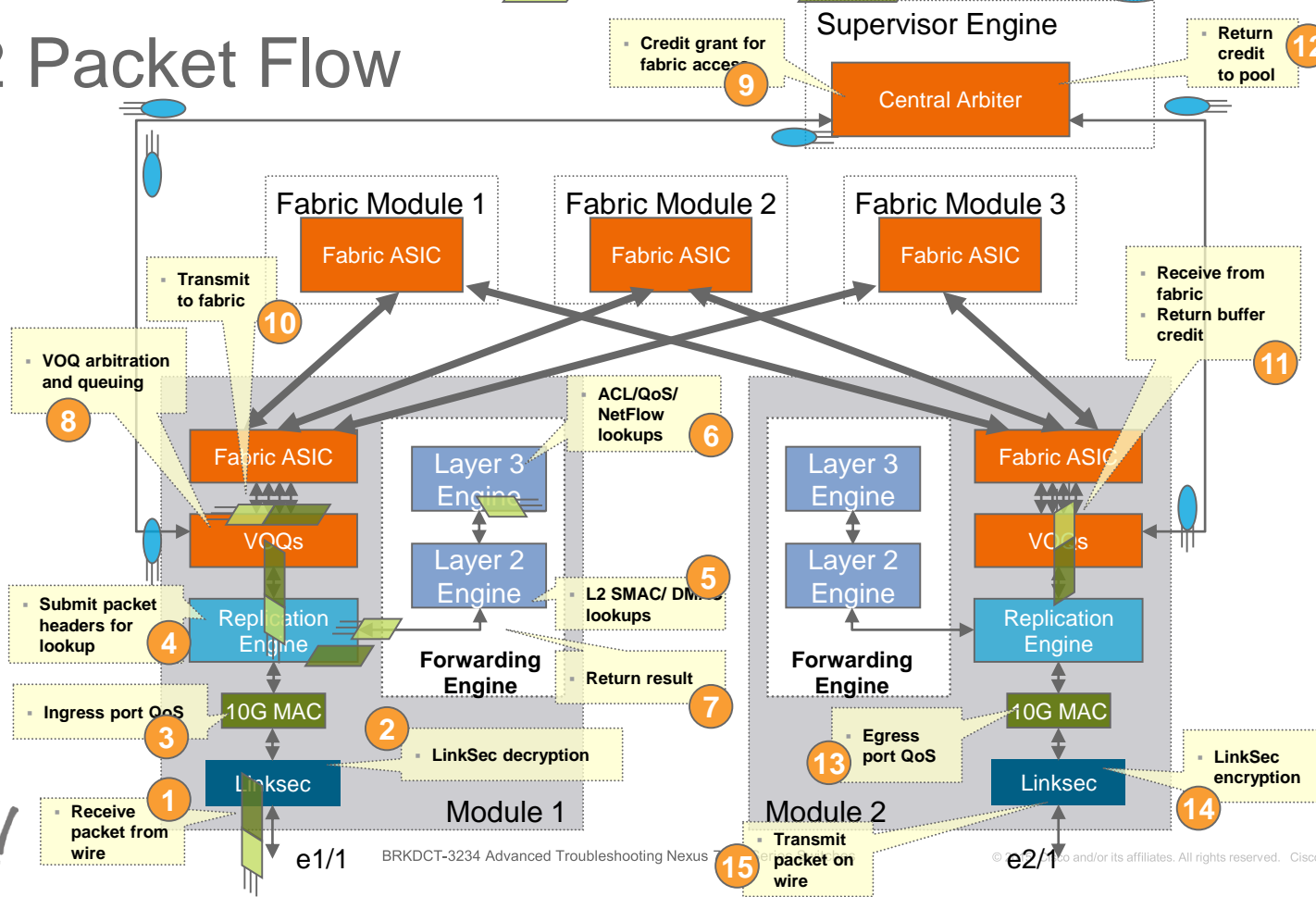
- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Table Topics
- Meet the Engineer 1:1 meetings
- Related sessions

