
cisco
You make possible

Monitoring and Troubleshooting Nexus 9000 (standalone) Switches

Yogesh Ramdoss
Principal Engineer, Customer Experience
@YogiCisco

BRKDCN-3020

cisco Live!
Barcelona | January 27-31, 2020

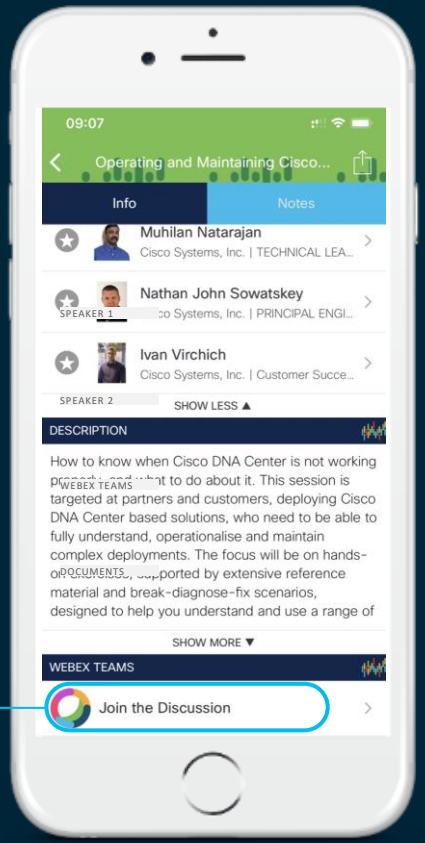
Cisco Webex Teams

Questions?

Use Cisco Webex Teams to chat with the speaker after the session

How

1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



Agenda

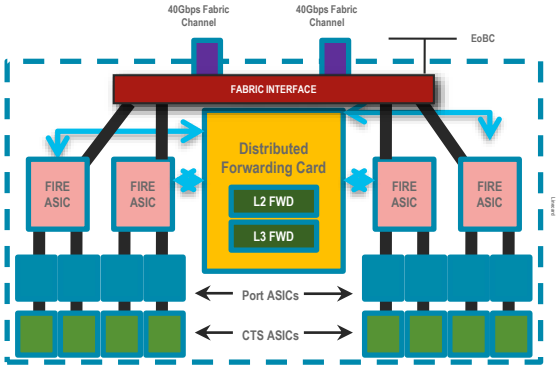
- Introduction
- Monitor and Health-Check
- Troubleshooting Tools
- Troubleshooting Traffic Forwarding
- Best Practices and Recommendations
- Summary and Take-Aways

Introduction

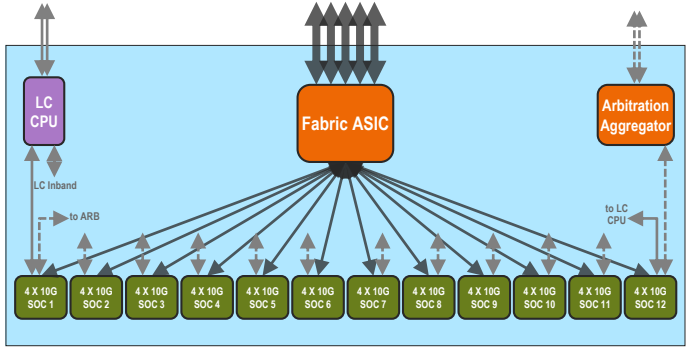
Switching Architecture Changes

Consolidation of Functions

FWD – Forwarding
FIRE – Fabric Interface and Replication Engine ASIC
CTS – Cisco TrustSec
SOC – Switch on Chip



32 x 10G Ports



48 x 10G Ports

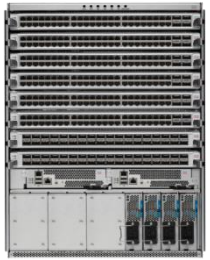
64 x 100G Ports

Design Shifts Resulting from Increasing Gate Density and Bandwidth

Catalyst 6807-XL

Nexus 7700

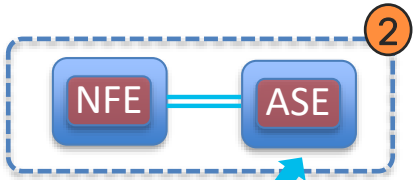
Nexus 9508



Generations of Nexus 9000

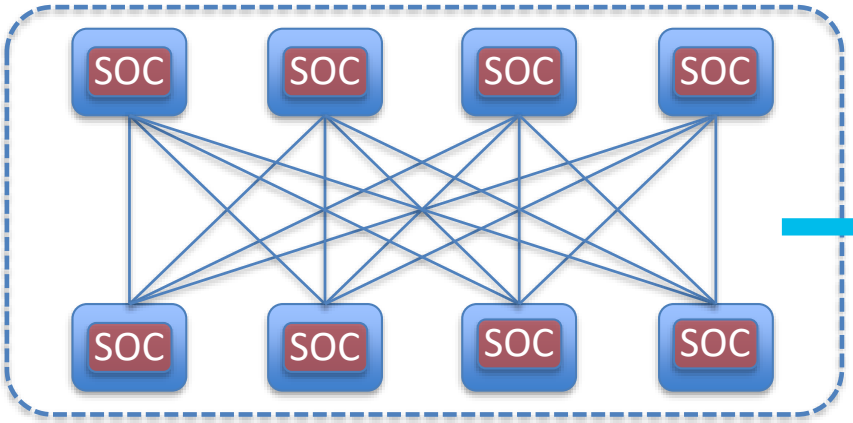
NFE – Network Forwarding Engine
ASE – Application Spine Engine
SOC – Switch On Chip

Merchant Silicon



Leverages merchant Silicon + Cisco ASIC to enhance services

Switch “is” the ASIC



Non-Blocking Leaf and Spine based CLOS Network inside the Switch

Nexus 9000 Product Family

Focus For This Session

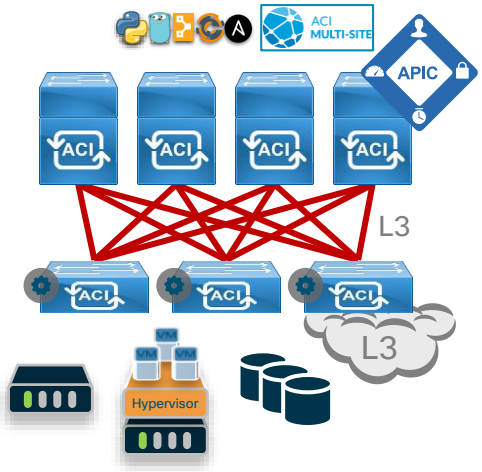
ASICs	Platforms
StrataXGS Trident*	94XX, 9636
StrataXGS Tomahawk*	9432C, C950X-FM-S
StrataXGS Trident* + Northstar	9396, 93128, 95XX
StrataXGS Trident* + Donner	9372, 9332, 93120
StrataDNX Jericho*	X9600-R/X-9600-RX
Tahoe-Sugarbowl	93XX-EX, 97XX-E/EX
Tahoe-Lacrosse	92XX, C950X-FM-E
Tahoe-Davos	92160YC
Rocky-Homewood	F/FX/FXP
Rocky-Bigsky	9364C,C95XX-FM-E2
Rocky-Heavenly	FX2
Rocky-Sundown	FX3

This session is going to discuss …

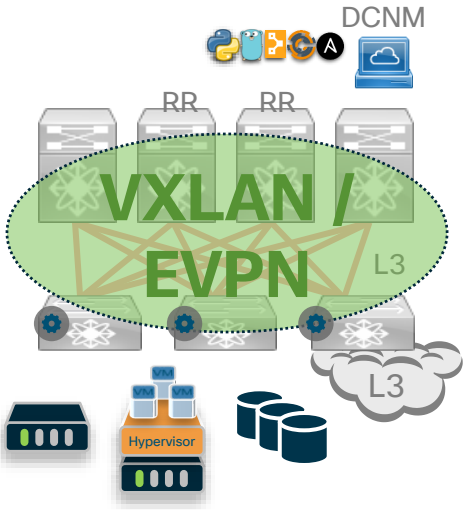
Cisco Cloud-Scale ASICs

* Merchant Silicon from Broadcom

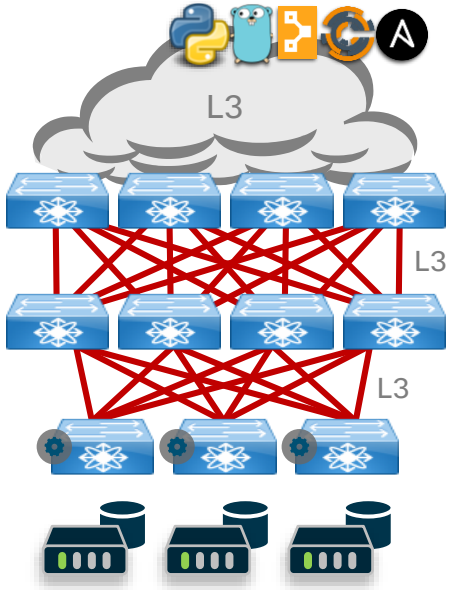
Building Data Center Fabrics with Nexus 9000



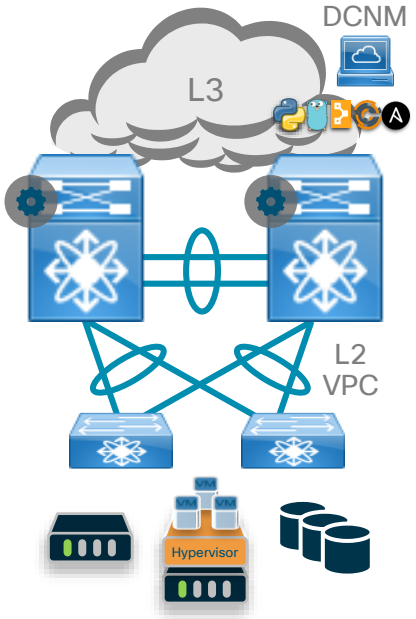
Application Centric Infrastructure (ACI) – Turnkey Fabric



Standalone – Programmable Fabric with VXLAN+EVPN



Standalone – Programmable IP Network



Standalone – Traditional Data Center Network

DCNM – Data Center Network Management

Just to let you know…

Reference ➔ 

- With wide range of Nexus9000 platforms available in the marketplace, this session is going to focus on models that are with Cloud-Scale ASICs and are at the cutting-edge.
- We will not be discussing hardware architecture in detail, but will provide a quick refresher
- With good number of topics to cover, we are not going to discuss Multicast, QoS or Buffering.
- Please hold on to your questions till end of the section.
- At any point of time during the presentation and after, you can ask your question in *Webex Teams* room.

Just to let you know...

Reference ➔ 

- Focus on Nexus 9000 models with Cloud-scale ASICs
- No deep-dive hardware architecture discussion. Will provide a quick refresher.
- No discussion on Multicast, QoS or Buffering
- Please hold on to your questions till end of the section.
- At any point of time during the presentation and after, you can ask your question in *Webex Teams* room.


Nexus 9000
... platform of possibilities

Monitor and Health-Check

Agenda

- Introduction
- Monitor and Health-Check
- Troubleshooting Tools
- Troubleshooting Traffic Forwarding
- Best Practices and Recommendations
- Summary and Take-Aways

- Hardware Diagnostics
- On-board Failure Logging
- Device Resource Usage
- Control-Plane Policing
- Hardware Rate-Limiters

Hardware Diagnostics

Configuration and Commands

run at bootup and detect faulty hardware before it is brought online by NX-OS. E.g., EOBCPortLoopback

No License Required

Setting bootup diagnostic level

```
N93128# config t
N93128(config)# diagnostic bootup level [bypass | complete]
```

detect runtime hardware errors, memory errors, hardware degradation, software faults, and resource exhaustion and more.

Activating a runtime diagnostic test and setting interval

```
N93128(config)# diagnostic monitor interval module <mod#>
            test <test-id | name | all> hour <hour> min <min> second <sec>
N93128(config)# diagnostic monitor module <mod#> test <test-id | name | all>
```

run once or at user-designated intervals. Help to localize faults.

Setting ondemand diagnostic test, starting and stopping

```
N93128# diagnostic ondemand iteration <count>
N93128# diagnostic ondemand action-on-failure {continue failure-count <num-fails> | stop}
N93128# diagnostic start module <mod#> test [test-id | name | all | non-disruptive ] [port
port-number | all ]
N93128# diagnostic stop module <mod#> test [test-id | name | all]
```

Hardware Diagnostics

Configuration and Commands

Diagnostic tests status and testing intervals:

Run "**show diagnostic result** <options>" to find the test results.

```
N93128# show diagnostic content module <mod | all>
Diagnostics test suite attributes:
B/C/* - Bypass bootup level test / Complete bootup level test / NA
P/*   - Per port test / NA
M/S/* - Only applicable to active / standby unit / NA
D/N/* - Disruptive test / Non-disruptive test / NA
H/O/* - Always enabled monitoring test / Conditionally enabled test / NA
F/*   - Fixed monitoring interval test / NA
X/*   - Not a health monitoring test / NA
E/*   - Sup to line card test / NA
L/*   - Exclusively run this test / NA
T/*   - Not an ondemand test / NA
A/I/* - Monitoring is active / Monitoring is inactive / NA
Module 1: 1/10G-T Ethernet Module (Active)
ID     Name                                Attributes     Testing Interval(hh:mm:ss)

____   ___________________________________ ____________   __________________
 1)    USB--------------------------->     C**N**X**T*      -NA-
16)    RewriteEngineLoopback--------->     CP*N***E**A      00:01:00
```

On-Board Failure Logging (OBFL)

Why we need it and what it does?

No License Required

- OBFL logs failure data to persistent storage
- Persistent storage: Non-volatile flash memory on the modules. Accessible for future analysis.
- Enabled by default for all features
- As OBFL Flash supports limited numbers of Read-Write operations, choose key set of features for logging.

On-Board Failure Logging (OBFL)

Configuration and Status

```
N93128(config)# hw-module logging onboard ?
  <CR>
  counter-stats          Enable/Disable OBFL counter statistics
  cpuhog                 Enable/Disable OBFL cpu hog events
  environmental-history  Enable/Disable OBFL environmental history
  error-stats            Enable/Disable OBFL error statistics
  interrupt-stats        Enable/Disable OBFL interrupt statistics
  module                 Enable/Disable OBFL information for Module
  obfl-logs              Enable/Disable OBFL (boot-uptime/device-version/obfl-history)

N93128# show logging onboard status
----------------------------
OBFL Status
----------------------------
    Switch OBFL Log:                                  Enabled
    Module:  1 OBFL Log:                              Enabled
    card-boot-history                                 Enabled
    card-first-power-on                               Enabled
    <snip>
```

On-Board Failure Logging (OBFL)

CLI Options

Nearly 20 different options!

```
N93128# show logging onboard ?
  boot-uptime             Boot-uptime
  card-boot-history       Show card boot history
  card-first-power-on     Show card first power on information
  counter-stats           Show OBFL counter statistics
  credit-loss             Show OBFL Credit Loss logs
  device-version          Device-version
  endtime                 Show OBFL logs till end time mm/dd/yy-HH:MM:SS
  environmental-history   Environmental-history
  error-stats             Show OBFL error statistics
  exception-log           Exception-log
  flow-control            Show OBFL Flow Control log
  internal                Show Logging Onboard Internal
  interrupt-stats         Interrupt-stats
  kernel-trace            Show OBFL Kernel Trace
  module                  Show OBFL information for Module
  obfl-history            Obfl-history
  obfl-logs               Show OBFL Tech Support Log.
  stack-trace             Stack-trace
  starttime               Show OBFL logs from start time mm/dd/yy-HH:MM:SS
  status                  Status
```

On-Board Failure Logging (OBFL)

Example – OBFL Exception Log

```
N93128# show logging onboard EXCEPTION-LOG
----------------------------------------------------------------
    Module:  1
----------------------------------------------------------------
<snip>
exception information --- exception instance 1 ----
Device Id          : 49
Device Name        : Temperature-sensor
Device Errorcode   : 0xc3101203
Device ID          : 49 (0x31)
Device Instance    : 01 (0x01)
Dev Type (HW/SW)   : 02 (0x02)
ErrNum (devInfo)   : 03 (0x03)
System Errorcode   : 0x4038001e Module recovered from minor temperature alarm
Error Type         : Minor error
PhyPortLayer       :
Port(s) Affected   :
<snip>
Time               : Sun Oct 20 13:41:51 2019
```

Device Resource Usage

Checking Usage of Resources

No License Required

show hardware capacity <options>

Resource	What it gives?
Module	Usage of Bootflash, Logflash, and NVRAM
Interface	Total Tx/Rx drops (per module) and ports with highest drop count
Forwarding	L2 CAM table resource, ACL resources, IPv4/v6 Unicast Host and Route entries resources, IPv4/v6 Multicast entries resources, QoS resources (aggregate and distributed policers) – **per module and per forwarding engine instance**
Fabric	Fabric channel bandwidth, current ingress and egress traffic rate
Power	PSU redundancy mode, total capacity, power reserved (for Sup, fabric modules and fans), and power drawn
EOBC (Ethernet Out of Band Channel)	Total packets forwarded, transmit rate, dropped packets

Device Resource Usage

Usage – Hardware Forwarding Resource

Command outputs are tailored to highlight key features

```
N9504# show hardware capacity FORWARDING
<snip>
INSTANCE 0x0: ACL Hardware Resource Utilization (Mod 1)
--------------------------------------------------------------------------------
                                      Used    Free    Percent Utilization
--------------------------------------------------------------------------------
Ingress L2 QOS                        2       254     0.78
 Ingress L2 QOS IPv4                  0               0.00
 Ingress L2 QOS IPv6                  0               0.00
 Ingress L2 QOS MAC                   0               0.00
 Ingress L2 QOS ALL                   2               0.78
 Ingress L2 QOS OTHER                 0               0.00
Ingress L2 SPAN ACL                   0       256     0.00
Ingress RACL                          2       1534    0.13
Ingress L3/VLAN QOS                   24      488     4.68
Ingress L3/VLAN SPAN ACL              0       256     0.00
SPAN                                  0       512     0.00
Egress RACL                           2       1790    0.11
Feature BFD                           3       103     2.83
<snip>
```

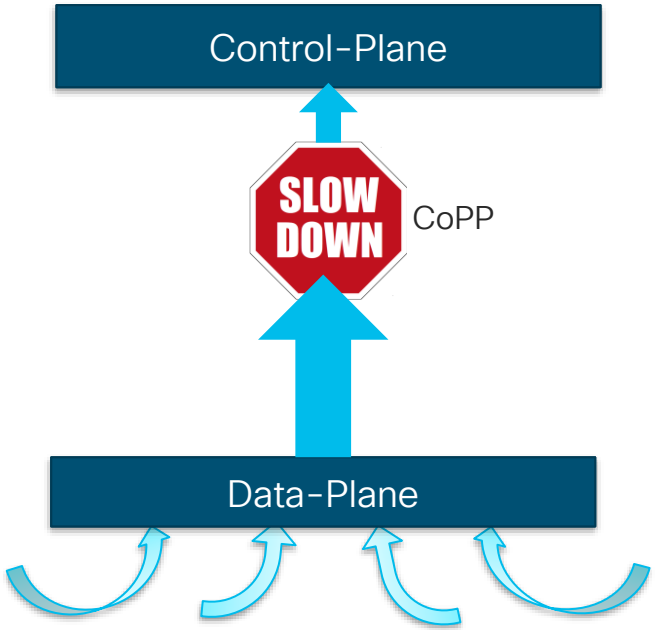
```
<snip>
LOU
 Both LOU Operands
 Single LOU Operands
 LOU L4 src port:
 LOU L4 dst port:
 LOU L3 packet len:
 LOU IP tos:
 LOU IP dscp:
 LOU ip precedence:
 LOU ip TTL:
TCP Flags
Protocol CAM
Mac Etype/Proto CAM
L4 op labels, Tcam 0
L4 op labels, Tcam 1
Ingress Dest info table
Egress Dest info table
<snip>
```

Control-Plane Policing (CoPP)

Things to Check

No License Required

- Choose either ***strict*** (default), ***moderate***, ***lenient*** or ***dense*** policy.
- CoPP is performed per forwarding-engine. Configure rates to make sure the aggregate traffic doesn't overwhelm CPU.
- Monitor drop counters continuously and justify drop counters.
- Remember… CoPP configuration is an on-going process.



