Troubleshooting Cisco Catalyst 3850 and 3650 Series Switches

Wafo Luc Tengueu, Sr. Mgr. Systems Engineering Sales
BRKCRS-3146
Session Abstract

• Cisco is bringing together the best of wired and wireless networking into “One Network” with Converged Access on the Catalyst 3850 and 3650 Switches

• In this session, learn about the capabilities of the 3850 and 3650 switches and troubleshoot common issues seen on the 3850 and 3650 running the IOS-XE Operating System. Learn about the switch architecture and troubleshooting hardware, RTU Licensing, Memory and CPU utilization, Stacking, High Availability, Forwarding features, QOS and a glimpse at our revamped IOS XE starting 16.1.1

• Target audience – network engineers and technicians who work with the Cisco Catalyst 3850/3650 Series
Your Instructor today …
Wafo Luc Tengueu
Sr. Mgr. Systems Engineering Sales, Enterprise West Territory

I’m with the West Territory of the US Enterprise Sales organization. My current focus is the successful selling and deployment of end to end Cisco technologies across Enterprise Networks. My Background is Software Engineering and I have been with Cisco for 16 Years. Prior to my current role, I have worked on a wide spectrum of technologies, platforms and held different positions ranging from Software Engineer, Technical Leader (Cat6500, 7200, Asr1000, ISRXXX) to Sr. Mgr. of Software Engineering.

My passion for our customer success has lead me to lead a team of System engineers in the San Francisco Bay Area.
Glossary

3850/3650 Switch

Active Switch

Standby Switch

FED – Forwarding Engine Driver

PDS – Packet Delivery Service

ASIC

Reference slide that may not be presented in the session

WCM – Wireless Controller Module

UADP – Unified Access Data Plane

3x50 – 3650 or 3850 Switch
Agenda

- Product Overview
- Image Management
- Troubleshooting Memory and CPU Resources
- Troubleshooting Stacking & High Availability
- Troubleshooting Hardware Forwarding
- Qos Implementation and Troubleshooting
- A glimpse at the future, IOS XE 16.X
- Summary
In this section, you will learn about ...

• Overview of the 3X50
• IOS-XE architecture
• Supported uplink modules
Catalyst 3850 Switch

- **Wireless CAPWAP Termination**
- **Up to 100 APs and 40G per switch**
- **Integrated Controller**
- **480 Gbps Stacking Bandwidth**
- **Up to 2000 Clients per Stack**
- **Full POE+**
- **Granular QoS/Flexible NetFlow**
- **Line Rate on All Ports**
- **80 Gbps Uplink Bandwidth**
- **FRU Fans, Power Supplies**
- **Stackpower**
- **SGT/SGACL**

*Built on Cisco’s Innovative “UADP” ASIC*
Catalyst 3650 Switch

- New Front-End Power Supplies
- Modular 160 Gbps 9 member Stack
- Up to 50 Aps/1000 clients per stack, and 40G per switch
- Wireless CAPWAP Termination
- Fixed 1G/10G Uplinks
- 40 Gbps Uplink Bandwidth
- Full POE+
- FRU Fans
- Up to 1000 Clients per Stack
- SGT/SGACL
- Granular QoS/Flexible NetFlow
- Line Rate on All Ports

The foundation for full wired and wireless convergence on a single platform.
Why IOS XE?

- Modern IOS to enable multi-core CPU
- Easy customer migration
- While maintaining IOS functionality and look and feel
- Allow hosted applications like Wireshark

IOS 12.2(52)SE

- Features Components
- Common Infrastructure / HA
- Management Interface
- Module Drivers
- Kernel

IOS XE 3.6.4.E

- IOSd
- Hosted Apps
- WCM

3.6.x Features

- Vrf aware for Ipv6 routing protocol
- Object Tracking for IpV6
- Bi-directional SXP support (Trustsec)
- Ipv6 Mcast routing
IOS XE Software Internals Overview

- Kernel
  - Driver Stubs
  - Generic Driver

- IOSd RP/LC
  - IOSd RP/LC Features PD
    - Platform Drivers
    - UADP ASIC Drivers
    - Low Level APIs
  - Forwarding Engine Driver
  - Forwarding & Feature Mgr (FFM)

- HA
- Consolidated Logging
- Internal IPC
- Licensing Services
- Comet Services
- Platform Manager
- System Manager

- IP Internal IPC
  - Packet Delivery Service
  - Libraries/Utilities Services

- Availability Framework
  - System Manager
  - Comet Services
  - Licensing Services
  - Consolidated Logging
  - HA

- IOS XE Software Internals Overview
  - Kernel
  - Driver Stubs
  - Generic Driver
  - Generic Driver
Recommended Release IOS-XE 3.6.4E

Why run the recommended release?