*Thank you*

# *Appendix I: Packet walks*

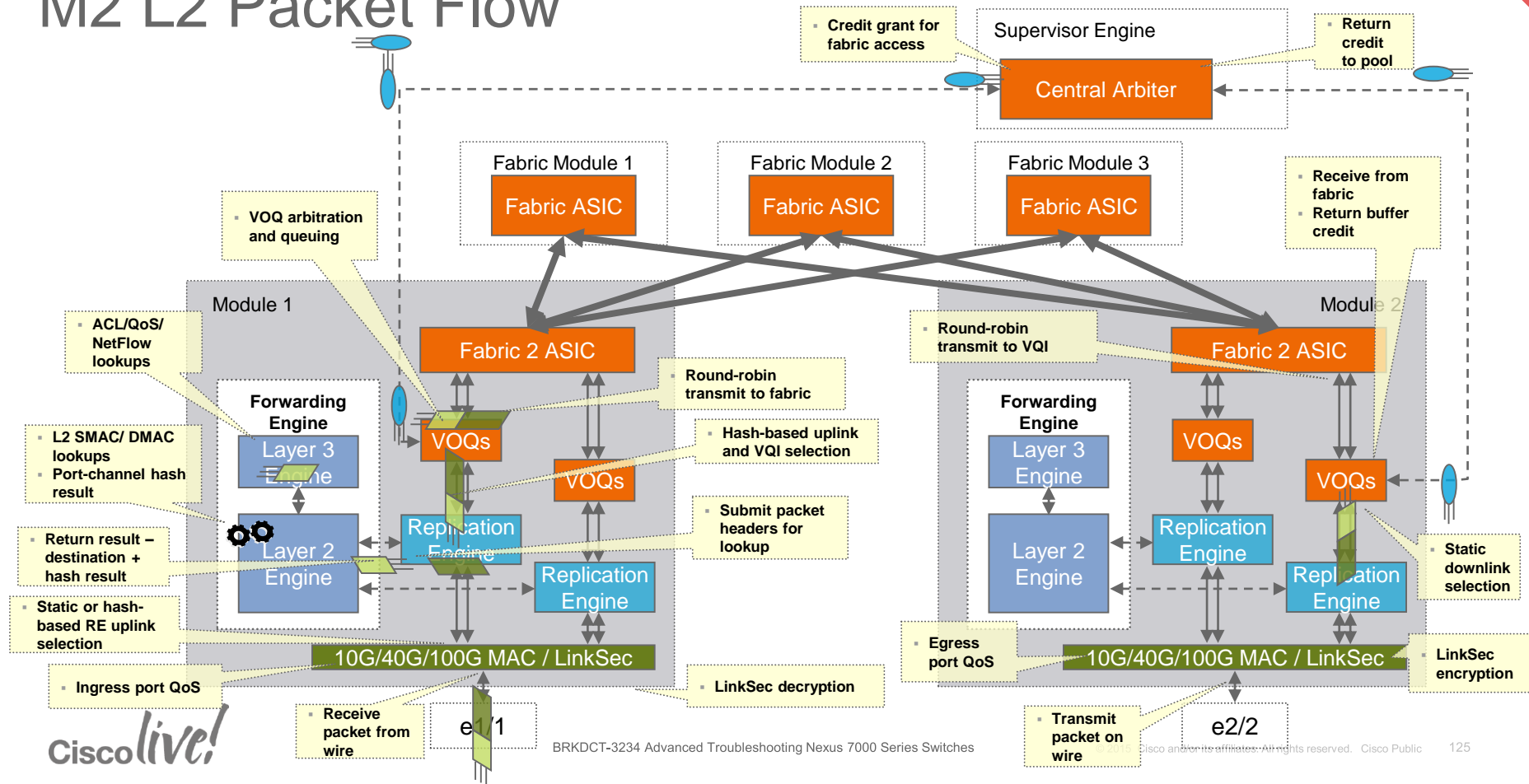
# M1 L2 Packet Flow

**HDR** = Packet Headers   **DATA** = Packet Data   **CTRL** = Internal Signalling



**HDR** = Packet Headers   **DATA** = Packet Data   **CTRL** = Internal Signalling

# M2 L2 Packet Flow

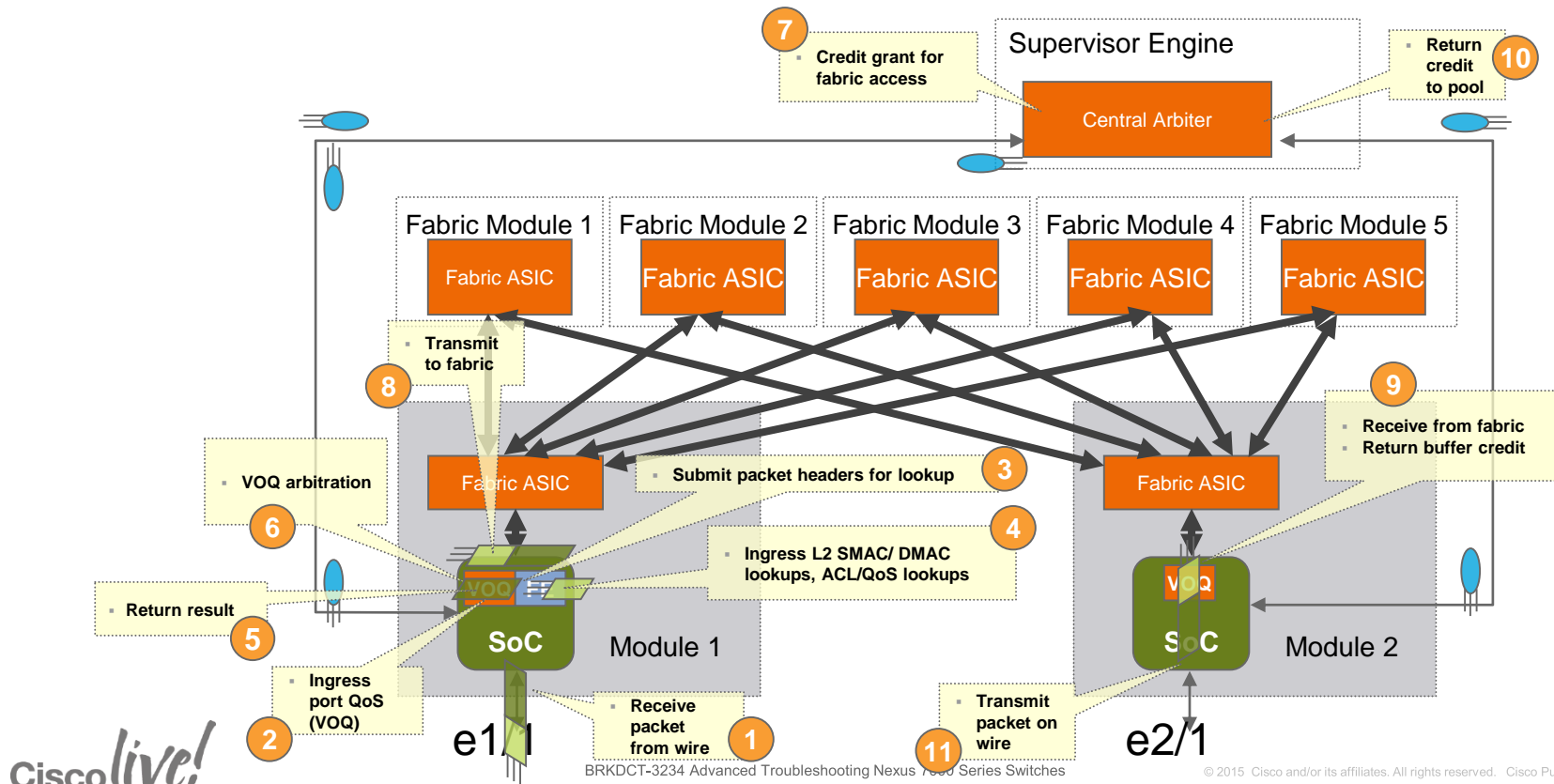


HDR = Packet Headers

DATA = Packet Data

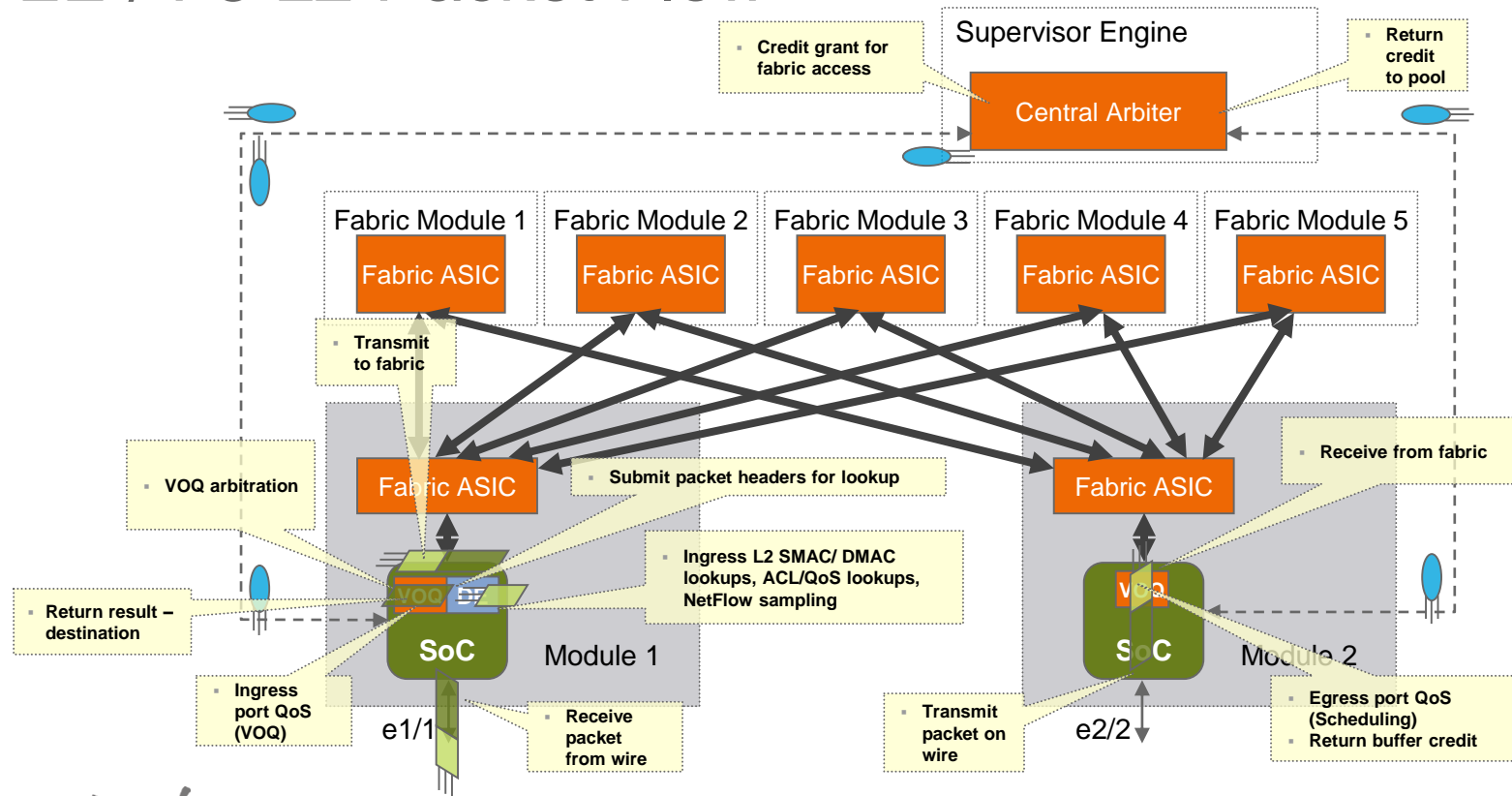
CTRL = Internal Signalling