Control-Plane Policing (CoPP)

Quick Check – Config and Stats

```
N9504# show copp status
Policy-map attached to the control-plane: copp-system-p-policy-strict
(match-any)

N9504# show policy-map interface control-plane | include class-map
    class-map copp-system-p-class-l3uc-data (match-any)
    class-map copp-system-p-class-critical (match-any)
    class-map copp-system-p-class-important (match-any)
    class-map copp-system-p-class-multicast-router (match-any)
    class-map copp-system-p-class-multicast-host (match-any)
    class-map copp-system-p-class-l3mc-data (match-any)
    class-map copp-system-p-class-normal (match-any)
    class-map copp-system-p-class-ndp (match-any)
    <snip>
    class-map copp-system-p-class-redirect (match-any)
    class-map copp-system-p-class-exception (match-any)
    class-map copp-system-p-class-exception-diag (match-any)
    <snip>
    class-map copp-system-p-class-undesirablev6 (match-any)
    class-map copp-system-p-class-l2-default (match-any)
```

Control-Plane Policing (CoPP)

Quick Check – Config and Stats (Contd.)

```
N9504# show policy-map interface control-plane
<snip>
  class-map copp-system-p-class-important (match-any)
      match access-group name copp-system-p-acl-hsrp
      match access-group name copp-system-p-acl-vrrp
      match access-group name copp-system-p-acl-hsrp6
      match access-group name copp-system-p-acl-vrrp6
      match access-group name copp-system-p-acl-mac-lldp
      match access-group name copp-system-p-acl-icmp6-msgs
      set cos 6
      police cir 3000 pps , bc 128 packets
      module 1 :
        transmitted 2121674 packets;
        dropped 143189 packets;
<snip>
  class-map class-default (match-any)
      set cos 0
      police cir 50 pps , bc 32 packets
      module 1 :
        transmitted 2231318 packets;
        dropped 4239 packets;
```

Do “clear copp statistics” and check again!

Hardware Rate-Limiters (HWRL)

Things to Check

No License Required

- Rate-limiters prevent redirected-due-to-exception packets from overwhelming CPU. E.g., ACL Log or Layer3 Glean

```
N9504# show hardware rate-limiter
Units for Config: packets per second (kilo bits per
second for span-egress)
Allowed, Dropped & Total: aggregated since last
clear counters
Module: 1
  R-L Class              Config     Allowed   Dropped     Total
 +----------------+----------+-------------+----------+--------+
  L3 MTU                    0           0          0          0
  L3 ttl                  500          65          0         65
  L3 glean                100    28874211    9539369   38413580
  L3 mcast loc-grp       3000           0          0          0
  access-list-log         100           0          0          0
  bfd                   10000           0          0          0
  exception                50           0          0          0
  span                     50           0          0          0
  <snip>
```

Have a close look at the allowed and dropped stats

Enable/disable or update rates with “**hardware rate-limiter …** ” config command.

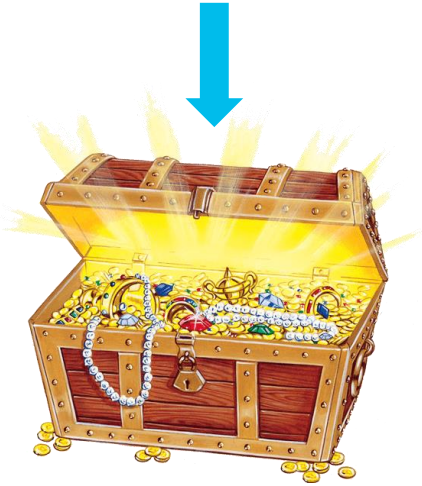
Clear stats with “**clear hardware rate-limiter …**” command.

Monitor and Health Check
Summary

- Hardware diagnostic capabilities… bootup, runtime and on-demand. Help to check hardware failure and run-time issues.
- OBFL helps to keep an eye on the systems' events and exceptions. Critical for analysis.
- Monitoring resource usage is critical, and it helps to implement precautionary measures
- Fine-tune CoPP and HWRL to protect control-plane and ensure stability

"**show tech-support detail**" command captures detailed hardware diagnostics results, OBFL, hardware capacity and usage, CoPP and HWRL statistics.

Never underestimate the power of **syslog** (*show logging log*), **interface counters and errors** (*show interface*) or **memory/CPU usage** (show process memory/CPU)



You are going to find valuable things!!


Nexus 9000
... platform of possibilities

Troubleshooting Tools

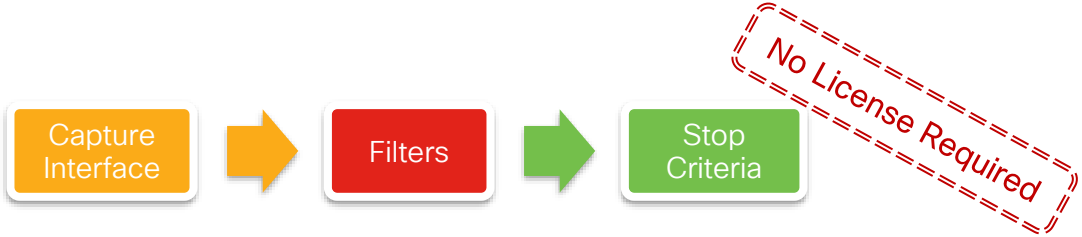
Agenda

- Introduction
- Monitor and Health-Check
- Troubleshooting Tools
- Troubleshooting Traffic Forwarding
- Best Practices and Recommendations
- Summary and Take-Aways

- Ethanalyzer
- SPAN to CPU
- Consistency Checkers
- Virtual TAC Assistant
- Port ACL / Router ACL

Ethanalyzer

Process and Configuration



(1) Identify Capture Interface

- mgmt – captures traffic on mgmt0 interface
- Inband – captures traffic sent to and received from the control-plane/CPU

(2) Configure Filter

- Display-Filter – captures all traffic but displays only the traffic meeting the criteria
- Capture-Filter – captures only the traffic meeting the criteria

(3) Define Stop Criteria

- By default, it stops after capturing 10 frames. Can be changed with **limit-captured-frames** configuration. 0 means no limit, runs until user issues **cntrl+C**
- **autostop** can be used, to stop the capture after specified duration, filesize, or number of files.

Ethanalyzer
Introduction

- Built-in tool to analyze the traffic sent and received by CPU. Helpful to troubleshoot High CPU or Control-plane issues like HSRP failover or OSPF adjacency flaps.
- Based on tshark code
- Two filtering approaches for configuring a packet capture

Display-Filter Example	Capture-Filter Example
“eth.addr==00:00:0c:07:ac:01”	“ether host 00:00:0c:07:ac:01”
“ip.src==10.1.1.1 && ip.dst==10.1.1.2”	“src host 10.1.1.1 and dst host 10.1.1.2”
“snmp”	"udp port 161”
“ospf”	“ip proto 89”

Ethanalyzer
Putting It All Together

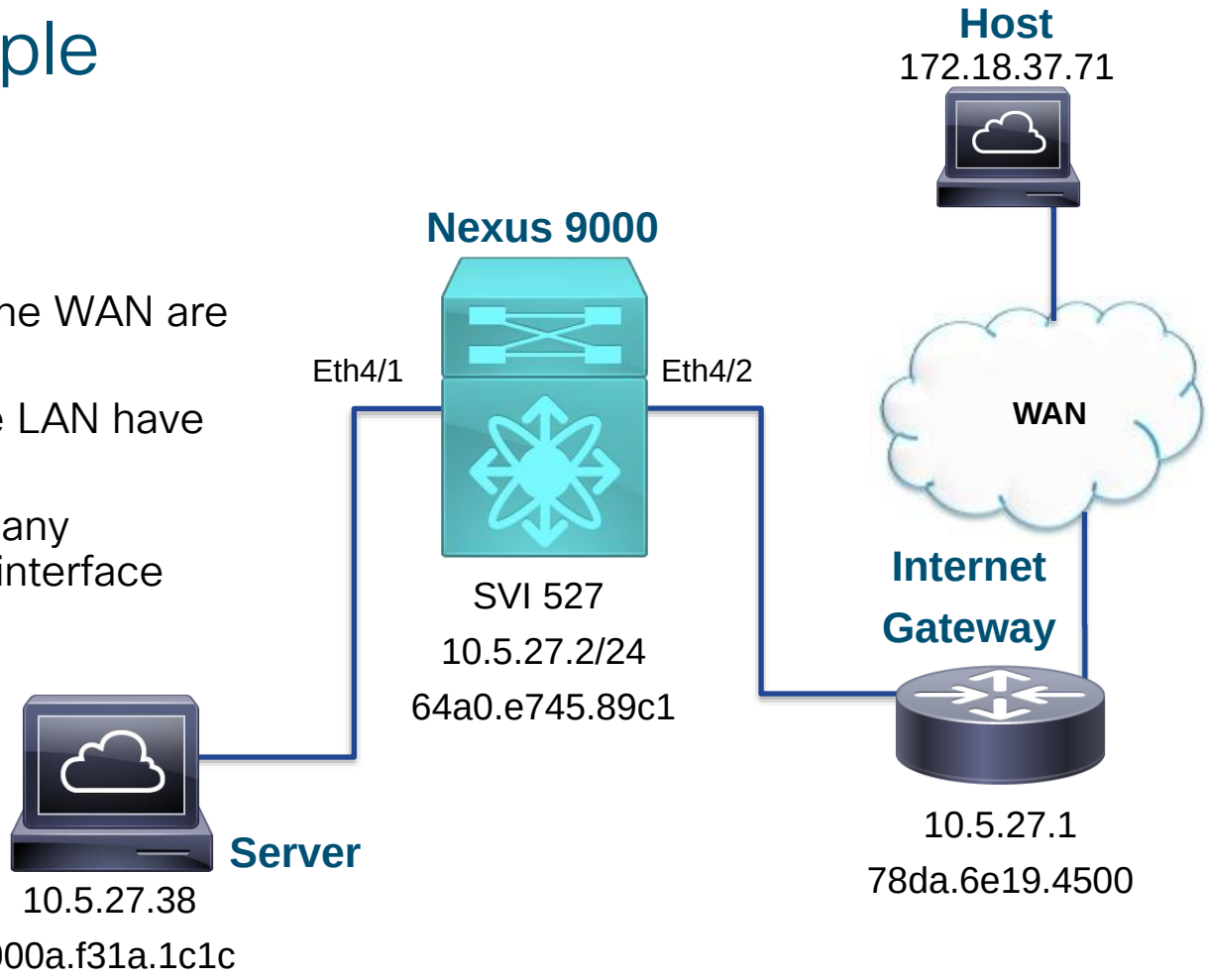
```
N9K# ethanalyzer local interface inband display-filter "stp" limit-captured-frames 0 capture-
ring-buffer filesize 200 write bootflash:stp_ring.pcap display autostop files 5
```

- Captures on the inband interface
- Uses a display-filter searching for “stp” frames
- Sets limit-captured-frames to zero to allow continuous capturing of frames
- Uses a capture-ring-buffer to create a new file every 200 KB
- Write files to bootflash:stp_ring.pcap, adding a timestamp as a prefix
- autostop after 5 files have been created

Real World Example

Slow Download Rate

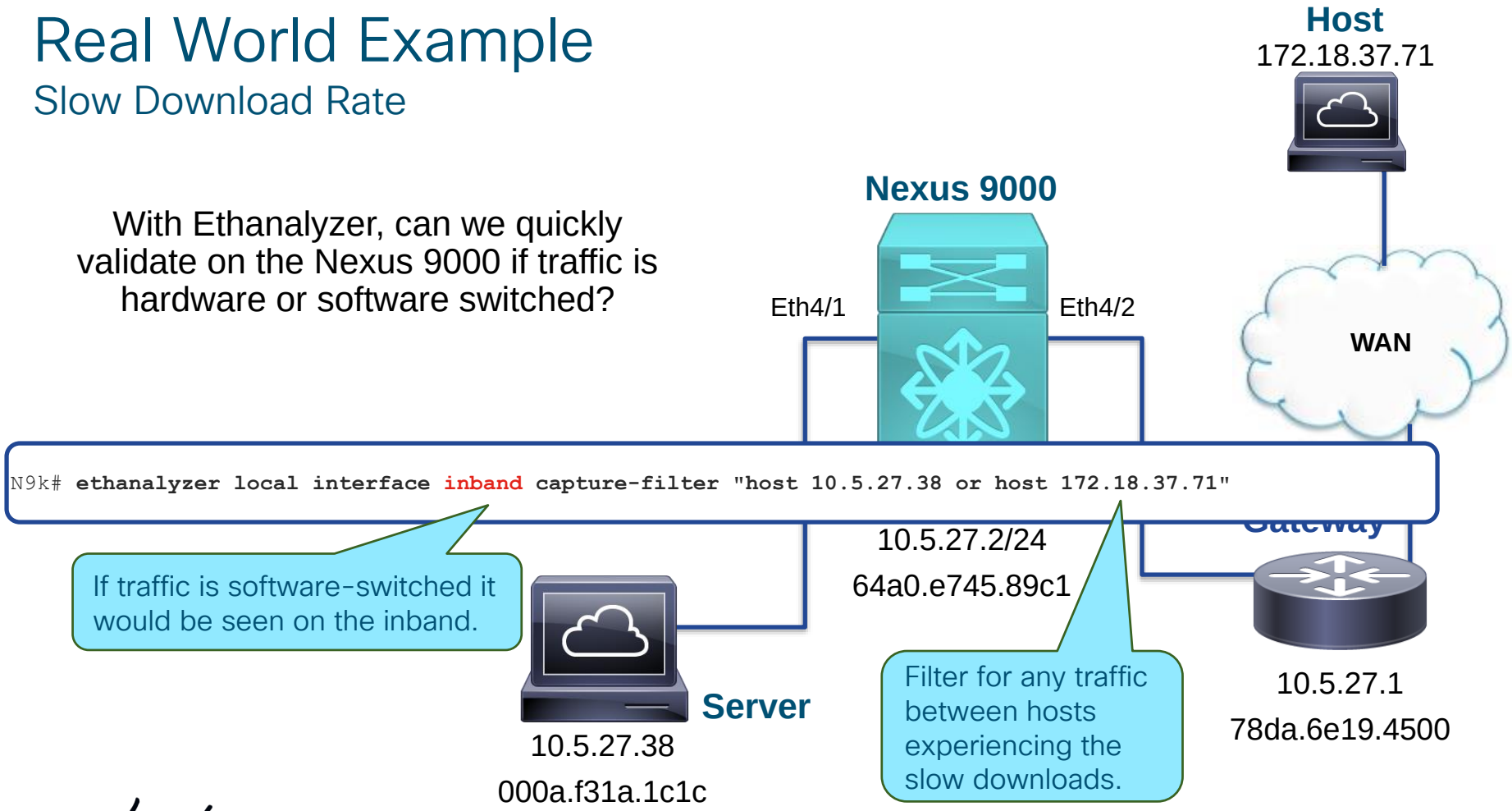
- Server in VLAN 527
- Downloads/Uploads over the WAN are slow
- Downloads/Uploads on the LAN have no problem
- No incrementing errors on any interface and low average interface utilization