- Many critical fixes in recommended release 3.6.4E
- Hardening effort around identity (802.1X, AAA Accounting, DHCP, ISE)
- IPDT (IP Device Tracking) turned off by default
- Several new features

# Network Modules on the 3850

## Aggregates Bandwidth toward Distribution/Core

<table>
<thead>
<tr>
<th>Type of interface</th>
<th>4x1G</th>
<th>2x1G 2x10G</th>
<th>4x10G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1G</td>
<td>-</td>
<td>-</td>
<td>GigabitEthernet 1 - 4</td>
</tr>
<tr>
<td>10G</td>
<td>-</td>
<td>GigabitEthernet 1 - 2 &amp; TenGigabitEthernet 3 - 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TenGigabitEthernet 1 - 4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of interface</th>
<th>8x10G</th>
<th>2x40G</th>
</tr>
</thead>
<tbody>
<tr>
<td>10G</td>
<td>TenGigabitEthernet 1 - 8</td>
<td>-</td>
</tr>
<tr>
<td>40G</td>
<td>-</td>
<td>FortyGigabitEthernet 1 – 2</td>
</tr>
</tbody>
</table>

### WS-C3850-NM-4-1G
- 4 x 1G
- SFP
- Supported on WS-C3850-24 & WS-C3850-48 Port

### WS-C3850-NM-2-10G
- 4 x 1G OR 2 x 10G OR 2 x 1G + 1 x 10G
- SFP & SFP+
- Supported on WS-C3850-24 & WS-C3850-48 Port

### WS-C3850-NM-4-10G
- Auto-sensing – All Combinations
- SFP & SFP+
- Supported on WS-C3850-48 only

### WS-C3850-NM-8-10G
- 8 x 10G
- SFP & SFP+
- Supported on WS-C3850-24XU & WS-C3850-12X48U

### WS-C3850-NM-2-40G
- QSFP
- Supported on WS-C3850-24XU & WS-C3850-12X48U
Catalyst 3850 Multigigabit Ethernet Switches

Why is it Needed?

<table>
<thead>
<tr>
<th># mgig ports</th>
<th>Advanced port capabilities</th>
<th>New high-speed uplinks</th>
</tr>
</thead>
<tbody>
<tr>
<td>3850 48-port</td>
<td>12 mGig ports</td>
<td>New 2x40G and 8x10G Uplink support</td>
</tr>
<tr>
<td>3850 24-port</td>
<td>24 mGig ports</td>
<td>New 2x40G and 8x10G Uplink support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cable Type</th>
<th>1G</th>
<th>2.5G</th>
<th>5G</th>
<th>10G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat5e</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>NOT SUPPORTED</td>
</tr>
<tr>
<td>Cat6</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Cat6a</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

- 802.11ac-2 (3.5Gbps), maintain switch to AP reach at higher speeds (future proof for higher speeds)
- Infrastructure investment protection
- Auto-negotiation of cable type of speeds supported
- **Brownfield** deployments can leverage existing Cat5e extending ROI and support mGig at 2.5G and 5G speeds at a distance of 100m
- **Greenfield** deployments with Cat6a will support 10G but can also now support mGig at 2.5G and 5G speeds at a distance of 100m
Express Mode Button known issue

- Several customers complained configuration was erased
- Certain types of Ethernet cables have protective boots that extend too far forward, issue when this type of cable is inserted on Port 1
- CSCuj17317: Certain snagless cables may press on the mode button causing reload
  - changed the old behavior, customer must save config before it can take effect
- Customer can also disable express setup
  - `3850(config)# no setup express`
- Field Notice FN63697
Agenda

- Product Overview
- Image Management
- Troubleshooting Memory and CPU Resources
- Troubleshooting Stacking & High Availability
- Troubleshooting Hardware Forwarding
- Qos Implementation and Troubleshooting
- A glimpse at the future, IOS XE 16.X
- Summary
In this section, you will learn about ...