# F2 L2 Packet Flow



**HDR** = Packet Headers    **DATA** = Packet Data    **CTRL** = Internal Signalling

# F2E / F3 L2 Packet Flow



# *Appendix II: ELAM Examples*



# ELAM



## Detailed Nexus 7000 Overview

M1/M2	F1	F2	F3
Eureka	Orion	Clipper	Flanker

- Each of the above ASICs has the **L2LKP/L2LU** role. The idea is to find the instance number of the L2LKP/L2LU ASIC for the ingress port so the ELAM can be performed on the correct ASIC. Attach to the module and issue **show hardware internal dev-port-map** to perform this task.
- To map source LTL index and destination LTL index to a port(s), use **show system internal pixm info ltl <index>**

# ELAM

## Nexus 7000 – M1/M2 Example



```
N7k# attach module 4
Attaching to module 4 ...
To exit type 'exit', to abort type '$.'
module-4# show hardware internal dev-port-map
```

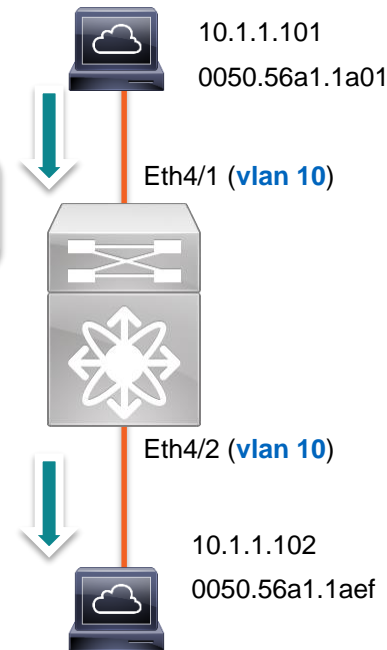
```
CARD_TYPE:          48 port 1G
>Front Panel ports:48
```

```
-----
Device name          Dev role          Abbr num_ins
-----
> Eureka              DEV_LAYER_2_LOOKUP    L2LKP  1
+-----+
+-----+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++-----+
+-----+
FP port|PHYS  |SECUR  |MAC_0  |RWR_0  |L2LKP  |L3LKP  |QUEUE  |SWICHF
-----+-----+-----+-----+-----+-----+-----+-----+
  1     0    0    0    0    0    0    0    0
  2     0    0    0    0    0    0    0    0
-----+-----+-----+-----+-----+-----+-----+-----+

```

There is only one Eureka ASIC for the entire card, therefore, specifying the instance is not necessary