Real World Example

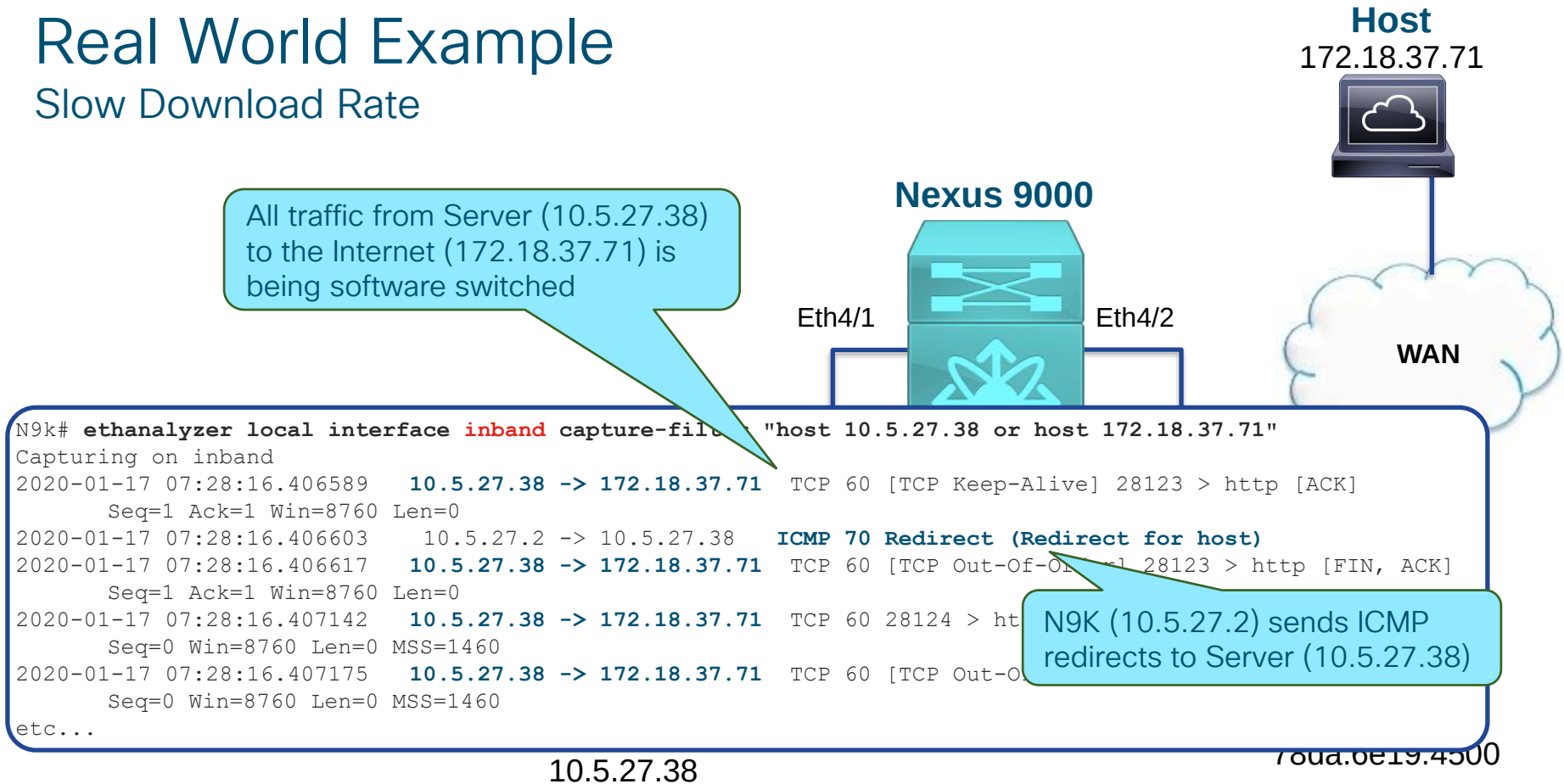
Slow Download Rate

With Ethanalyzer, can we quickly validate on the Nexus 9000 if traffic is hardware or software switched?



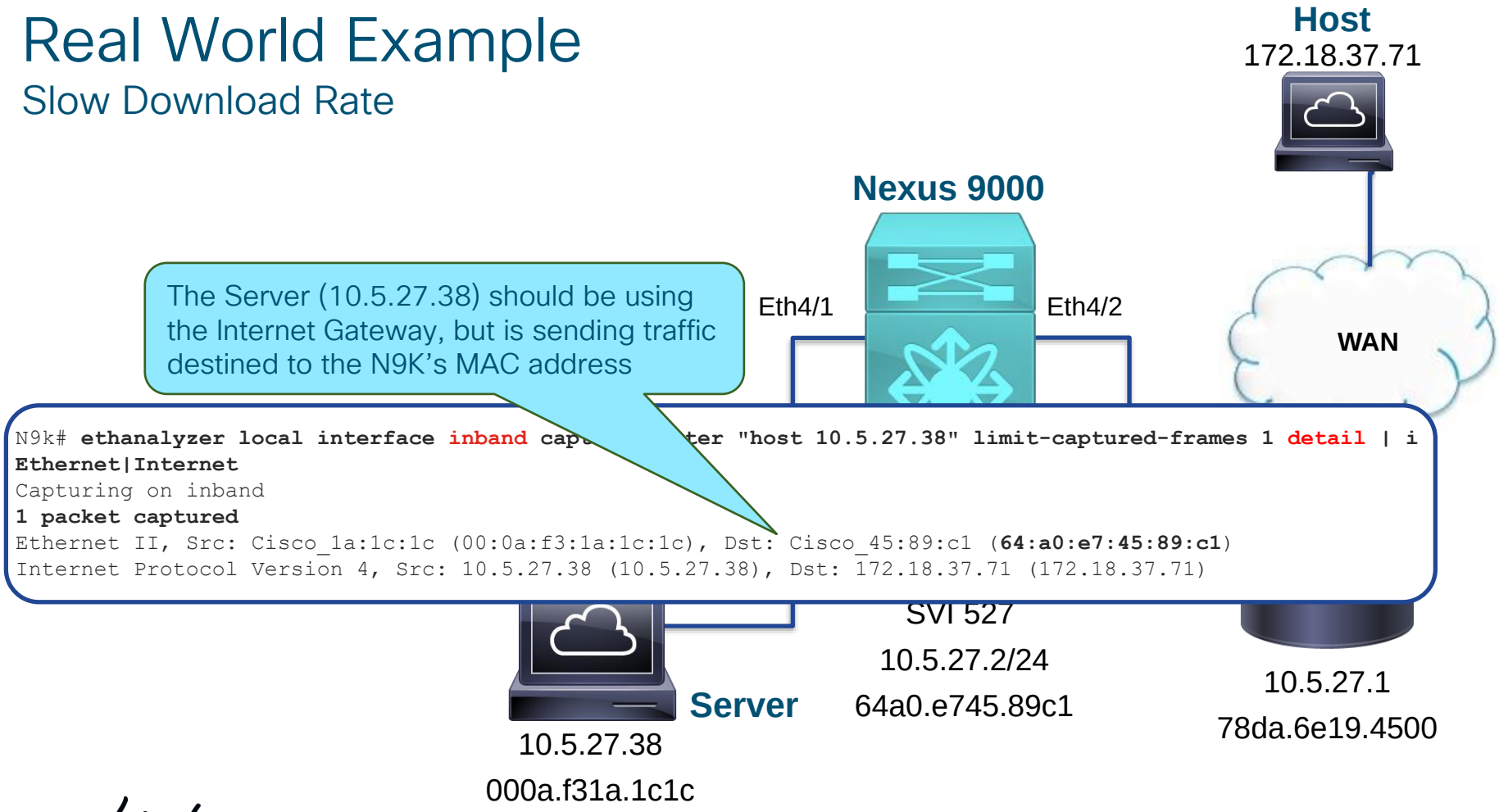
Real World Example

Slow Download Rate



Real World Example

Slow Download Rate



Real World Example

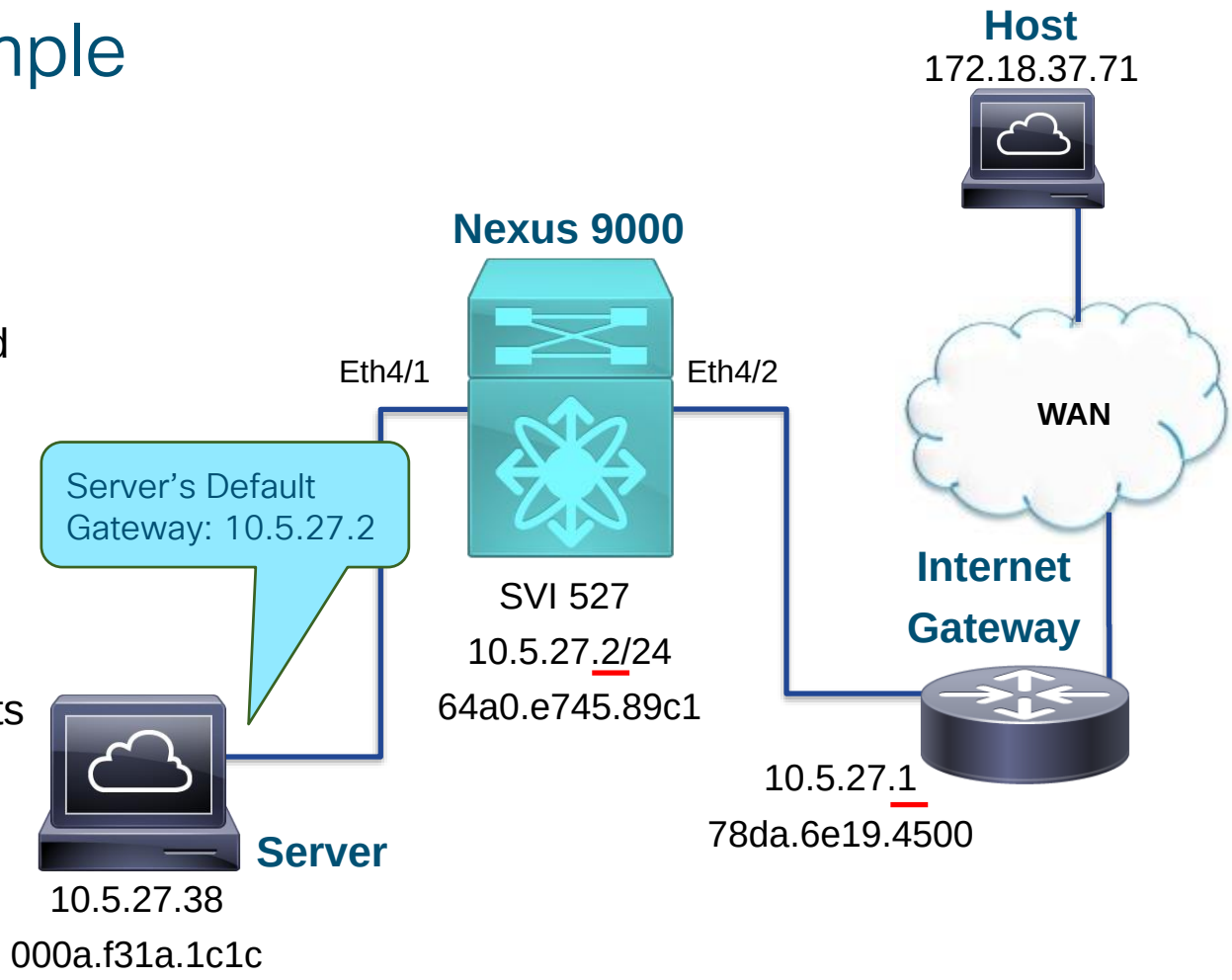
Slow Download Rate

Root cause:

- Server has a firewall enabled to block ALL ICMP Redirects to avoid poisoning

Fix Options:

1. Re-configure the Server's firewall to allow ICMP redirects
2. Add a route for WAN subnets to the Server, with Internet Gateway as next-hop
3. Configure "no ip redirects" under the SVI VLAN527

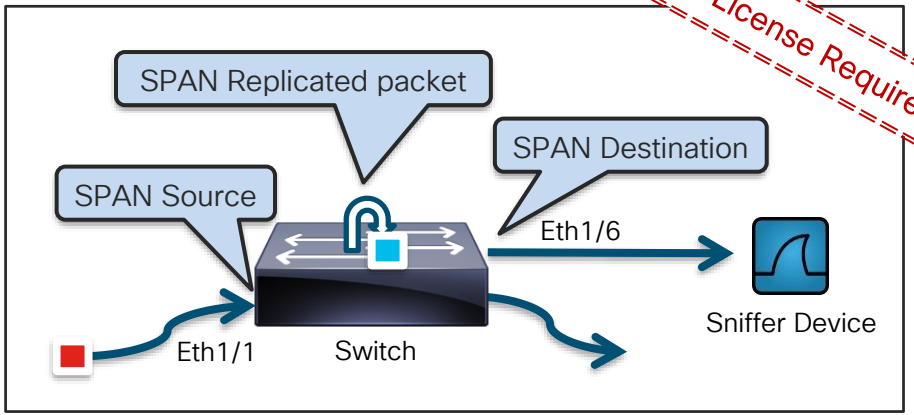


SPAN to CPU

Introduction and Configuration

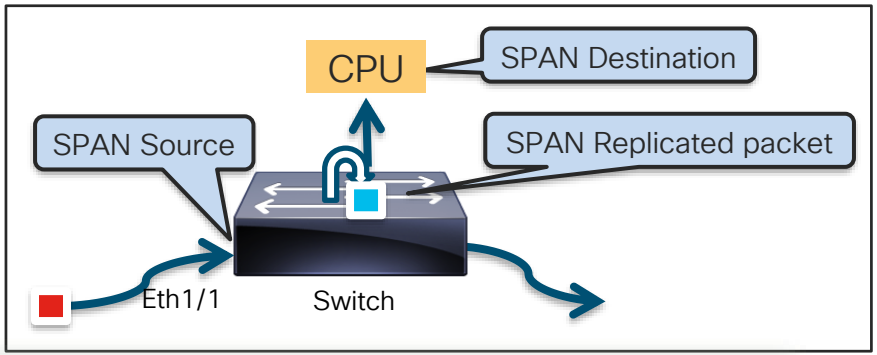
Switch **P**ort **AN**alyzer (SPAN) mirrors the traffic from source ports/VLANs to destination port(s).

```
monitor session 1
  source interface eth1/1
  destination interface eth1/6
```



In SPAN to CPU, the destination port is the CPU in the switch.

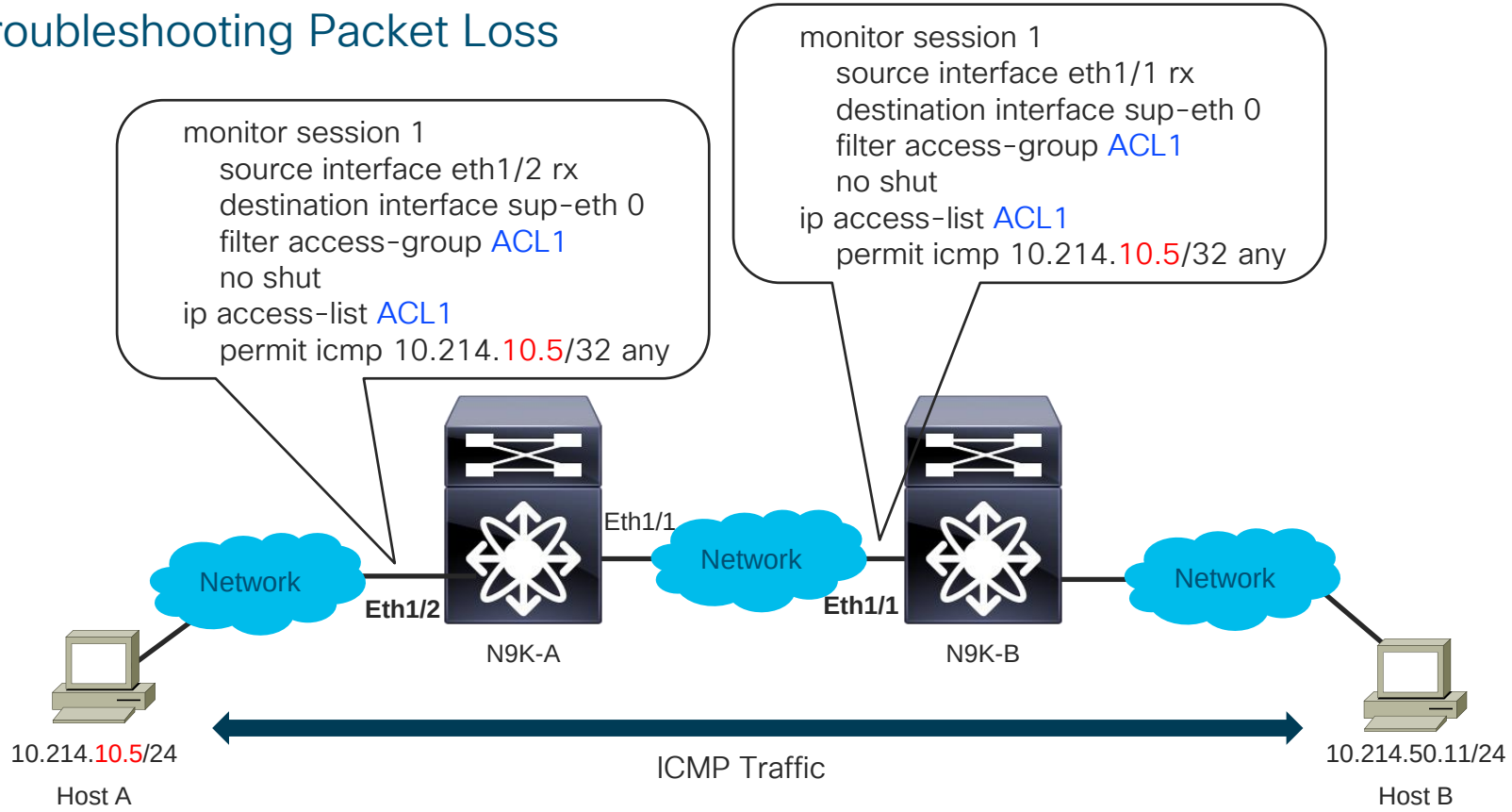
```
monitor session 1
  source interface eth1/1
  destination interface sup-eth 0
  <options>
```



But, how to differentiate the regular control-plane packets to SPAN to CPU packets?