- 3x50 Image naming convention
- Packages in the image
- Install vs. bundle boot
- Password recovery
Image Naming Convention

cat3k_caa-universalk9.SPA.03.06.04.E.152-2.E2.bin

- Converged Access
- Access Switch
- Universal License
- S - Digitally Signed
- P - Production
- A - Key Version
- IOS-XE Version
- IOSd Version
Booting IOS XE Software
What is new?

Install Boot (default mode)

- Packages are installed on flash
- Supports AP image pre-download
- No additional memory requirement
- Image must be installed in flash:
  - software expand
  - software install
- boot flash:packages.conf

Bundle Boot

- Packages are expanded in RAM
- No support for AP image pre-download
- Additional memory required
- Image can be booted from flash:, usbflash: or tftp:
  - boot flash:cat3k_caa-universalk9.SPA.03.03.03.SE.150-1.EZ3.bin
## IOS XE Packages

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat3k_caa-base</td>
<td>Kernel distribution and Open Source Software</td>
</tr>
<tr>
<td>cat3k_caa-infra</td>
<td>IOS XE infrastructure software – Installer, HA manager, etc.</td>
</tr>
<tr>
<td>cat3k_caa-iosd-universalk9</td>
<td>IOSd Software</td>
</tr>
<tr>
<td>cat3k_caa-platform</td>
<td>Platform software – FED, Stack Manager, etc.</td>
</tr>
<tr>
<td>cat3k_caa-drivers</td>
<td>Platform Drivers and UADP libraries</td>
</tr>
<tr>
<td>cat3k_caa-wcm</td>
<td>Wireless Controller Module software, AP images and Wireless Web GUI</td>
</tr>
</tbody>
</table>
Critical known issues with Install mode

- **CSCuu10600**: "%Signature verification failed" during IOS-XE upgrade, stack/INSTALL
- **CSCuw82216**: Catalyst3850: Upgrade in install mode corrupts the flash - EXT2-fs error
- These 2 issues are fixed in our recommended release. Work around: upgrade Via USB Flash or switch to Bundle mode

<table>
<thead>
<tr>
<th>Switch</th>
<th>Ports</th>
<th>Model</th>
<th>SW Version</th>
<th>SW Image</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32</td>
<td>WS-C3850-24P</td>
<td>03.06.04E</td>
<td>cat3k_caa-universalk9</td>
<td>INSTALL</td>
</tr>
</tbody>
</table>

Configuration register is 0x102

```
3850# show version
Switch Ports Model              SW Version      SW Image                Mode
------ ----- ----- ------------- -------------- ----------------------- ----
1 32    WS-C3850-24P 03.06.04E  cat3k_caa-universalk9 INSTALL
```

```
3850# show boot
-----------------------------------
Switch 1
-----------------------------------
Current Boot Variables:
BOOT variable = flash:packages.conf;

Boot Variables on next reload:
BOOT variable = flash:packages.conf;
Manual Boot = no
Enable Break = no
```
From Install mode to bundle mode … End

```
3850(config)# no boot system
3850(config)# boot system switch all flash:cat3k_caa-universalk9.SPA.03.06.04.E.152-2.E2.bin
3850(config)# do write mem
Building configuration...
Compressed configuration from 5100 bytes to 2737 bytes[OK]
```

```
3850# reload
Reload command is being issued on Active unit, this will reload the whole stack
Proceed with reload? [confirm]
<Snip>...
3850# show version
Cisco IOS Software, IOS-XE Software, Catalyst L3 Switch Software (CAT3K_CAA-UNIVERSALK9-M), Version 03.06.03.E
RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Mon 27-Aug-15 18:24 by prod_rel_team
<Snip>...
Switch Ports Model              SW Version        SW Image              Mode
------ ----- -----             ----------          -----------           ----
   1 32 WS-C3850-24P            03.06.04.SE       cat3k_caa-universalk9 BUNDLE

Configuration register is 0x102
```
3x50 Password recovery

Password recovery on 3x50 do NOT follow the 3750 family procedure

Power cycle switch and hold the Mode button until the status LED gets amber, that will get you in Boot Loader prompt (Switch:)

Switch: flash_init
Switch: SWITCH_IGNORE_STARTUP_CFG=1
Switch: SWITCH_DISABLE_PASSWORD_RECOVERY=0

Switch: boot

--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]: no
Press RETURN to get started!
Switch> enable
Switch#
3850/3650 Password recovery - End