L2LKP for M1 is Eureka ASIC. Ingress port (Eth4/1) is on instance 0



# ELAM

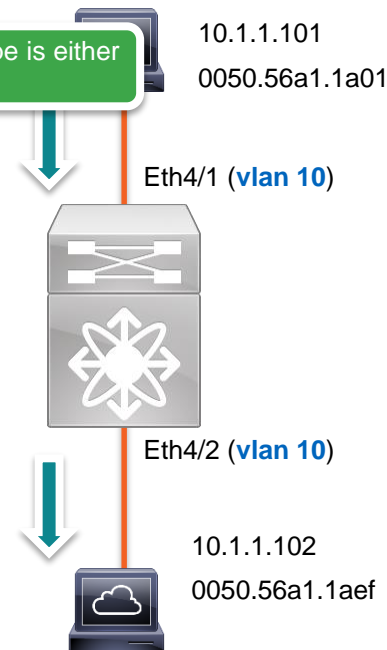
## Nexus 7000 – M1/M2 Example



```
module-4# elam asic eureka
module-4(eureka-elam)# trigger dbus dbi ingress ?
  ipv4    IPv4 Frame Format
  ipv6    IPv6 Frame Format
  other    Compact Format - Others
module-4(eureka-elam)# trigger dbus dbi ingress ipv4 if ?
<CR>
destination-ipv4-address  Destination IP Address
destination-mac-address   Destination MAC Address
length                   The dbus Frame Length
source-index              Source Index
source-ipv4-address       Source IP Address
source-mac-address        Source MAC Address
tcp                      TCP
tcp-flag                  TCP Flag
type                     Frame Type on the dbus
udp                      UDP
vlan-id                   Vlan ID Number
etc...
```

Similar to 6500, M1 packet type is either  
ipv4, ipv6, or other

Large list of options in  
which to build the trigger



# ELAM

## Nexus 7000 – M1/M2 Example



```
module-4# elam asic eureka
module-4(eureka-elam)# trigger dbus dbi ingress ipv4 if source-ipv4-address
10.1.1.101 destination-ipv4-address 10.1.1.102 rbi-corelate
module-4(eureka-elam)# trigger rbus rbi pb1 ip if cap2 1
module-4(eureka-elam)# start
module-4(eureka-elam)# status
Instance: 1
```

```
EU-DBUS: Triggered
trigger dbus dbi ingress ipv4 if source-ipv4-address 10.1.1.101 destina
address 10.1.1.102 rbi-corelate
```

```
EU-RBUS: Triggered
trigger rbus rbi pb1 ip if cap2 1
```

This example triggers based off source and destination IP address. Note that for M1/M2 ELAM, **rbi-corelate** is required for RBUS to trigger

Must manually configure rbus trigger. Choose pb1 or pb2 (based on ingress port and board architecture). The **rbi-corelate** in the dbus trigger sets the **cap2** bit which is used to trigger the rbus.



10.1.1.101

1a01

Eth4/2 (vlan 10)



10.1.1.102

0050.56a1.1aef

# ELAM

## Nexus 7000 – M1/M2 Example



(some output omitted)  
module-4(eureka-elam) # **show dbus**

**seq = 0x17**

vlan = 10

**source\_index = 0x00a21**

l3\_protocol = 0x0 (0:IPv4, 6:IPv6)

l3\_protocol\_type = 0x06, (1:ICMP, 2:IGMP, 4:IP, 6:TCP, 7:UDP)

dmac = 00.50.56.a1.1a.ef

smac = 00.50.56.a1.1a.01

ip\_ttl = 0x40

ip\_source = 010.001.001.101

ip\_destination = 010.001.001.102

tcp source port = 0x0050

tcp dest port = 0x2309

tcp sequence no = 0x00af5322

tcp acknowledgement no = 0x0e8aff20

module-4(eureka-elam) # **show rbus**

**seq = 0x17**

**flood = 0x1**

**dest\_index = 0x00048**

vlan = 10

ttl = 0x40

Ensure that sequence number in  
dbus and rbus match



10.1.1.101

0050.56a1.1a01

N7k# show system internal pixm info ltl **0xa21**

Type	LTL
-----	
PHY_PORT	Eth4/1

The **flood bit** is set in the rbus result which means this frame is flooded on vlan 10. The flood-bit is the most significant bit in the destination index. Therefore destination index changes from 0x0048 to 0x8048:

N7k# show system internal pixm info ltl **0x8048**

IFIDX	LTL
-----	
Eth4/1	0x0a21
Eth4/2	0x0a20

# ELAM

## Nexus 7000 – F2/F3 Example

Note F2 and F3 ELAM use identical syntax. The primary difference is the internal ASIC name:

F2 – Clipper

F3 - Flanker



```
N7k# attach module 6
Attaching to module 6 ...
To exit type 'exit', to abort type '$.'
module-6# show hardware internal dev-port-map
```

```
-----
CARD_TYPE:          48 port 10G
>Front Panel ports:48
-----
```

```
Device name          Dev role          Abbr num_inst:
-----
```

```
> Clipper FWD          DEV_LAYER_2_LOOKUP    L2LKP  12
-----
```

```
+-----+
+-----++FRONT PANEL PORT TO ASIC INSTANCE MAP+++-----+
+-----+
-----
```

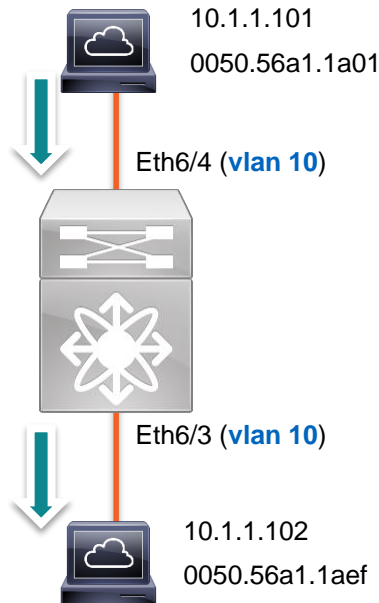
```
FP port |  PHYS | MAC_0 | L2LKP | L3LKP | QUEUE | SWICHF
```

```
...
```

3	0	0	0	0	0	0
4	0	0	0	0	0	0

12 different FE on F2 module

Ingress port 4 is on FE 0



# ELAM

## Nexus 7000 – F2/F3 Example



```
module-6# elam ASIC clipper instance 0
module-6(clipper-elam)# layer2
module-6(clipper-l2-elam)# trigger dbus ?
```

```
arp      ARP Frame Format
fc       Fc hdr Frame Format
ipv4     IPV4 Frame Format
ipv6     IPV6 Frame Format
other    L2 hdr Frame Format
pup      PUP Frame Format
rarp     Rarp hdr Frame Format
valid    On valid packet
```

```
module-6(clipper-l2-elam)# trigger dbus ipv4 ingress if ?
```

<CR>

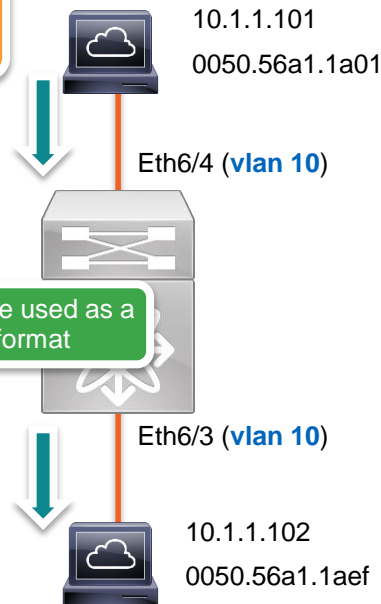
```
destination-ipv4-address
destination-mac-address
source-index
source-ipv4-address
source-mac-address
vlan
etc...
```

```
destination ipv4 address
Inner destination mac address
Source index
source ipv4 address
Inner source mac address
Vlan
```

Clipper ASIC is responsible for layer2 functions and layer3. Similar to M1, we want to see the results from the L2LKP

More packet types available in Clipper parser. Ensure the correct format is chosen for the flow of interest