SPAN to CPU

Troubleshooting Packet Loss

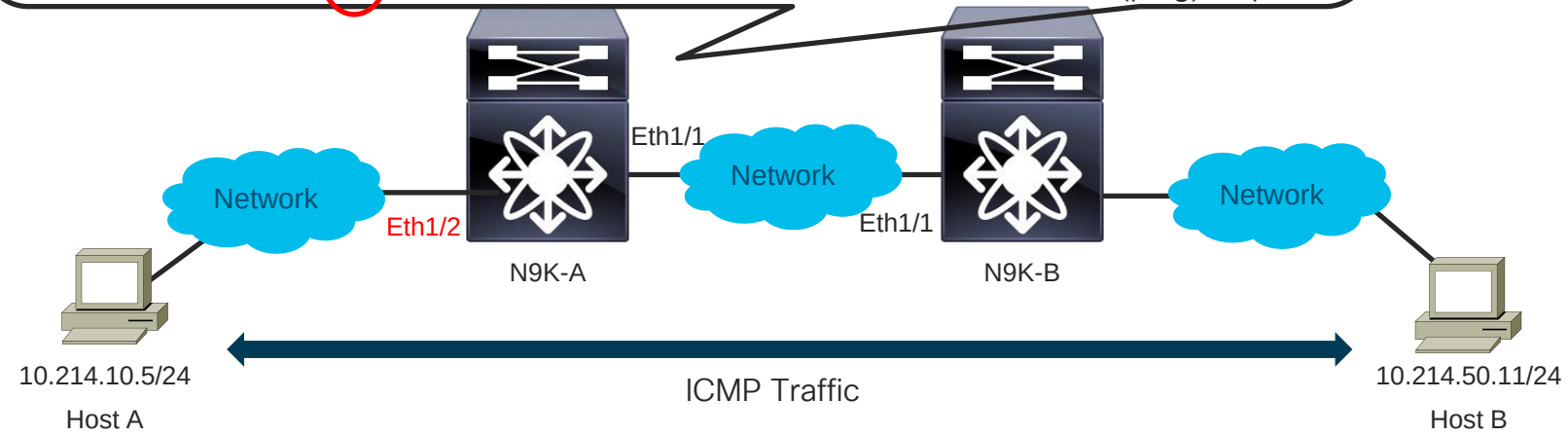


SPAN to CPU

Troubleshooting Packet Loss

Captures only the **SPAN to CPU** packets, not regular packets!!

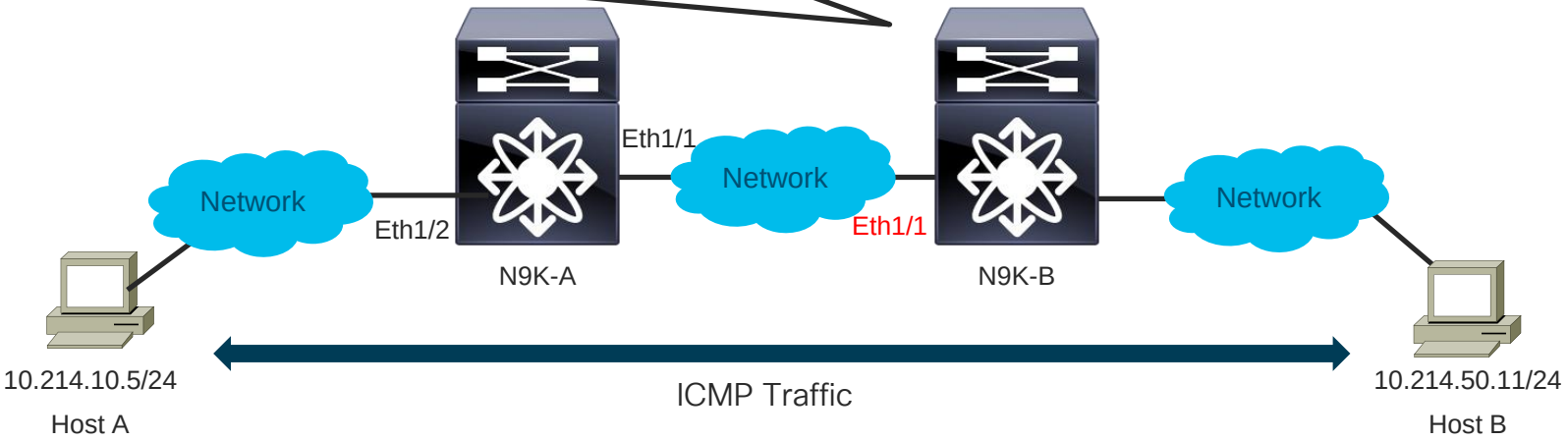
```
N9K-A# ethanalyzer local interface inband mirror display-filter "icmp"
Capturing on inband
2019-11-19 03:47:21.164790  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
2019-11-19 03:47:21.165562  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
2019-11-19 03:47:21.166266  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
2019-11-19 03:47:21.166930  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
2019-11-19 03:47:23.167589  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
```



SPAN to CPU

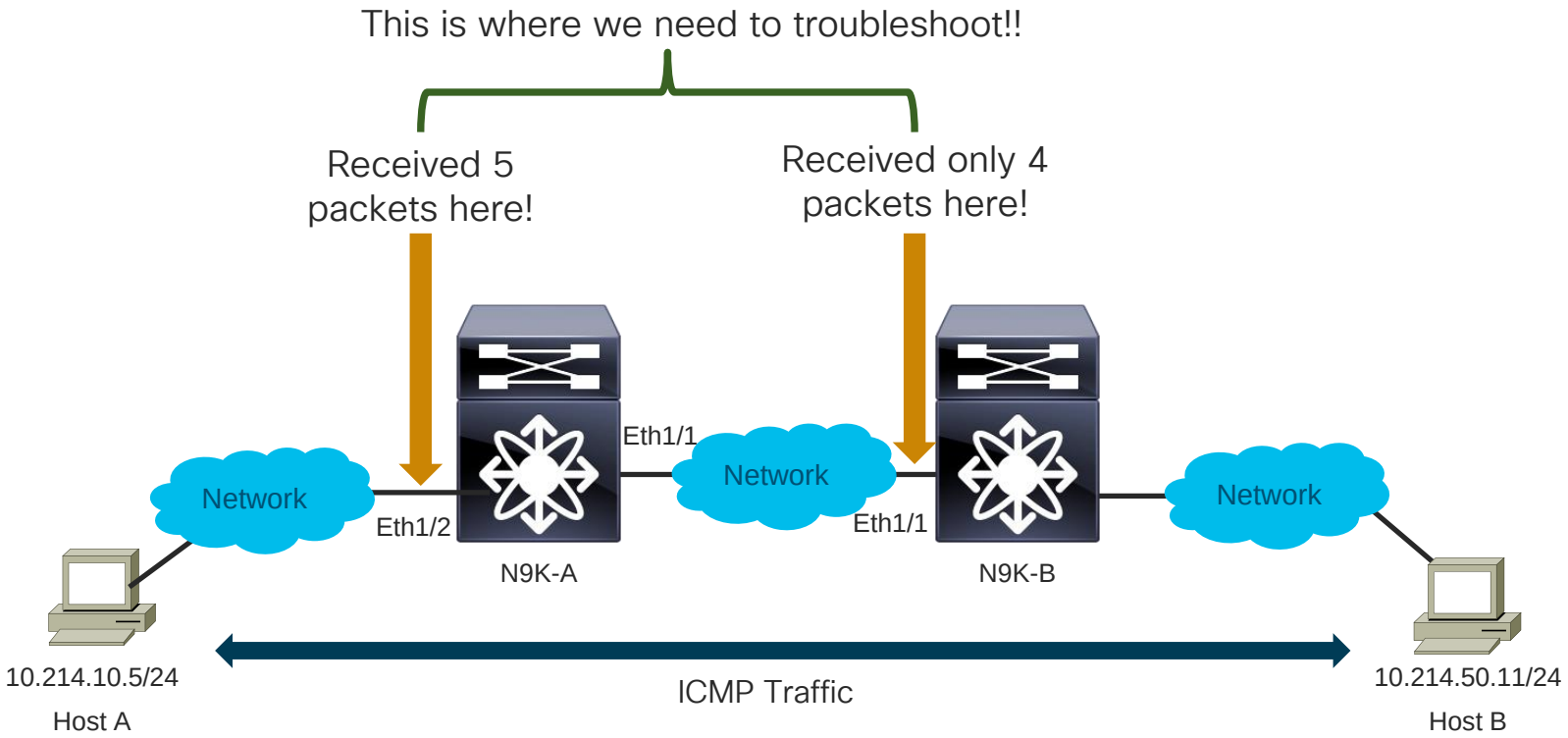
Troubleshooting Packet Loss (contd.)

```
N9K-B# ethanalyzer local interface inband mirror display-filter "icmp"
Capturing on inband
2019-11-19 03:47:21.164982  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
2019-11-19 03:47:21.165941  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
2019-11-19 03:47:21.166611  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
2019-11-19 03:47:23.167992  10.214.10.5 -> 10.214.50.11 ICMP Echo (ping) request
```



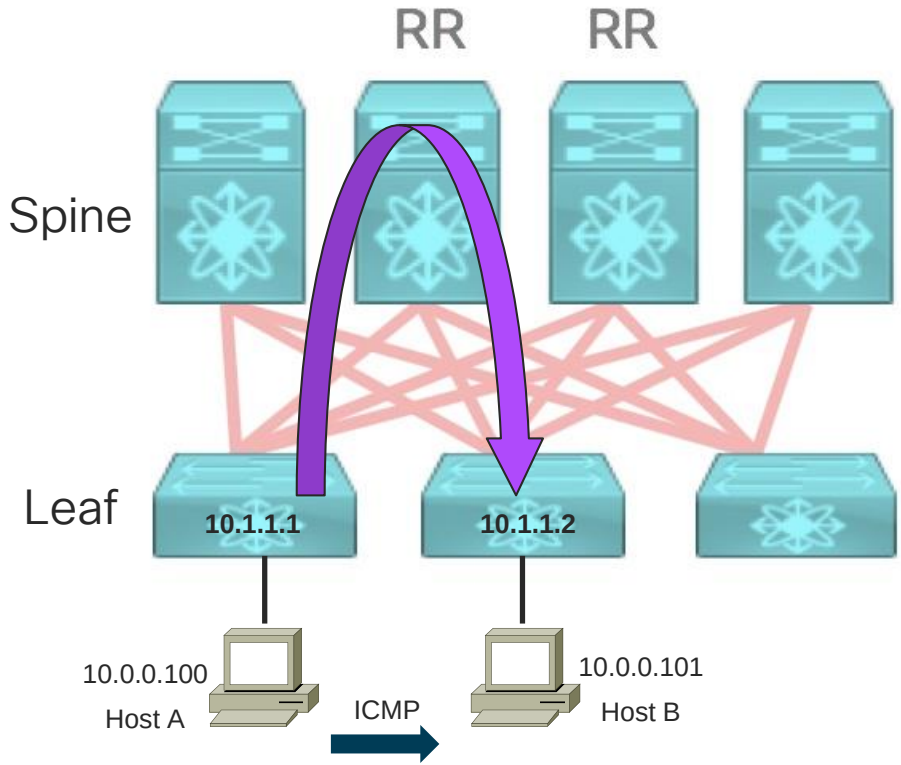
SPAN to CPU

Narrow-scoped Troubleshooting



SPAN to CPU

VXLAN – Topology and Traffic Flow



SPAN to CPU

VXLAN Decode Example

Available in release 7.0(3)I7(4), 9.2(1) and later releases

```
N9200# ethanalyzer local interf inband mirror display-filter icmp limit-cap 0 detail
Frame 1 (148 bytes on wire, 148 bytes captured)
<snip>
    [Protocols in frame: eth:ip:udp:vxlan:eth:ip:icmp:data]   <<< frame structure
Ethernet II, Src: 78:0c:f0:a2:2b:df (78:0c:f0:a2:2b:df), Dst: 70:0f:6a:f2:8c:05
(70:0f:6a:f2:8c:05)
    <snip>
    Type: IP (0x0800)
Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 10.1.1.2 (10.1.1.2)   <<< VTEPs
    Version: 4
    Header length: 20 bytes
    <snip>
    Source: 10.1.1.1 (10.1.1.1)
    Destination: 10.1.1.2 (10.1.1.2)
User Datagram Protocol, Src Port: 22790 (22790), Dst Port: 4789 (4789)   <<< VXLAN Attributes
    Source port: 22790 (22790)
    Destination port: 4789 (4789)
<snip>
```

SPAN to CPU

VXLAN Example (Contd.)

```
Virtual eXtensible Local Area Network
    Flags: 0x08
    <snip>
    VXLAN Network Identifier (VNI): 10990010  <<< VNI for vlan 10
    Reserved: 0
Ethernet II, Src:, 00:aa:aa:aa:10:10 (00:aa:aa:aa:10:10) Dst: 00:bb:bb:bb:20:20  <<< Inner MAC
(00:bb:bb:bb:20:20)
    <snip>
    Type: IP (0x0800)
Internet Protocol, Src: 10.0.0.100 (10.0.0.100), Dst: 10.0.0.101 (10.0.0.101)  <<< Inner IPs
<snip>
    Source: 10.0.0.100 (10.0.0.100)
    Destination: 10.0.0.101 (10.0.0.101)
Internet Control Message Protocol    <<< Original ICMP
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0x3597 [correct]
    Identifier: 0xb00f
<snip>
```

SPAN to CPU

Things to Know

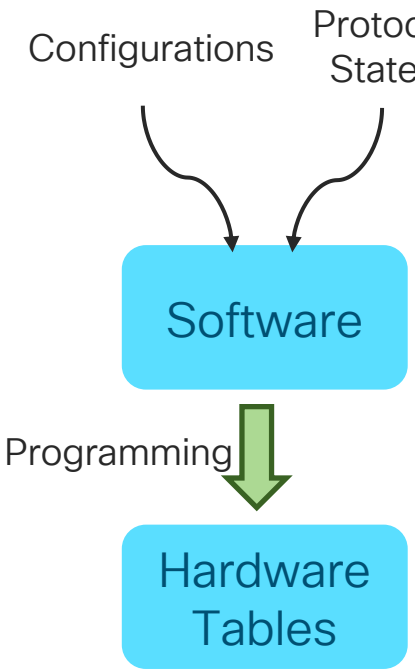
- All SPAN replication is done in the hardware with no impact to CPU
- SPAN packets to CPU are rate-limited, and excess packets are dropped in the inband path. Use *"hardware rate-limiter span …"* command to change the rate.
- Starting from 7.0(3)I7(1) onwards, SPAN packets truncation is supported only in Nexus 9300-EX/FX/FX2 platforms
- SPAN is not supported for management ports

Consistency Checkers

What it does?

No License Required

Consistency Checkers compares the software state against the hardware state for consistency, and report PASSED or FAILED.




```
N9K# show consistency-checker ?
  copp             Verify copp programming from software context
  egress-xlate     Check PVLAN egress-xlate
  fex-interfaces   Compares software and hardware state of fex interfaces
  forwarding       Display Forwarding Information
  l2               L2 consistency
  l3               L3 consistency
  l3-interface     Compares software and hardware properties of L3 interf
  link-state       Compares software and hardware link state of interfaces
  membership       Check various memberships  <-- VLANs, Port-Channel
  pacl             Verify pacl programming in the hardware
  racl             Verify racl programming in the hardware
  stp-state        Verify spanning tree state in the hardware
  vacl             Verify vacl programming in the hardware
  vpc              Verify vpc state in the hardware
  vxlan            VxLAN consistency checker
```

Consistency Checkers
Example – Unicast Route and vPC

Consistency-Checker for an IP address. Same can be used for a prefix.

```
N9K# show consistency-checker forwarding single-route ipv4 10.127.101.1 prefix 32 vrf
L3-Inner
Starting consistency check for v4 route 10.127.101.1/32 in vrf L3-Inner
Consistency checker passed for 10.127.101.1/32
```

Consistency-Checker for vPC

```
N9K# show consistency-checker vpc source-interface port-channel 45
VPC 45 name Po45
    Validating vpc 45 member: Ethernet1/1/3
Error vpc 45, is_vpc is not 1 and remote vpc state is Up
VPC Consistency Check Failed
```

Virtual TAC Assistant

Commands Cascading

What is it?

- It takes output and parameters from one command and pass them on to the next command as inputs and cascade them through the entire sequence of troubleshooting.

How it helps with troubleshooting?