Switch# configure terminal
Switch(config)# no enable password
Switch(config)# no enable secret
Switch(config)# enable secret <New Password>

Switch(config)# no system ignore startupconfig switch all
Switch(config)# system disable password recovery switch all

Switch(config)# end
Switch# write memory "or" copy running-config startup-config
Agenda

- Product Overview
- Image Management
- Troubleshooting Memory and CPU Resources
- Troubleshooting Stacking & High Availability
- Troubleshooting Hardware Forwarding
- QoS Implementation and Troubleshooting
- A glimpse at the future, IOS XE 16.X
- Summary
In this section, you will learn about ...

3x50 CPU complex and CPU Punt Path
- Reasons for punting packets to the CPU
- Identifying and capturing packets punted to CPU
- Troubleshooting high CPU utilization
- Troubleshooting memory Utilization
CPU Complex

- 4GB DDR3 w/ ECC
- 10/100/1000 RJ-45 Ethernet Mgmt
- USB/RJ-45 Console
- 2MB L2 Cache
- UADP 1
- UADP 2
- PCIe
- 4GB DDR3 w/ ECC
- 2GB Flash
- 64MB Bootloader
- Boot Bus
- FPGA for Stack Power
- I2C
- SGMII
- UART
- FPGA for PHY, LED, etc.
- I2C
- ACT II
- RTC
- DDR3 - 1333
- PCIe
- 500 MHz, 4 core CPU
- 27
CPU Utilization

Why Should I be concerned about high CPU utilization?

It is very important to protect the control plane for network stability, as resources (CPU, Memory and buffer) are shared by control plane and data plane traffic (sent to CPU for further processing)

What are the usual symptoms of high CPU usage?

- Control plane instability e.g., OSPF flap
- Reduced switching / forwarding performance
- Slow response to Telnet / SSH
- SNMP poll miss

At what percentage level should I start troubleshooting?

It depends on the nature and level of the traffic. It is very essential to find a baseline CPU usage during normal working conditions, and start troubleshooting when it goes above a specific threshold.

E.g., Baseline CPU usage 25%. Start troubleshooting when the CPU usage is consistently at 50% or above.
CPU Punt Path Architecture

Why is it important?

- **Processes Control Packets**
  - IOSd
    - Punt Shim
  - WCM
    - Processes Wireless Control Packets

- **Interfaces with UADP ASIC and Packet Delivery Service (PDS)**
  - Packet Handler
    - Forwarding Engine Driver
      - 32 RX PDS Queues
      - 8 TX PDS Queues
      - 32 RX Queues
      - 8 TX Queues
  - UADP ASIC
## Why Should Packets be sent to the CPU?

<table>
<thead>
<tr>
<th>Common Cause</th>
<th>Recommended Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same interface forwarding</td>
<td>change design, use “no ip redirect”</td>
</tr>
<tr>
<td>ACL logging</td>
<td>disable ACL logging</td>
</tr>
<tr>
<td>ACL deny causing switch to send ICMP unreachable</td>
<td>no ip unreachable</td>
</tr>
<tr>
<td>Forwarding/Feature exception (out of TCAM/adj space)</td>
<td>reduce TCAM usage</td>
</tr>
<tr>
<td>SW-supported feature</td>
<td>disable the feature or reduce the amount of traffic</td>
</tr>
<tr>
<td>IP packets with TTL&lt;2 or options</td>
<td>disable the offending traffic</td>
</tr>
<tr>
<td>Broadcast Storm</td>
<td>Fix STP loop, disable traffic</td>
</tr>
<tr>
<td>Unexpected control/data traffic</td>
<td>Control Plane Policing (CoPP), Deny ACL</td>
</tr>
<tr>
<td>Software Bug</td>
<td>Open a Service Request</td>
</tr>
</tbody>
</table>
### Decoding CPU Queues