Large list of fields that can be used as a trigger for each frame format



# ELAM

## Nexus 7000 – F2/F3 Example



```
module-6(clipper-l2-elam)# trigger dbus ipv4 ingress if source-ipv4-address  
10.1.1.101 destination-ipv4-address 10.1.1.102
```

```
module-6(clipper-l2-elam)# trigger rbus ingress if trig
```

```
module-6(clipper-l2-elam)# start
```

```
module-6(clipper-l2-elam)# status
```

```
ELAM instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if source-  
address 10.1.1.101 destination-ipv4-address 10.1.1.102
```

```
L2 DBUS Triggered
```

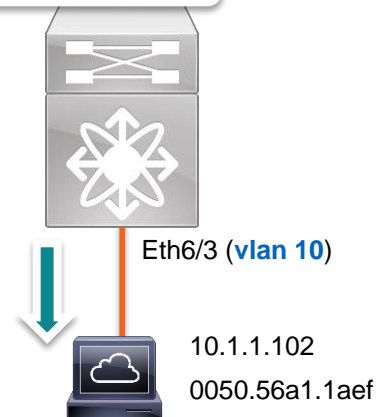
```
ELAM instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
```

```
L2 RBUS Triggered
```

Trigger based on source and destination  
IP address. No rbi-correlate required for  
Clipper

Simple rbus trigger: if trig

(vlan 10)





# ELAM

## Nexus 7000 – F2/F3 Example



```
module-6(cli-12-elam)# show dbus
```

```
-----  
L2 DBUS CONTENT - IPV4 PACKET  
-----
```

```
...  
vlan : 0xa  
source-index : 0x3  
sequence-number : 0x3f  
...  
source-ipv4-address: 10.1.1.101  
destination-ipv4-address: 10.1.1.102  
destination-mac-address 0050.56a1.1aef  
source-mac-address: 0050.56a1.1a01
```

```
module-6(cli-12-elam)# show rbus
```

```
-----  
L2 RBUS INGRESS CONTENT  
-----
```

```
12-rbus-trigger : 0x1  
di-ltl-index : 0x2  
source-index : 0x3  
sequence-number : 0x3f  
13-multicast-di : 0x0  
vlan-id : 0xa
```

A lot of information about the ingress packet is displayed in the dbus headers. Here we can validate the traffic was received on source index (0x3) which is port Eth6/4.

```
N7k# show system internal pixm info ltl 0x3  
Type LTL  
-----  
PHY_PORT Eth6/4
```

The destination index (0x2) maps to port Eth6/3

```
N7k# show system internal pixm info ltl 0x2  
Type LTL  
-----  
PHY_PORT Eth6/3
```



10.1.1.102  
0050.56a1.1aef

# *Appendix III: Core Files (Process Restarts)*

# Core Files

- Core Files are created when a process restarts
- Knowing **what, where & when** is key

```
2015 Jan 17 22:13:26 Nexus %SYSMGR-2-SERVICE_CRASHED: Service "__inst_001__ospf" (PID 6219) hasn't caught signal 6 (core will be saved)
```

```
2015 Jan 27 00:30:18 Nexus VDC-1 %$ %SYSMGR-SLOT8-2-SERVICE_CRASHED: Service "mtm" (PID 1600) hasn't caught signal 6 (core will be saved).
```

- Processes run on Supervisors and Linecards
- Helpful to then gather `show-tech + show tech ospf`

## Core Files cont...

- Copy the core to remote server. Keeping in mind what slot the core is on.

```
N7K# sh cores vdc-all
```

VDC	Module	Instance	Process-name	PID	Date (Year-Month-Day Time)
2	5	1	ospf-1	6219	2015-01-27 22:13:26
3	5	1	netstack	6919	2015-01-27 23:32:46

```
N7K# dir logflash://module-5/vdc_3/core
```

10052989	Jan 27 23:33:18 2015	1359329598_0x509_netstack_log.6919.tar.gz
----------	----------------------	---

```
N7K# copy core: ?
```

core: Enter URL "core://<module-number>/<process-id>[/instance-num]"

If "show cores" is empty, always check persistent storage logflash



*TOMORROW starts here.*