- speeds up troubleshooting
- avoids missing out commands
- avoids entering wrong commands inputs
- no need to know the procedure or methodology

Virtual TAC Assistant

L2 MAC – Command Options

```
DC2-VTEP# show troubleshoot ?
  L2  Display L2 information
  L3  Display L3 information

DC2-VTEP# show troubleshoot L2 ?
  mac  MAC address

DC2-VTEP# show troubleshoot L2 mac ?
  E.E.E              Address (Option 1)
  EE-EE-EE-EE-EE-EE  Address (Option 2)
  EE:EE:EE:EE:EE:EE  Address (Option 3)
  EEEE.EEEE.EEEE     Address (Option 4)

DC2-VTEP# show troubleshoot L2 mac 0001.0203.0405 vlan 100 ?
  <CR>
  >       Redirect it to a file
  >>      Redirect it to a file in append mode
  detail  Print detailed debugging info for mac/interface
  |       Pipe command output to filter
```

Virtual TAC Assistant

L2& L3 – Command Options

Validates programming of a MAC Address in a given VLAN

```
DC2-VTEP# show troubleshoot ?
  L2  Display L2 information
  L3  Display L3 information

DC2-VTEP# show troubleshoot L3 ?
  ipv4  Choose IPv4 address
  ipv6  Choose IPv6 address

DC2-VTEP# show troubleshoot L3 ipv4 172.16.144.254 ?
  src-ip  Source IP for routing hash CLI
  vrf     Check routes for a specific VRF

DC2-VTEP# show troubleshoot L3 ipv4 172.16.144.254 vrf ?
  WORD  Vrf name

DC2-VTEP# show troubleshoot L3 ipv4 172.16.144.254 vrf tenant-1 ?
  <CR>
  >      Redirect it to a file
  >>     Redirect it to a file in append mode
  |      Pipe command output to filter
```

Virtual TAC Assistant

Example - L3 IPv4

```
DC2-VTEP# show troubleshoot L3 ipv4 172.16.144.254 vrf tenant-1
CHECKING HARDWARE ASIC TYPE slot 1 quoted "show hardware internal dev-version”
<snip>
CHECK ROUTE IN PI RIB show ip route 172.16.144.254 vrf tenant-1
<snip>
CHECK ROUTE IN PD FIB show forwarding route 172.16.144.254/32 vrf tenant-1
<snip>
CHECK HOST ROUTE IN HARDWARE show hardw internal tah L3 v4host | grep 172.16.144.254
<snip>
CHECK FOR THE ADJACENCY show hardware internal tah l3 adjacency 0xd0001"
<snip>
CHECK ROUTE IN SOFTWARE PT
sh hardw internal tah l3 trie detail 172.16.144.254/32 table 3"
<snip>
CHECK FOR THE ROUTE IN E-TABLE
show hardware internal tah sdk l3 sw-table e-table | grep 172.16.144.254"
<snip>
CHECK FOR THE ROUTE IN HASH-TABLE
show hardware internal tah sdk l3 sw-table ipv4 hash-table | grep 172.16.144.254"
<snip>
RUNNING CONSISTENCY CHECKER
Consistency checker passed for 172.16.144.254/32
```

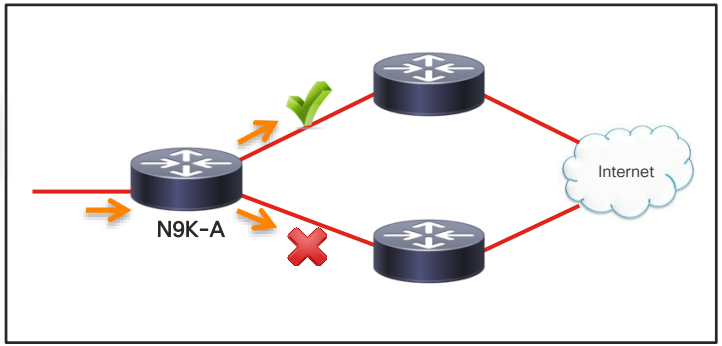
Step-by-step methodical check

Virtual TAC Assistant

Example – ECMP Hardware Programming Failure Detection

```
N9K-A# show troubleshoot L3 ipv4 0.0.0.0 vrf default
<snip>
************************************
CHECK ROUTE IN PI RIB
************************************
show ip route 0.0.0.0 vrf default
<snip>
0.0.0.0/0, ubest/mbest: 2/0
    *via 10.3.25.33, [1/0], 3w4d, static
    *via 10.3.25.37, [1/0], 3w3d, static
<snip>
************************************
CHECK ROUTE IN HARDWARE TCAM
************************************
show hardware internal tah l3 v4lpm prefix 0.0.0.0/0 table 1
<snip>
 Idx         | vrf      | ip/len          | mpath |   nump  | base/l2ptr  |cc|sr|dr|td|dc|de|li|hr|
 ------------|----------|-----------------|-------|---------|-------------|--|--|--|--|--|--|--|--|
 6/2043      |        1 | 0.0.0.0/0       | 0     | 256     | 0x2000001f  |  |  |  |  |  |  |  |  |
<snip>
Consistency checker failed for 0.0.0.0/0
```

Two entries.. ECMP!



Only one entry in the hardware

Port ACL / Router ACL

Tool and Requirements

- For intermittent packet loss issue specifically in scenarios where the exact packet count can be defined, Router ACL (RACL) and Port ACL (PACL) can be a useful tool
- Requires TCAM allocation for PACL followed by switch reload.

```
N9K(config-if)#  ip port access-group test1 in
ERROR: TCAM region is not configured. Please
configure TCAM region and retry the command
```

TCAM space is limited. The choice for what is best for you depends entirely on the specific use-case. By default, all TCAM space is already allocated, so you need to decide where you want to 'steal' TCAM space from in order to allocate elsewhere.

No License Required

```
N9K# show run | include ignore-case tcam
hardware access-list tcam region ing-ifacl  512
hardware access-list tcam region ing-racl  1024
```

IFACL = PACL

```
N9K# show system internal access-list globals
<snip>
------------------------------------------------------------
                INSTANCE 0 TCAM Region Information:
------------------------------------------------------------
Ingress:
--------
                Region      TID      Base      Size    Width
------------------------------------------------------------
                   NAT       13         0         0        1
           Ingress PACL        1         0       512        1
           Ingress VACL        2         0         0        1
           Ingress RACL        3       512      1024        1
         Ingress RBACL         4         0         0        1
        Ingress L2 QOS         5      1536       256        1
    Ingress L3/VLAN QOS        6      1792       512        1
<snip>
Total configured size: 4096
Remaining free size: 0
Note: Ingress SUP region includes Redirect region

Egress:
--------
<snip>
```

Port ACL / Router ACL

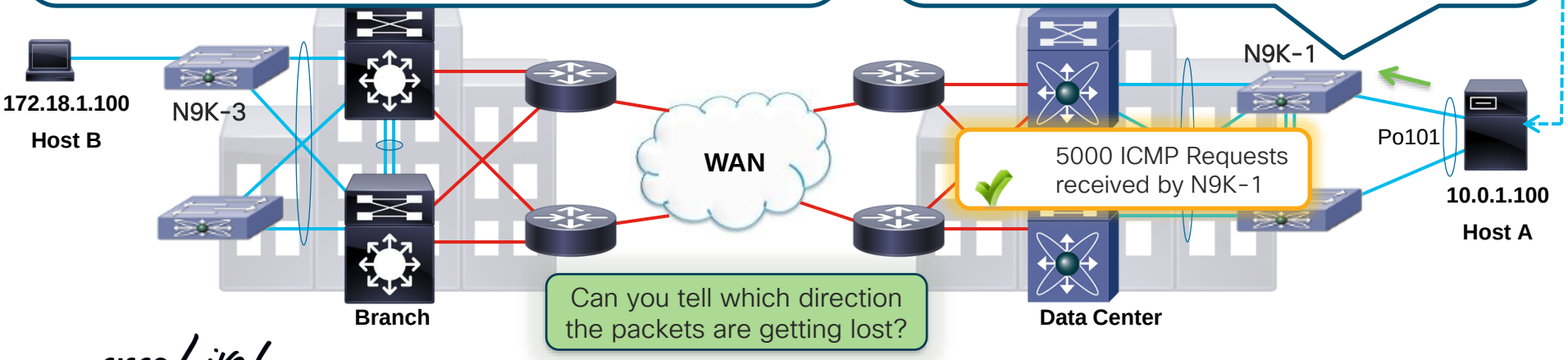
Troubleshooting Packet Loss

```
root@Server~$ ping 172.18.1.100 -c 5000 -W 1 -i 0
<snip>
5000 packets transmitted, 4886 packets received, 0.2% packet loss,
```

Using a Port-ACL (PACL) to match bridged traffic on an L2 switchport

```
N9K-1# show ip access-lists 101
IPV4 ACL 101
  statistics per-entry
  10 permit icmp 10.0.1.100/32 172.18.1.100/32 [match=5000]
  20 permit ip any any [match=323321]
```

```
ip access-list 101
  statistics per-entry
  10 permit icmp 10.0.1.100/32 172.18.1.100/32
  20 permit ip any any
! Apply to server ingress interface
interface port-channel101
  ip port access-group 101 in
```



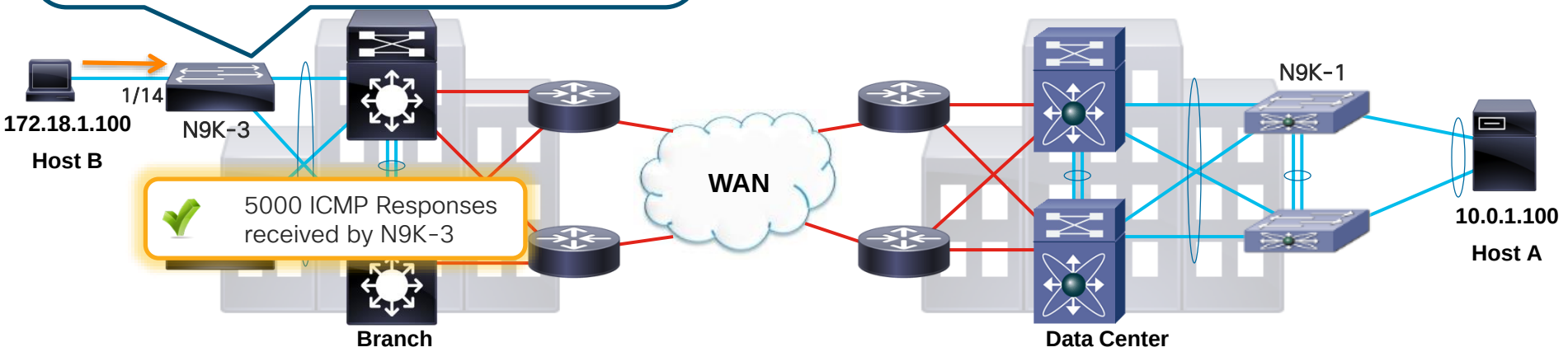
Port ACL / Router ACL

Troubleshooting Packet Loss (Contd.)

Using a Port-ACL (PACL) to match bridged traffic on an L2 switchport

```
ip access-list 101
  statistics per-entry
  10 permit icmp 172.18.1.100/32 10.0.1.100/32
  20 permit ip any any
! Apply to server ingress interface
interface Ethernet1/14
  ip port access-group 101 in
```

```
N9K-3# show ip access-lists 101
IPV4 ACL 101
  statistics per-entry
  10 permit icmp 172.18.1.100/32 10.0.1.100/32 [match=5000]
  20 permit ip any any [match=221747]
```



More Tools

- SPAN / ERSPAN, SPAN-on-Drop
- Embedded Logic Analyzer Module (ELAM)
- Flow Tracer
- VXLAN, DME and KSTACK Consistency Checkers
- Streaming Hardware Telemetry
- Flexible Netflow / sFlow

Tools and Supported Products

Summary

Tool	Supported in Nexus 9000 (Broadcom)?	Supported in Nexus 9000 (Tahoe/Rocky)?	Impact
Ethanalyzer	yes	yes	
SPAN to CPU	yes[1]	yes	
Consistency Checkers	yes[2]	yes	
Virtual TAC Assistant	yes[2]	yes	
PACL/RACL[3]	yes	yes	

1 = “dMirror” feature
2 = Limited capabilities
3 = TCAM carving needed


Nexus 9000
... platform of possibilities

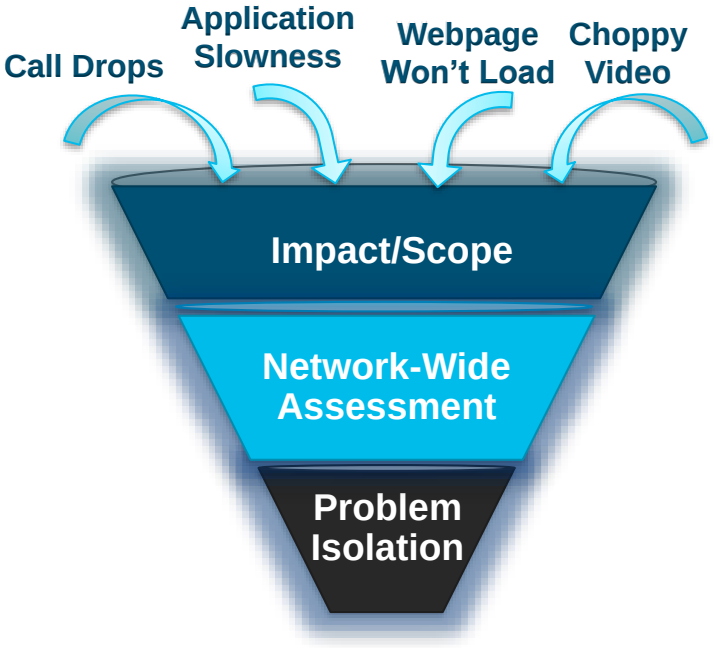
Troubleshooting Traffic Forwarding

"It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts."

Sherlock Holmes (A Scandal in Bohemia)

Troubleshooting Methodology

- Define the problem, understand the impact, and determine the scope of the problem based on the information gathered. This helps you to make progress towards resolution.
- Perform network-wide assessment. Check SNMP, syslogs, Netflow data, real-time performance/SLA monitoring tools for alerts, unexpected events, threshold violations etc.
- Choose the right tool(s) and troubleshooting procedure(s) to isolate the problem at a granular level and diagnose to achieve a **fast resolution.**



Agenda

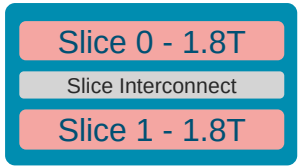
- Introduction
- Monitor and Health-Check
- Troubleshooting Tools
- Troubleshooting Traffic Forwarding
- Best Practices and Recommendations
- Summary and Take-Aways

- Nexus 9000 Hardware Forwarding – Refresher
- Path-of-the-Packet Troubleshooting
  - Control-Plane Traffic
  - Data-Plane Traffic