#### Example Of SSH Packets Stuck

```markdown
Switch# show platform punt client

| tag   | buffer      | jumbo | fallback | packets alloc | packets free | packets conv | packets bytes | packets failures |
|-------|-------------|-------|----------|--------------|--------------|--------------|----------------|----------------|-----------------|
| 65536 | 0/1024/1600 | 0/0   | 0/512    | 64845        | 64845        | 0            | 3371071       | 0              | 0               |
| 65544 | 0/96/1600   | 0     | 0/32     | 0            | 1947         | 0            | 612588        | 0              | 0               |
| 65545 | 0/512/1600  | 0/32  | 0/512    | 13563        | 137795       | 0            | 24587306      | 0              | 0               |
| 65546 | 0/512/1600  | 0/32  | 0/256    | 10903        | 10903        | 0            | 650232        | 0              | 0               |
| 65551 | 0/512/1600  | 0/0   | 0/256    | 56           | 56           | 0            | 12088         | 0              | 0               |
| 65561 | 411/512/1600| 0/0   | 0/128    | 557245       | 556834       | 0            | 39010862      | 0              | 0               |
| 65562 | 0/512/1600  | 0/16  | 0/256    | 0            | 0            | 0            | 0             | 0              | 0               |
```

- **CPU Queue number 25 (65561 – 65536)**
- **Number of packets awaiting processing**
- **Size of Queue**
- **Size of each Buffer**

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Public
Displaying packets in the queue

Switch# show buffers detailed process iosd assigned packet | beg ng3k_rx25

Buffer information for ng3k_rx25 buffer at 0x35E98E8C
data_area 0x35E9932C, refcount 1, next 0x0, flags 0x80
linktype 7 (IP), enctype 1 (ARPA), encsize 14, rxtype 1
if_input Vlan10, if_output 0x0 (None)

source: 10.32.111.83, destination: 10.33.21.219, id: 0x4BE0, ttl: 63,
TOS: 0 prot: 6, source port 51378, destination port 22

35E99382: 6400F124 F1C11410 9FE43A49 08004500 d.q$qA...d:I..E.
35E99392: 00984BE0 40003F06 56110A20 6F530A21 ..K`@.?..V.. oS.!
35E993A2: 15DBC8B2 0016588A DB9F6C34 421A5018 /[H2..X[.14B.P.
35E993B2: FFFF8666 000072A2 E1AB5431 78970F84 ...f..r"a+T1x...
# Troubleshooting High CPU

## Identify the Culprit

**Switch# show proc cpu sorted**

<table>
<thead>
<tr>
<th>PID</th>
<th>Runtime(ms)</th>
<th>Invoked uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>TTY</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>5533</td>
<td>120300</td>
<td>1608989</td>
<td>74</td>
<td>0.29</td>
<td>0.40</td>
<td>0.42</td>
<td>1088 fed</td>
</tr>
<tr>
<td>5535</td>
<td>44890</td>
<td>1401868</td>
<td>32</td>
<td>0.24</td>
<td>0.11</td>
<td>0.10</td>
<td>0 stack-mgr</td>
</tr>
<tr>
<td>10582</td>
<td>416280</td>
<td>5787047</td>
<td>71</td>
<td>34.25</td>
<td>0.57</td>
<td>0.62</td>
<td>34816 iosd</td>
</tr>
<tr>
<td>6201</td>
<td>111520</td>
<td>119850</td>
<td>930</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0 cpumemd</td>
</tr>
<tr>
<td>5534</td>
<td>38430</td>
<td>3608873</td>
<td>10</td>
<td>0.10</td>
<td>0</td>
<td>0.10</td>
<td>0 platform_mgr</td>
</tr>
<tr>
<td>10578</td>
<td>115030</td>
<td>4737397</td>
<td>24</td>
<td>0.10</td>
<td>0</td>
<td>0.11</td>
<td>0 wcm</td>
</tr>
<tr>
<td>5455</td>
<td>1500</td>
<td>40856</td>
<td>36</td>
<td>0.05</td>
<td>0</td>
<td>0.05</td>
<td>0 slproc</td>
</tr>
<tr>
<td>6183</td>
<td>5270</td>
<td>211347</td>
<td>24</td>
<td>0.05</td>
<td>0.05</td>
<td>0.04</td>
<td>0 obfld</td>
</tr>
<tr>
<td>6185</td>
<td>4320</td>
<td>110250</td>
<td>39</td>
<td>0.05</td>
<td>0.01</td>
<td>0.03</td>
<td>0 console_relay</td>
</tr>
<tr>
<td>6198</td>
<td>20900</td>
<td>186795</td>
<td>111</td>
<td>0.05</td>
<td>0.02</td>
<td>0.00</td>
<td>0 ffm</td>
</tr>
<tr>
<td>1</td>
<td>1700</td>
<td>1112</td>
<td>1528</td>
<td>0.00</td>
<td>0.09</td>
<td>1.43</td>
<td>0 init</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>138</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0 kthreadd</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1634</td>
<td>6</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0 migration</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0 sirq-high</td>
</tr>
</tbody>
</table>