Nexus 9000 Traffic Forwarding

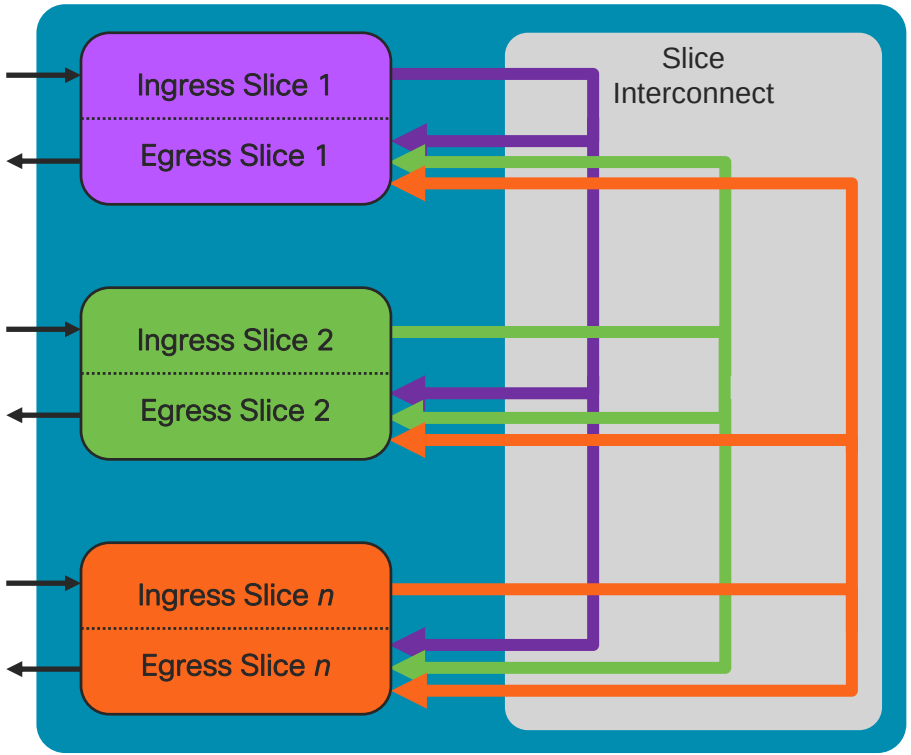
SoC and Slice

- SoC has one or more slices, and a slice interconnect if more than one slice

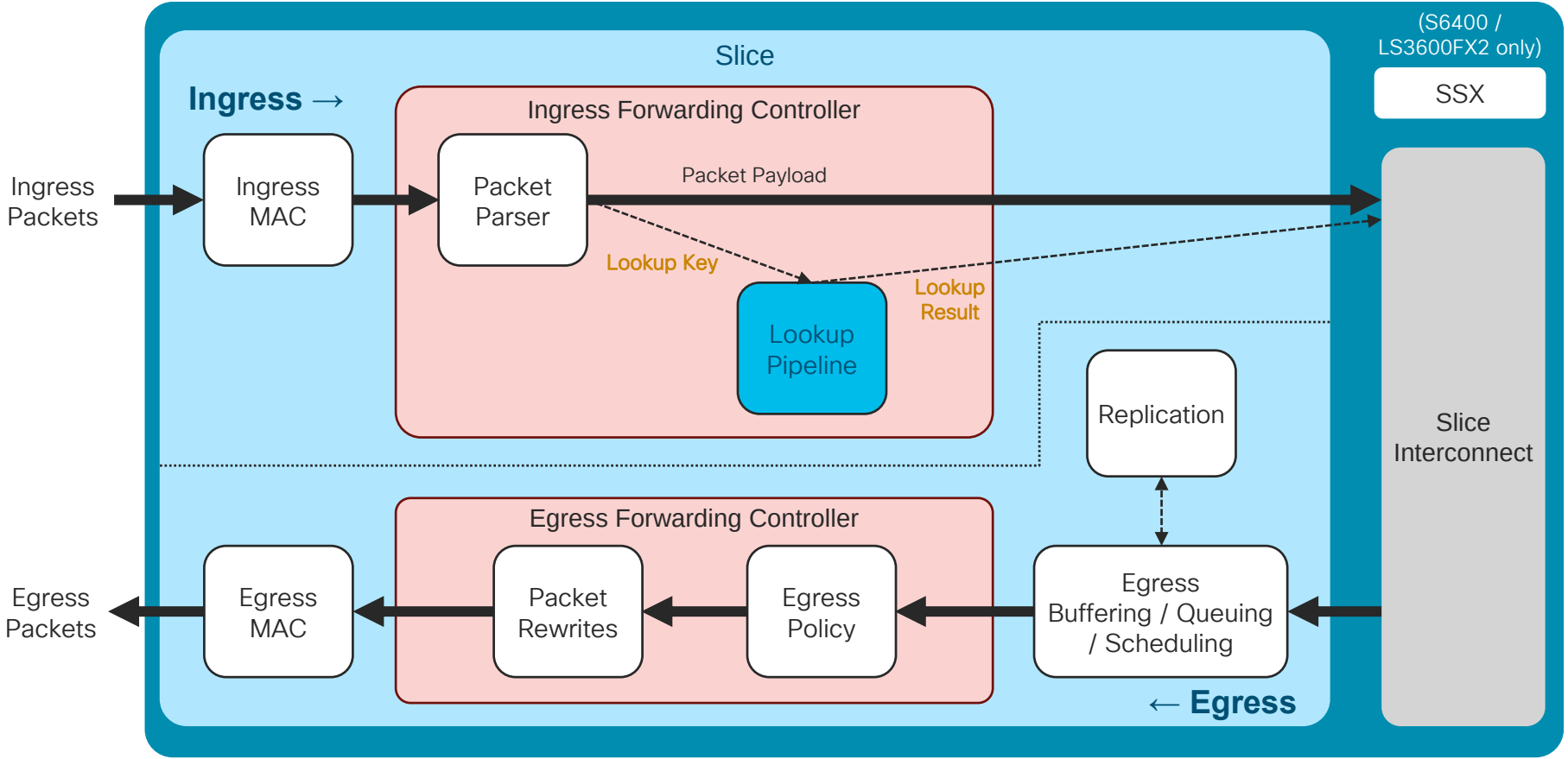


LS3600FX2 – 36x 100G

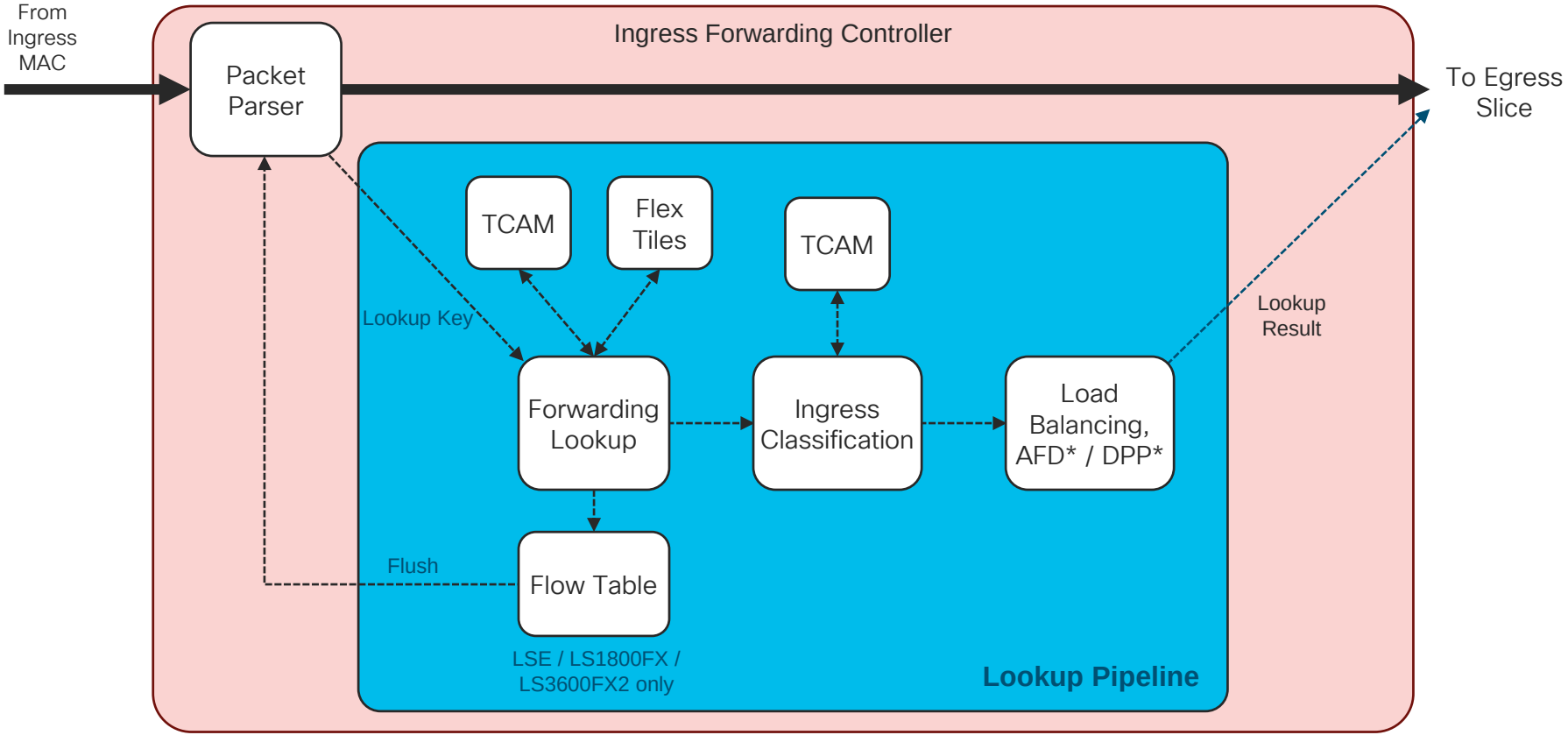
- Slice
  - a self-contained forwarding complex controlling subset of ports on single ASIC
  - Separated into Ingress and Egress functions
  - Ingress of each slice connected to egress of all slices
  - Slice interconnect provides non-blocking any-to-any interconnection between slices



Slice Forwarding Path



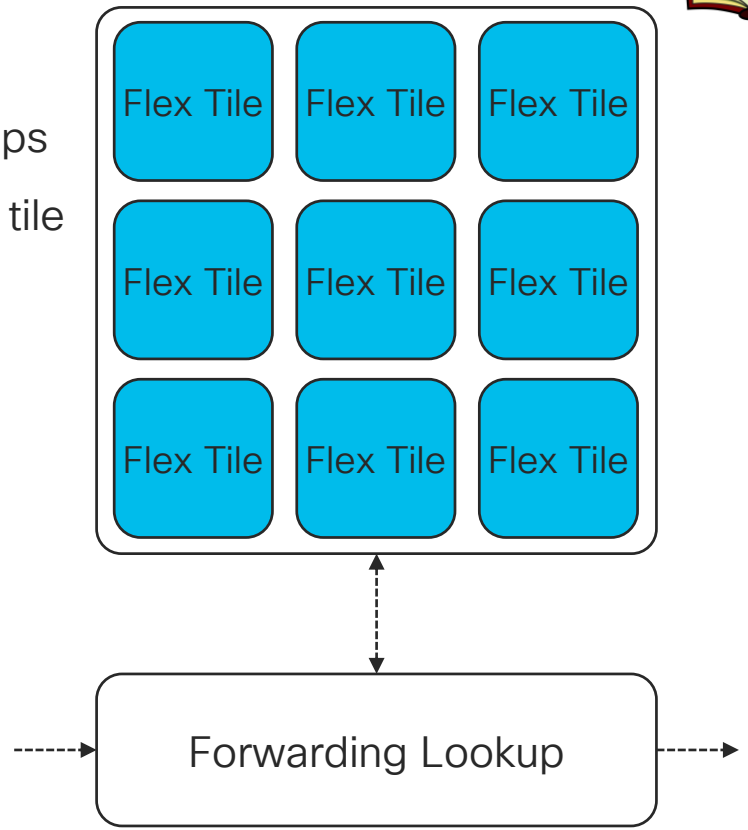
Ingress Lookup Pipeline



*AFD = Approx. Fair Dropping *DPP = Dynamic Packet Prioritization

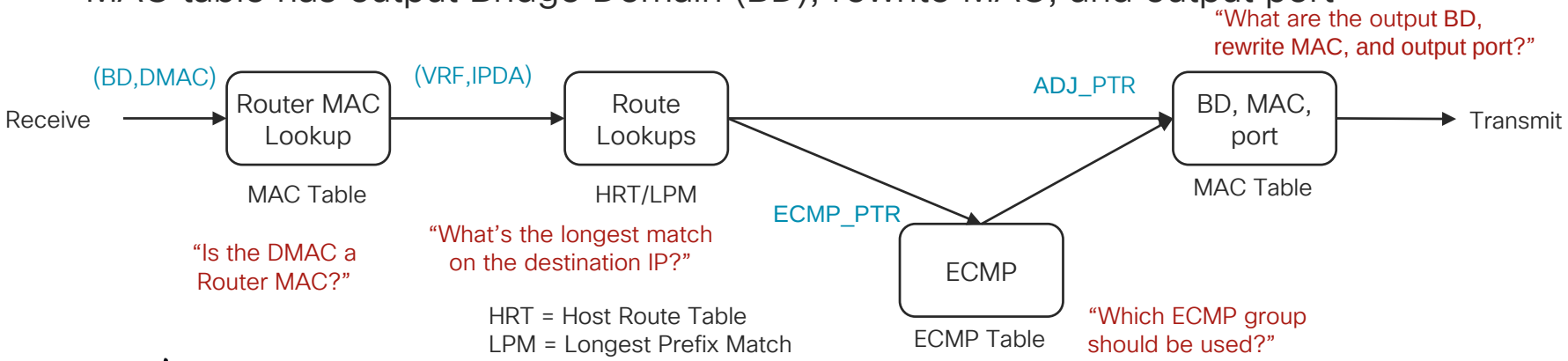
Flexible Forwarding Tiles

- Provide fungible pool of table entries for lookups
- Number of tiles and number of entries in each tile varies between ASICs
- Variety of functions, including:
  - IPv4/IPv6 unicast longest-prefix match (LPM)
  - IPv4/IPv6 unicast host-route table (HRT)
  - IPv4/IPv6 multicast (*,G) and (S,G)
  - MAC address/adjacency tables
  - ECMP tables



IP Unicast Forwarding

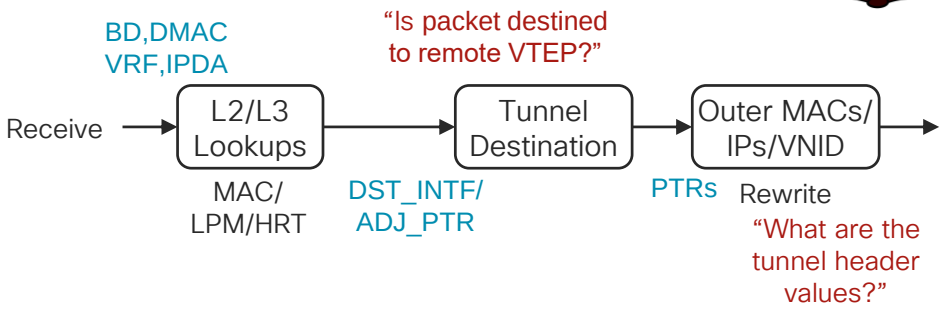
- Router MAC match triggers L3 lookup
- Hardware performs exact-match on VRF and longest-match on IP Destination Addr
- Lookup result returns either adjacency pointer (index into MAC table), or ECMP pointer
- MAC table has output Bridge Domain (BD), rewrite MAC, and output port



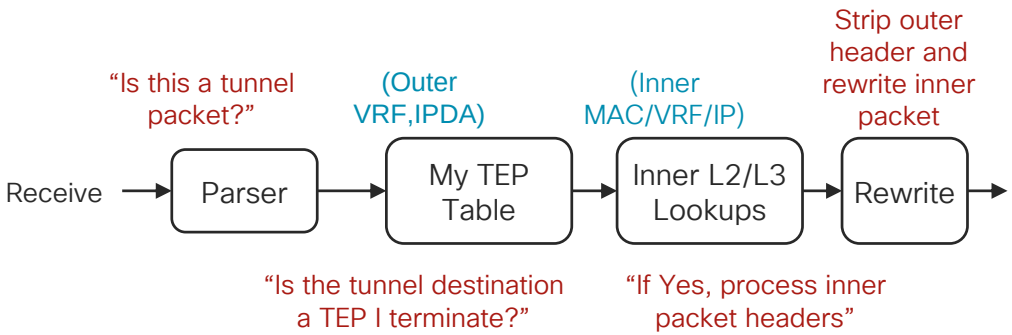
VXLAN Forwarding

- VXLAN and other tunnel encapsulation/ decapsulation performed in single pass
- Encapsulation
  - L2/L3 lookup drives tunnel destination
  - Rewrite block drives outer header fields (tunnel MACs/IPs/VNID, etc.)
- Decapsulation
  - Packet parser determines whether and what type of tunnel packet
  - Forwarding pipeline determines whether tunnel is terminated locally, drives inner lookups

Encapsulation

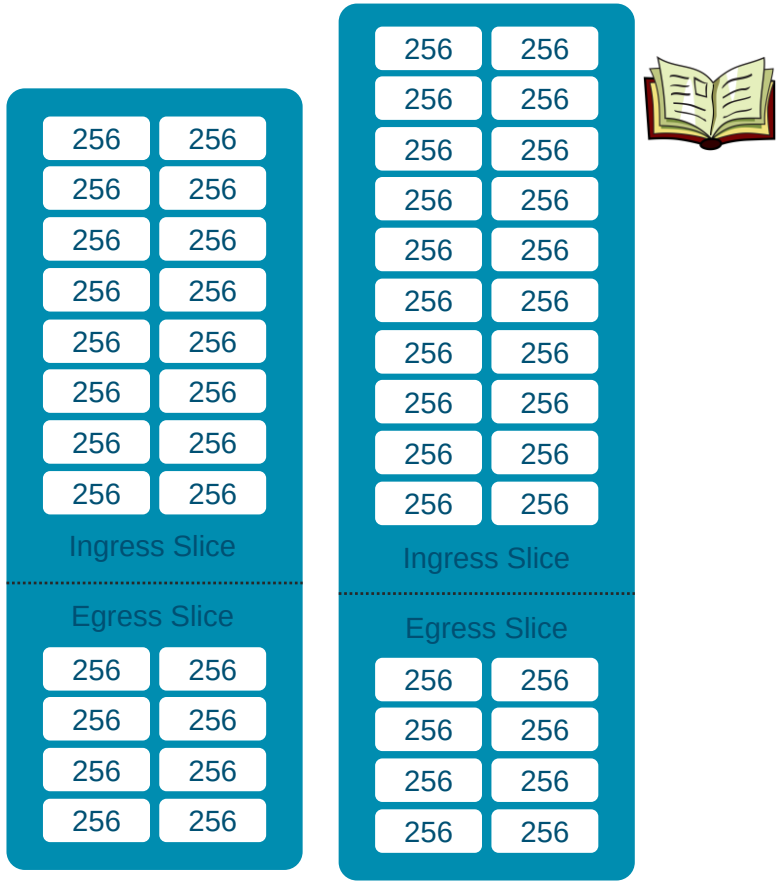


Decapsulation



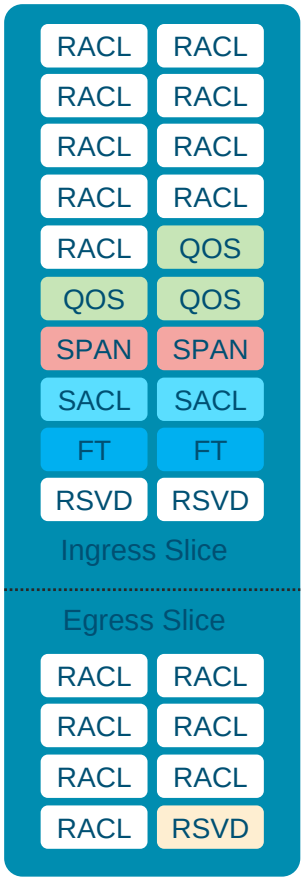
Classification TCAM

- Dedicated TCAM for packet classification
- Capacity varies depending on platform
- Leveraged by variety of features:
  - RACL / VACL / PACL
  - L2/L3 QOS
  - SPAN / SPAN ACL
  - NAT
  - COPP
  - Flow table filter (LS1800FX/ LS3600FX2)



TCAM Region Resizing

- Default carving allocates 100% of TCAM and enables:
  - Ingress / Egress RACL
  - Ingress QOS
  - SPAN, SPAN ACLs
  - Flow table filter (LS1800FX / LS3600FX2 only)
  - Reserved regions
- Based on features required, user can resize TCAM regions to adjust scale
  - To increase size of a region, some other region must be sized smaller
- Region sizes defined at initialization – changing allocation requires system reboot
  - Configure all regions to desired size (“hardware access-list tcam region”), save configuration, and reload



Path of the Packet

Control-Plane Traffic - Setup

Nexus 9508 with 97XX modules

```
N9508-A# show mod
Mod Ports             Module-Type                         Model              Status
--- ----- ------------------------------------ ---------------------- --------
2     52   48x10/25G + 4x40/100G Ethernet Module N9K-X97160YC-EX        ok
3     32   32x100G Ethernet Module               N9K-X9732C-EX          ok
5     36   36x100G Ethernet Module               N9K-X9736C-EX          ok
22    0    8-slot Fabric Module                  N9K-C9508-FM-E         ok
23    0    8-slot Fabric Module                  N9K-C9508-FM-E         ok
24    0    8-slot Fabric Module                  N9K-C9508-FM-E         ok
26    0    8-slot Fabric Module                  N9K-C9508-FM-E         ok
27    0    Supervisor Module                     N9K-SUP-B              active *
28    0    Supervisor Module                     N9K-SUP-B              ha-standby
29    0    System Controller                     N9K-SC-A               active
30    0    System Controller                     N9K-SC-A               standby
```