**4 Core CPU**

**Platform Processes**

**IOS-XE Processes**

---

137% across 4 Cores
## Troubleshooting High CPU

**Switch# show processes cpu detailed process iosd sorted**

<table>
<thead>
<tr>
<th>PID</th>
<th>T C</th>
<th>TID</th>
<th>Runtime(ms)</th>
<th>Invoked uSecs</th>
<th>5Sec (%)</th>
<th>1Min (%)</th>
<th>5Min (%)</th>
<th>TTY</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>10582</td>
<td>L</td>
<td>451160</td>
<td>6379641 70</td>
<td>34.25</td>
<td>0.71</td>
<td>0.60</td>
<td>34816</td>
<td>iosd</td>
<td></td>
</tr>
<tr>
<td>10582</td>
<td>L 0</td>
<td>10582</td>
<td>414060    0</td>
<td>24.00</td>
<td>0.60</td>
<td>0.50</td>
<td>34816</td>
<td>iosd</td>
<td></td>
</tr>
<tr>
<td>10582</td>
<td>L 3</td>
<td>11543</td>
<td>36980</td>
<td>180107</td>
<td>0.11</td>
<td>0.10</td>
<td>0</td>
<td>iosd.fastpath</td>
<td></td>
</tr>
<tr>
<td>10582</td>
<td>L 2</td>
<td>11544</td>
<td>120</td>
<td>4777</td>
<td>0.00</td>
<td>0.00</td>
<td>34816</td>
<td>iosd.aux</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>57680</td>
<td>5216</td>
<td>3.00</td>
<td>0.33</td>
<td>0.22</td>
<td>0</td>
<td>Check heaps</td>
<td></td>
</tr>
<tr>
<td>304</td>
<td>I</td>
<td>2200</td>
<td>1790</td>
<td>12.17</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>HTTP CORE</td>
<td></td>
</tr>
<tr>
<td>218</td>
<td>I</td>
<td>2370</td>
<td>14495</td>
<td>8.33</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>IP Input</td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>I</td>
<td>190</td>
<td>214</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>RSMP Server</td>
<td></td>
</tr>
<tr>
<td>306</td>
<td>I</td>
<td>10</td>
<td>23</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>SEP NODE PROC</td>
<td></td>
</tr>
<tr>
<td>305</td>
<td>I</td>
<td>2</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IPC ISSU Dispatch</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>I</td>
<td>220</td>
<td>336</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Pool Manager</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>I</td>
<td>0</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>HA-IDB-SYNC</td>
<td></td>
</tr>
</tbody>
</table>

- **Core 0**: CPU utilization for five seconds: 96%; one minute: 7%; five minutes: 6%
- **Core 1**: CPU utilization for five seconds: 5%; one minute: 1%; five minutes: 1%
- **Core 2**: CPU utilization for five seconds: 0%; one minute: 0%; five minutes: 0%
- **Core 3**: CPU utilization for five seconds: 41%; one minute: 1%; five minutes: 1%

### Drill Down Deeper

- **Interrupt Switched traffic (Wireless Control)**

High CPU caused by HTTP traffic

Check heaps

HTTP CORE

IP Input

RSMP Server

SEP NODE PROC

IPC ISSU Dispatch
# Command Summary - High CPU

<table>
<thead>
<tr>
<th>Troubleshooting Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check CPU usage on IOS threads</td>
<td>show process cpu detailed process iosd [sorted]</td>
</tr>
<tr>
<td>Check CPU usage on platform dependent and Nova threads</td>
<td>show process cpu detailed process {fed</td>
</tr>
<tr>
<td>Check traffic on the RX and TX CPU queues</td>
<td>show platform punt client, show platform punt tx</td>
</tr>
<tr>
<td>Check details of CPU queues</td>
<td>show platform punt statistics port-asic 0 cpuq 0 direction {rx</td>
</tr>
</tbody>
</table>
### Energy Consumption