Modules

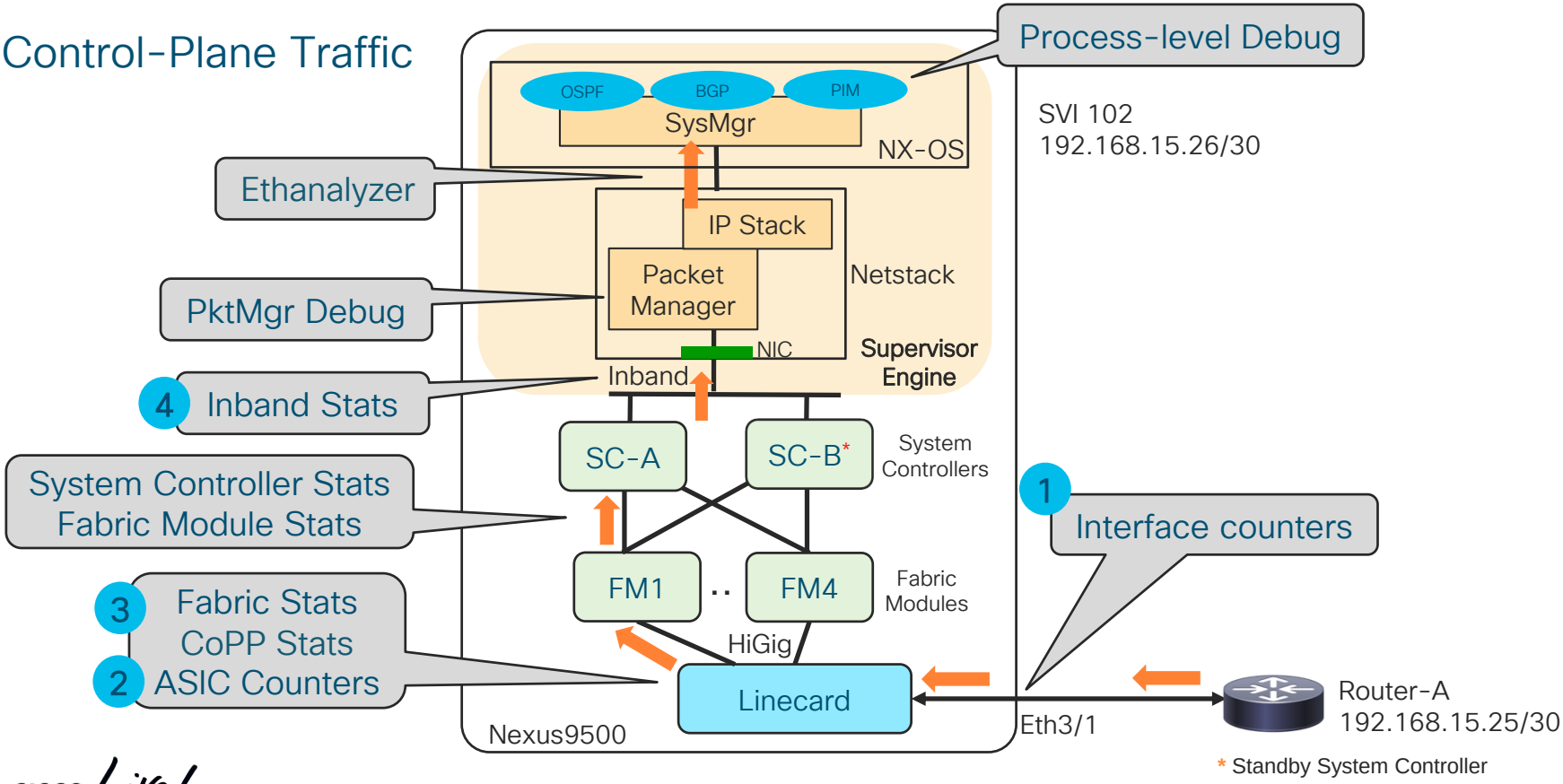
Fabric Modules

Supervisor Engines

System Controllers

Path of the Packet

Control-Plane Traffic



Path of the Packet

Control-Plane Traffic: Interface Counters

1

```
N9508-A# show interface e3/1
Ethernet3/1 is up
admin state is up, Dedicated Interface
<snip>
 RX
    0 unicast packets  11 multicast packets  2 broadcast packets
    13 input packets  2294 bytes
    0 jumbo packets  0 storm suppression bytes
    0 runts  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
 TX
    0 unicast packets  3 multicast packets  0 broadcast packets
    3 output packets  702 bytes
    0 jumbo packets
    0 output error  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 Tx pause
```

Do you remember the slide with a **treasure chest**?

Path of the Packet

Control-Plane Traffic: ASIC Counters

```
N9508-A# show system internal interface ii3/1/1 counters
Internal Port Statistics for Slot: ii3/1/1  If_Index 0x4a100000
================================================================
<snip>
Mac Pktflow:
  Rx Counters:
    <snip>
  Tx Counters:
    <snip>
Mac Control:
    Rx Pause:        0x0000000000000000/0
    Tx Pause:        0x0000000000000000/0
    Reset:           0x0000000000000000/0
Mac Errors:
    Undersize:       0x0000000000000000/0
    Runt:            0x0000000000000000/0
    Crc:             0x0000000000000000/0
    Input Errors:    0x0000000000000001/1
    <…continued…>
```

```
<…continued…>
    In Discard:      0x0000000000000000/0
    Giants:          0x0000000000000001/1
    Output Errors:   0x0000000000000000/0
    Output Discard:  0x0000000000000000/0
    Bad Proto:       0x0000000000000000/0
    Collision:       0x0000000000000000/0
    Late Collision:  0x0000000000000000/0
    No Carrier:      0x0000000000000000/0
```

Path of the Packet

Control-Plane Traffic: **ASIC Counters** (for front-panel ports)

2

```
N9508-A# show hardware internal interface asic counters mod 3
Important Counters/Drops
-------------- ---------------------------------------------------------------------------
Interface Drop Reasons for the Interface, See below output for detail if any
--------- ---------------------------------------------------------------------------

          |9|9|9|9|9|9|8|8|8|8|8|8|8|8|8|7|7|7|7|7|7|7|7|7|6|6|6|6|6|6 ..... 0|0|0|0|0|0
          |5|4|3|2|1|0|9|8|7|6|5|4|3|2|1|0|9|8|7|6|5|4|3|2|1|0|9|8|7|6|5|4|3 ..... 6|5|4|3|2|1
Eth3/1    |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|X|.|.|.|. ..... .|.|.|.|.|.
Eth3/2    |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.
Eth3/3    |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.
Eth3/4    |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.
<snip>
Eth3/32   |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.

Drop Conditions
-------------- ------------------------------ ------

        67 :    TAHOE Ingress DROP_ACL_DROP
```

Indicates traffic dropped because of ACL!!

Do “clear hardware internal interface-all asic counters mod <mod#>” to clear the conditions

Path of the Packet

Tahoe ASIC Counters

#	Description
1	DROP_PARSE_ERR
2	DROP_EOF_ERR
3	DROP_OUTER_IDS_G0
4	DROP_OUTER_IDS_G1
5	DROP_OUTER_IDS_G2
6	DROP_OUTER_IDS_G3
7	DROP_OUTER_IDS_G4
8	DROP_OUTER_IDS_G5
9	DROP_OUTER_IDS_G6
10	DROP_OUTER_IDS_G7
11	DROP_OUTER_XLATE_MISS
12	DROP_INFRA_ENCAP_SRC_TEP_MISS
13	DROP_INFRA_ENCAP_TYPE_MISMATCH
14	DROP_UC_TENANT_MYTEP_ROUTE_MISS
15	DROP_TENANT_MYTEP_BRIDGE_MISS
16	DROP_ARP_ND_UCAST_MISS
17	DROP_QIQ_EXPECT_2_QTAGS
18	DROP_MC_DVIF_MISS
19	DROP_SHARD_OVERRIDE_VLAN_XLATE_MISS
20	DROP_FCF_CHECK_FAILED
21	DROP_TTL_EXPIRED
22	DROP_SECURITY_GROUP_DENY
23	DROP_LOOPBACK_OUTER_HEADER_MISMATCH
24	DROP_OVERLAYL2_OUTER_HEADER_MISMATCH
25	DROP_MC_IIC

#	Description
26	DROP_MC_GIPO_MISS
27	DROP_UC_HIT_NO_PATH
28	DROP_UNUSED
29	DROP_AC_SUP_DROP
30	DROP_AC_POL_DROP
31	DROP_AC_STORM_POL_DROP
32	DROP_FAST_CONV_LOOP_PREVENT
33	DROP_PP_BOUNCE_MYTEP_MISS
34	DROP_VLAN_MBR_INPUT
35	DROP_IEOR_PP_RETURN_PC_2_HG2_MISS
36	DROP_IEOR_UPLINK_UC_SAME_IF
37	DROP_IEOR_SPINE_PROXY_PC_2_HG2_MISS
38	DROP_VIF_MISS
39	DROP_UNEXPECTED_VFT
40	DROP_MISSING_VNTAG
41	DROP_VLAN_XLATE_MISS
42	DROP_RBID_FTAG_MISS
43	DROP_IP_MTU_CHECK_FAILURE
44	DROP_UC_RPF_FAILURE
45	DROP_MC_RPF_FAILURE
46	DROP_L3_BINDING_FAILURE
47	DROP_IP_UNICAST_FIB_MISS
48	DROP_FIB_SA
49	DROP_FIB_DA
50	DROP_NSH_NOT_ALLOWED

Path of the Packet

Tahoe ASIC Counters

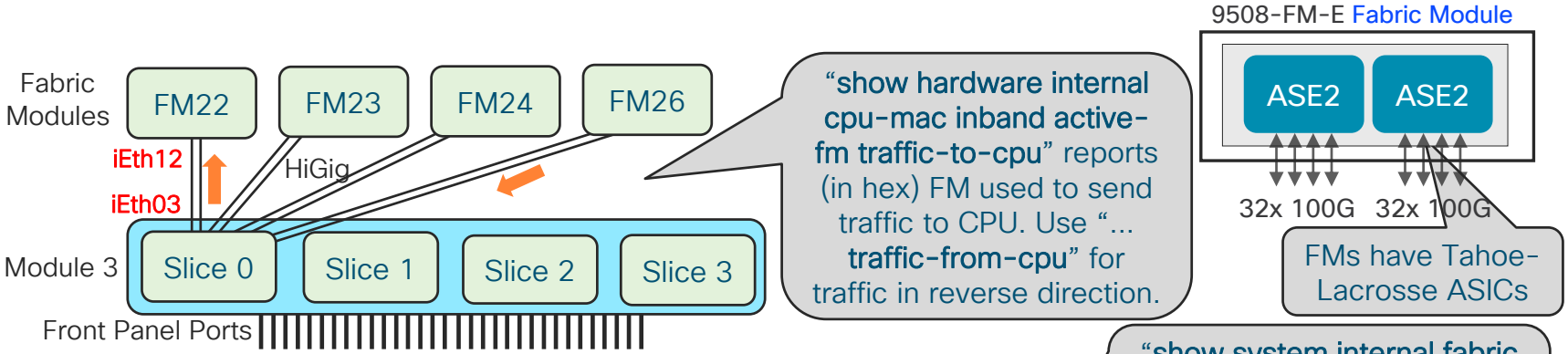
#	Description
51	DROP_SRC_VLAN_MBR
52	DROP_NSH_SRC_SW_CHK_FAILED
53	DROP_L2MP_IIC_FAILED
54	DROP_L2MP_ON_CE_BD
55	DROP_L2MP_ENCAP_FROM_EDGE
56	DROP_L2MP_NOENCAP_FROM_CORE
57	DROP_OUTER_TTL_EXPIRED
58	DROP_INCORRECT_VNTAG_TYPE
59	DROP_L2MP_FTAG_COMP_MISS
60	DROP_IPV6_UC_LINK_LOCAL_CROSS_BD
61	DROP_IPV6_MC_SA_LOCAL_DA_GLOBAL_SVI
62	DROP_IPV6_MC_SA_LOCAL_DA_GLOBAL_L3IF
63	DROP_ROUTING_DISABLED
64	DROP_FC_LOOKUP_MISS
65	DROP_NO_SGT_FROM_CORE
66	DROP_IP_SELF_FWD_FAILURE
67	DROP_ACL_DROP
68	DROP_SMAC_MISS
69	DROP_SECURE_MAC_MOVE
70	DROP_NON_SECURE_MAC
71	DROP_L2_BINDING_FAILURE
72	DROP_INNER_IDS_G0
73	DROP_INNER_IDS_G1

#	Description
74	DROP_INNER_IDS_G2
75	DROP_INNER_IDS_G3
76	DROP_INNER_IDS_G4
77	DROP_INNER_IDS_G5
78	DROP_INNER_IDS_G6
79	DROP_INNER_IDS_G7
80	DROP_INFRA_ENCAP_SRC_TEP_DROP
81	DROP_SPLIT_HORIZON_CHECK
82	DROP_MC_FIB_MISS
83	DROP_MC_L2_MISS
84	DROP_UC_DF_CHECK_FAILURE
85	DROP_UC_PC_CFG_TABLE_DROP
86	DROP_ILLEGAL_EXPL_NULL
87	DROP_MPLS_LOOKUP_MISS
88	DROP_OUTER_CBL_CHECK
89	DROP_NULL_SHARD_WITH_E_BIT_SET
90	DROP_LB_DROP
91	DROP_NAT_FRAGMENT
92	DROP_ILLEGAL_DCE_PKT
93	DROP_DCI_VNID_XLATE_MISS
94	DROP_DCI_SCLASS_XLATE_MISS
95	DROP_DCI_2ND_UC_TRANSIT

Path of the Packet

Control-Plane Traffic: FM and Linecards Connectivity

ASE – Application Spine Engine



"show system internal fabric connectivity module 22" reports connectivity from Fabric Modules' perspective.

```
N9508-A# show system internal fabric connectivity mod 3
Internal Link-info Linecard slot:3
------------------------------------------------------------------
 LC-Slot  LC-Unit  LC-iEthLink    MUX   FM-Slot   FM-Unit   FM-iEthLink
------------------------------------------------------------------
    3        0      iEth03         -       22        0       iEth12
    3        0      iEth05         -       22        1       iEth44
<snip>
    3        1      iEth11         -       22        0       iEth11
    3        1      iEth13         -       22        1       iEth43
<snip>
```

slice #

Path of the Packet

Control-Plane Traffic: **Linecards Drops** (on **HiGig** links to Fabric Modules)

3

Good news!! No drop in the module on the interface connected to fabric module

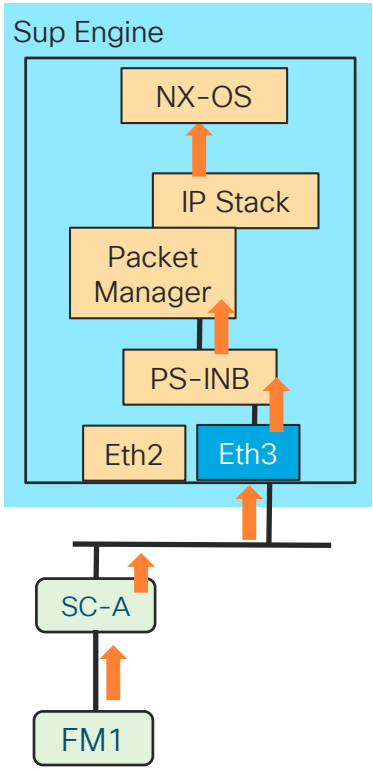
```
N9508-A# show hardware internal fabric interface asic counters mod 3
Important Counters/Drops
--------------- ------------------------------------------------------------------------
Interface Drop Reasons for the Interface, See below output for detail if any
--------- ------------------------------------------------------------------------------
          |9|9|9|9|9|9|8|8|8|8|8|8|8|8|8|7|7|7|7|7|7|7|7|7|6|6|6|6|6|6|6 ..... 0|0|0|0|0|0
          |5|4|3|2|1|0|9|8|7|6|5|4|3|2|1|0|9|8|7|6|5|4|3|2|1|0|9|8|7|6|5|4|3 ..... 6|5|5|3|2|1
iEth1     |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.
iEth2     |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.
iEth3     |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.
iEth4     |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.
<snip>
iEth32    |.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|. ..... .|.|.|.|.|.

Drop Conditions
--------------- ------------------------------------------------------------------------
```

Path of the Packet

Control-Plane Traffic: Inband Counters

4

```
N9508-A# show hardware internal cpu-mac inband counters
eth2      Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          BROADCAST MULTICAST  MTU:9400  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
eth3      Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          UP BROADCAST RUNNING MULTICAST  MTU:9400  Metric:1
          RX packets:8484226 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4523271 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:860671333 (820.8 MiB)  TX bytes:493276319 (470.4 MiB)
ps-inb    Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          UP BROADCAST RUNNING MULTICAST  MTU:9400  Metric:1
          RX packets:14327 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14312 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:38890552 (37.0 MiB)  TX bytes:37871460 (36.1 MiB)
```



Path of the Packet

Control-Plane Traffic: **Inband Statistics**

4

```
N9508-A# show hardware internal cpu-mac inband stats
<snip>
eth3 stats:
RMON counters                    Rx                    Tx
----------------------+------------+--------------------
total packets                8406058               4481386
<snip>
65-127 bytes packets         8391840               4470748
<snip>
broadcast packets                 15                561531
multicast packets                  0                     0
<snip>
Error counters
--------------------------------+--
CRC errors ..................... 0
Alignment errors ............... 0
Symbol errors .................. 0
Sequence errors ................ 0
RX errors ...................... 0
<… continued …>
```

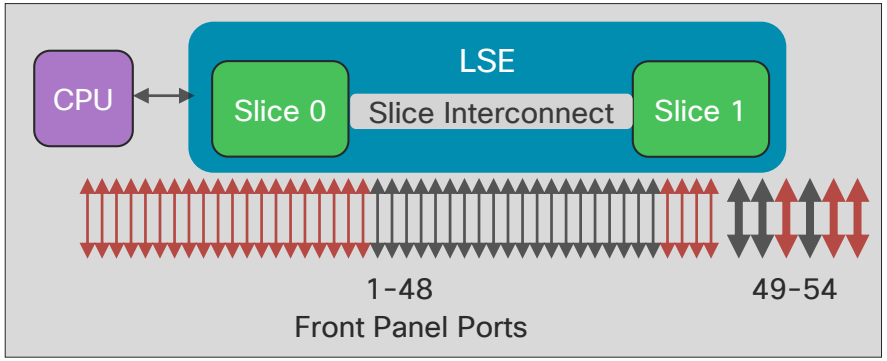
```
<… continued …>
Missed packets (FIFO overflow)   0
Single collisions ..............  0
Excessive collisions ...........  0
Multiple collisions ............  0
Late collisions ................  0
Collisions .....................  0
Defers .........................  0
Tx no CRS  .....................  0
Carrier extension errors .......  0
Rx length errors ...............  0
FC Rx unsupported ..............  0
Rx no buffers ..................  0
Rx undersize ...................  0
Rx fragments ...................  0
Rx oversize ....................  0
Rx jabbers .....................  0
Rx management packets dropped ..  0
Tx TCP segmentation context ....  0
Tx TCP segmentation context fail  0
Rate statistics
------------------------------+---------
Rx packet rate (current/peak)   160 / 1254 pps
Tx packet rate (current/peak)   112 / 889 pps
<snip>
```

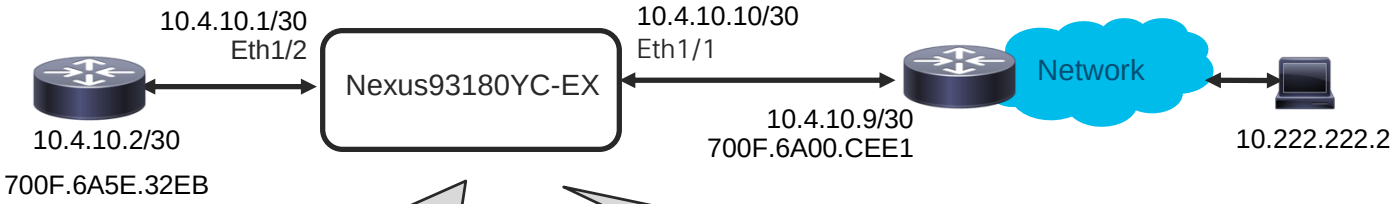
Good health-check.
Set a baseline!!

Path of the Packet

Data-Plane: **L3 Flow**

Troubleshooting communication failure for traffic flowing through Nexus 9300





1. Check Forwarding Information Base (FIB) in Software and Hardware
2. Check Routing Table in the ASIC
3. Check route programmed in the ASIC

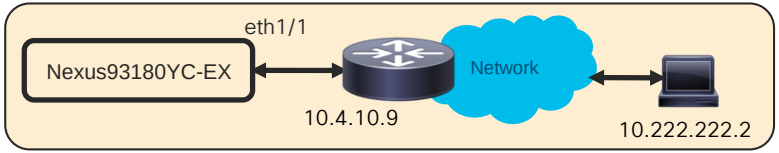
4. Check Adjacency in Software and Hardware
5. Check Adjacency Table in the ASIC
6. Check Adjacency programmed in the ASIC, and verify re-write mac and egress interface

Slice 0
Slice 1

Path of the Packet

Data-Plane: L3 Flow – Check SW/HW FIB

1



Check Forwarding Information Base (FIB) in Software

```
DC1-BGW1# show ip route 10.122.122.2
IP Route Table for VRF "default"
<snip>
10.222.222.2/32, ubest/mbest: 2/0
    *via 10.4.10.9, [20/0], 3d23h, bgp-65000, external, tag 65001
```

Check Forwarding Information Base (FIB) in Hardware

make sure the results are matching

```
DC1-BGW1# show forwarding route 10.222.222.2
slot  1
=======
IPv4 routes for table default/base
------------------+---------------+-----------------+------------+-----------------+
Prefix            | Next-hop      | Interface       | Labels     | Partial Install |
------------------+---------------+-----------------+------------+-----------------+
10.222.222.2/32     10.4.10.9       Ethernet1/1
```

Path of the Packet

Data-Plane: L3 Flow – Route Programmed in ASIC

2

Table #1 is for default VRF. To find table number for other VRFs, use "**show hardware internal tah L3 v4host**" command

Entry in Tahoe-Sugarbowl Routing Table

```
module-1# show hardware internal tah L3 10.222.222.2/32 table 1
DLeft location: 0x0
FP location   : 255/4/0xc
**EPE label
*Flags:
CC=Copy To CPU, SR=SA Sup Redirect,
DR=DA Sup Redirect, TD=Bypass TTL Dec,
DC=SA Direct Connect,DE=Route Default Entry,
LI=Route Learn Info,HR=Host as Route
 HW Loc     | Ip Entry      |   VRF | MPath | NumP  | Base/L2ptr |CC|SR|DR|TD|DC|DE|LI|HR|
 -----------|---------------|-------|-------|-------|------------|--|--|--|--|--|--|--|--|
 4/16       | 10.222.222.2  |     1 | No    | 0     | 0x40000002 |  |  |  |  |  |  |  |  |

 AdjId      | FP            | BD    | DMac               | DstIdx | DstIsPtr |
 -----------|---------------|-------|--------------------|--------|----------|
 0xd0004    | 13/0/0x4      | 4104  | 70:0f:6a:00:ce:e1  | 1      | No       |
```

the physical interface where the packet is going to be sent out. "**show hardware internal tah interface ethernet 1/1 | inc src_intf_num**" should report "**1**"

Adj entry ID

Location for the entry in the ASIC

BD/VLAN and Destination MAC for ***egress*** traffic

Do "**attach module <num>**" to attach to specific module.

Path of the Packet

Data-Plane: L3 Flow – Route Programmed in ASIC

3

Location for the entry in the ASIC

Route programmed in Tahoe-Sugarbowl ASIC

```
module-1# debug hardw intern sug dump asic 0 slice 0 table 4:tah_sug_fpa_fibtcam 16
field-per-l
asic instance is 0
asic slice is 0
tbl name is 4:tah_sug_fpa_fibtcam
Block base address: 0x04000000
1st table entry address: 0x04040100
ENTRY[16] = {
l3_tcam_entry_ip_type=0x00000000
l3_tcam_entry_v6_opt_vld=0x00000000
l3_tcam_entry_vrf_type=0x00000000
l3_tcam_entry_vrf=0x00000001
l3_tcam_entry_host_ip=0x0adede02
<snip>
l3_tcam_entry_mask_vrf=0x00ffc000
l3_tcam_entry_mask_host_ip=0x00000000
valid=0x00000001
 }
module-1#
```

Table number

VRF number

10.222.222.2 in hex

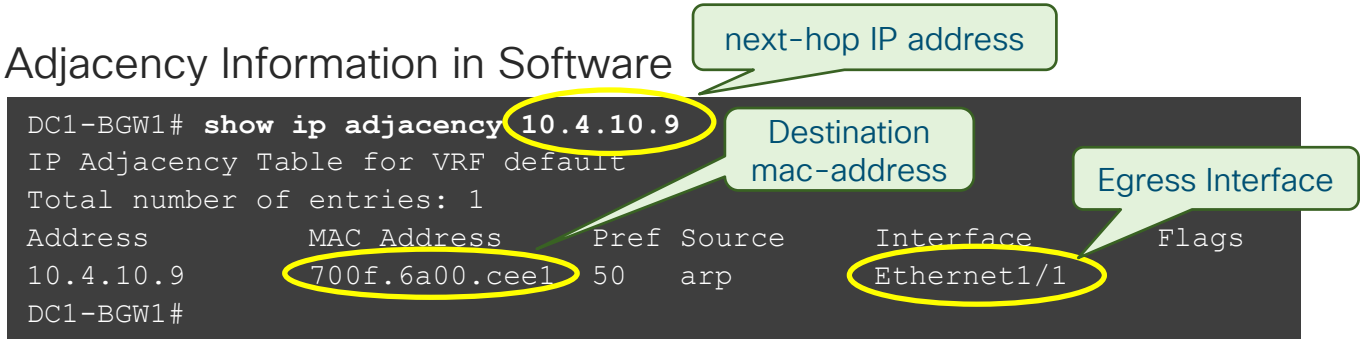
validity

Path of the Packet

Data-Plane: L3 Flow – Adjacency Programmed in ASIC

4

Adjacency Information in Software



Adjacency Information in Hardware

make sure the results are matching

```
DC1-BGW1# show forwarding adjacency 10.4.10.9
slot  1
=======
IPv4 adjacency information
next-hop          rewrite info    interface
--------------- --------------- -------------
10.4.10.9         700f.6a00.cee1  Ethernet1/1
DC1-BGW1#
```

Path of the Packet

Data-Plane: L3 Flow – Adjacency Programmed in ASIC

5 & 6

Entry in Tahoe-Sugarbowl Adjacency Table

```
module-1# show hardware internal tah L3 adjacency 0xd0004
 AdjId    | FP          | BD    | DMac               | DstIdx | DstIsPtr |
 --------|-------------|------|--------------------|--------|----------|
 0xd0004 | 13/0/0x4    | 4104  | 70:0f:6a:00:ce:e1  | 1      | No       |
```

From Step #2

Adjacency Entry programmed in the Tahoe-Sugarbowl ASIC

```
module-1# debug hardware internal sug dump asic 0 slice 0 fp 13 table
0:tah_sug_fpx_fptile 0x4 field-per-line | grep l2_entry_mac
tile_entry_l2_entry_mac_entry_mackey_vld=0x00000001
tile_entry_l2_entry_mac_entry_mackey_fid_type=0x00000000
tile_entry_l2_entry_mac_entry_mackey_fid_vld=0x00000001
tile_entry_l2_entry_mac_entry_mackey_fid=0x00001008
tile_entry_l2_entry_mac_entry_mackey_mac=0x0000700f:0x6a00cee1
tile_entry_l2_entry_mac_entry_entry_type=0x00000000
tile_entry_l2_entry_mac_entry_intf=0x00000001
tile_entry_l2_entry_mac_entry_learn_info=0x00000002
```

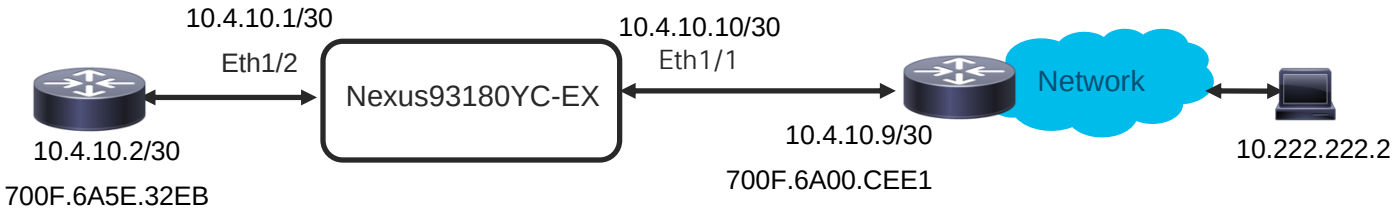
Re-write destination mac-addr

Interface: 0x1 = Eth1/1.
"show hardware internal tah interface ethernet 1/1 | inc src_intf_num" should report "1"

Path of the Packet

Data-Plane: **L3 Flow**

Troubleshooting communication failure for an L3 Flow



What we just did?

✅ Verified Routing and Adjacency Tables in the Software (steps 1 and 4)

✅ Verified Routing and Adjacency Tables in the ASIC (steps 2 and 5)

✅ Verified Routing and Adjacency entries programmed in the ASIC (steps 3 and 6)

Do you remember **Virtual TAC Assistant** and its benefits?

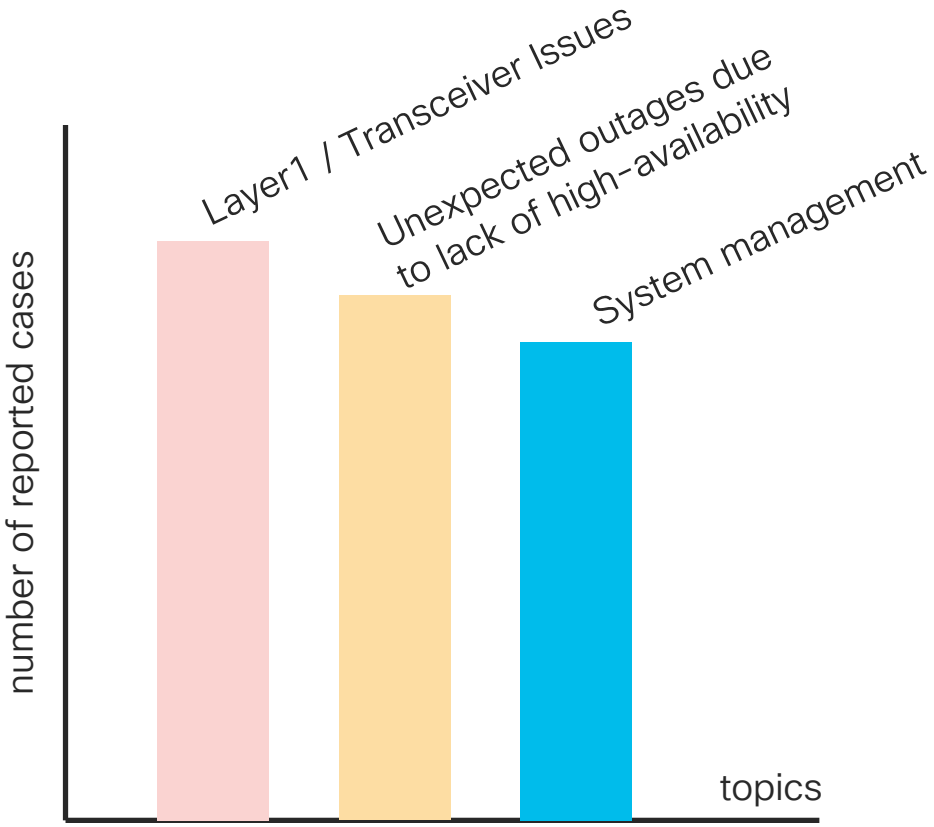
Best Practices and Recommendations

Agenda

- Introduction
- Monitor and Health-Check
- Troubleshooting Tools
- Troubleshooting Traffic Forwarding
- Best Practices and Recommendations
- Summary and Take-Aways

Based on true data!!

Customer-reported Problems Trends



Best Practices and Recommendations

Layer 1 and Transceivers

- Connect the cable/media at both ends, insert the transceivers completely and through following commands verify speed, duplex, capabilities, supported modes and DOM values.

  show interface eth x/y transceiver details

  show interface eth x/y capabilities

  show interface brief - check for the interface tuple display and others

  show interface eth x/y status

- Enable auto-negotiation at both ends. Yes, we need it!
- Check transparent device or circuit in the middle, if any
- Have you checked Transceiver compatibility? Review Transceiver Compatibility Matrix at https://tmgmatrix.cisco.com/
- Internal event-history commands can be helpful to determine which device have initiated link-down first.

Best Practices and Recommendations

Redundancy and High-Availability

- Do you have port-channel members distributed?
  - Have peer-link, peer-keepalive, vPC members distributed across modules and chassis
- Are you taking advantage of… ?
  - **vPC peer-gateway** – to avoid traffic looping over peer-link, and for optimized forwarding
  - **vPC peer-switch** – to build single L2 logical domain from spanning-tree perspective
  - **vPC L3 peer-router** – letting routing adjacency build on vPC VLANs
  - **vPC auto-recovery** – to avoid dual-active condition (on by default)
- Are you taking advantage of… ?
  - Graceful Insertion and Removal (GIR)
- Do you have enough room to handle transient bursty traffic?
- Do you have enough resources free for new feature(s)? Refer latest Scalability Guide

Best Practices and Recommendations

System Management – Choose right NX-OS version

Nexus 9000 Recommended Software bulletin at Cisco.com

General Recommendation for New and Existing Deployments:

Platform	Recommended Release
Nexus 9000	7.0(3)I7(7)

Earlier Recommendations and Releases:

Type	Release Number
Current Long-lived Release	7.0(3)I7(x)
Upcoming Long-lived Release	9.3(x)*
Previous Long-lived Release and Recommended Software	7.0(3)I4(x) / 7.0(3)I4(8b)
Short-lived Releases	7.0(3)I1(x), 7.0(3)I3(x), 7.0(3)I5(x), 7.0(3)I6(x), and 9.2(x)*

* If 9.2(x) or 9.3(x) is needed to deploy new hardware or features, use the latest version available on CCO

Summary
&
Take-Aways

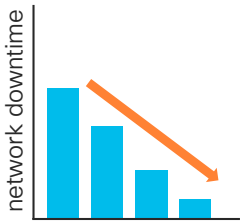
Summary

What you need to do?	How its going to help you?
monitor health and resource usage	proactively identify bottlenecks and hotspots
get familiar with built-in tools	attain better visibility and localize issues
know path-of-the-packet in a device and relevant troubleshooting commands	get data before theorize and reduce downtime
implement best practices	achieve higher network availability

Higher gate density and bandwidth achievements are transforming hardware architecture and functions consolidation. Nexus 9000 is at the core of these transformation, and flexible to fit datacenter design of your choice.
Nexus 9000 is the platform of possibilities!!

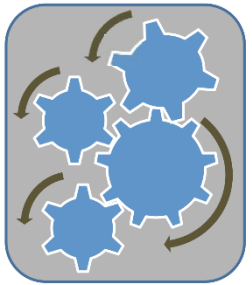
Take-Aways

Nexus 9000 have RICH SET OF CLIs, FEATURES and TOOLS that are developed keeping all of you in mind.



Closely monitoring devices' health, and knowing troubleshooting techniques significantly reduce network downtime

Wealth of knowledge shared in this session ENABLES AND EMPOWERS EACH ONE OF YOU to achieve the goals of your organization.



References and Useful Links

- Nexus 9000 Configuration Guide
- Cisco Nexus 9000 Series NX-OS Troubleshooting Guide
- Nexus 9000 Scalability Guide NX-OS version 9.3(3)
- Transceiver Compatabilty Matrix
- Nexus 9000 Recommended Software Bulletin
- Nexus 9000 Programmability Guide
- Open NX-OS Programmabiity – User Guide
- Cisco Nexus 3000/9000 NX-API REST SDK User Guide and API Reference
- Nexus 3000/9000 Series Telemetry Sources
- Nexus 9000 GitHub Repository

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

cisco Live!

Continue your education

Demos in the Cisco campus

Walk-in Labs

Meet the engineer 1:1 meetings

Related sessions

Thank you

cisco Live!

You make **possible**