#### Known High CPU Issues

**FED and Stack Mgr High CPU**

```
Switch# show proc cpu sorted
```

<table>
<thead>
<tr>
<th>PID</th>
<th>Runtime(ms)</th>
<th>Invoked</th>
<th>uSecs</th>
<th>5Sec</th>
<th>1Min</th>
<th>5Min</th>
<th>TTY</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>5700</td>
<td>2311985</td>
<td>24103536</td>
<td>2114</td>
<td>49.58</td>
<td>49.70</td>
<td>49.72</td>
<td>0</td>
<td>stack-mgr</td>
</tr>
<tr>
<td>5698</td>
<td>1475012</td>
<td>42309915</td>
<td>522</td>
<td>25.80</td>
<td>25.74</td>
<td>25.76</td>
<td>1088</td>
<td>led</td>
</tr>
<tr>
<td>12472</td>
<td>1779005</td>
<td>16386467</td>
<td>90</td>
<td>1.49</td>
<td>1.58</td>
<td>1.65</td>
<td>0</td>
<td>stack-mgr</td>
</tr>
<tr>
<td>6239</td>
<td>3163525</td>
<td>50452155</td>
<td>150</td>
<td>0.30</td>
<td>0.31</td>
<td>0.31</td>
<td>0</td>
<td>fed</td>
</tr>
<tr>
<td>43</td>
<td>3496392</td>
<td>43374714</td>
<td>17</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0</td>
<td>sirq-net-rx/3</td>
</tr>
<tr>
<td>29</td>
<td>70700</td>
<td>12468288</td>
<td>0</td>
<td>0.05</td>
<td>0.01</td>
<td>0.03</td>
<td>0</td>
<td>sirq-timer/2</td>
</tr>
<tr>
<td>5699</td>
<td>1747090</td>
<td>31690173</td>
<td>20</td>
<td>0.05</td>
<td>0.10</td>
<td>0.11</td>
<td>0</td>
<td>platform_mgr</td>
</tr>
</tbody>
</table>

**High CPU Issues**

- **FED and Stack Mgr** High CPU across several cores at about 150% across cores.
- **Fed Mgr** at about 75% across cores.
Known High CPU Issues

Functional Impact and Root Cause

Why is Stack-Mgr Process High?
- top/htop output in kernel and show process cpu report different values.
- Kernel counter gets rolled over and once they roll-over their values do not change – Cosmetic issue in display – Bug in Linux Kernel

Why is FED Mgr High?
- Several configuration issues unveiled code flaws that needed to be addressed in Software
- Frequent Mac Flaps and Mac Learnings events
- Frequent STP TCN (Topology Change Notification)
## Known High CPU Issues - End

### Associated defects

<table>
<thead>
<tr>
<th>DDTS</th>
<th>Description</th>
<th>Fixed Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCuo98789</td>
<td>ARP broadcast for vlan which is not SVI punted to CPU incase of Layer 2</td>
<td>3.3(4)SE</td>
</tr>
<tr>
<td>CSCuh47950</td>
<td>Routing Protocol packets cause unknown protocol drops in L2 only vlan</td>
<td>3.6(0)E, 3.3(4)SE</td>
</tr>
<tr>
<td>CSCup05630</td>
<td>Changing Aging timer does not change timer on Active/Local switch</td>
<td>3.6(1)E, 3.3(4)SE</td>
</tr>
<tr>
<td>CSCup24497</td>
<td>Serviceability and enhancement for OOB</td>
<td>3.6(1)E, 3.3(4)SE</td>
</tr>
<tr>
<td>CSCup15995</td>
<td>SifExceptionInterruptA8 need to handle all conditions besides balloting</td>
<td>3.3(4)SE</td>
</tr>
<tr>
<td>CSCup39058</td>
<td>show process cpu different from top/htop in linux kernel</td>
<td>3.3(4)SE</td>
</tr>
</tbody>
</table>

All bugs Fixed in recommended release 3.6(4)E
Memory Utilization (RAM)

Why Should I be concerned about high memory utilization?

It is very important have enough free memory to support features and network convergence events that require transient memory.

What are the usual symptoms of high memory usage?

- Memory utilization of process(es) keeps increasing
- System runs out of buffers and software packet forwarding stops
- Memory allocation failures are reported
- System crashes after reporting out of memory

At what percentage level should I start troubleshooting?

It depends on the nature and level of feature config on the switch. It is very essential to find a baseline memory usage during normal working conditions, and start troubleshooting when it goes above specific threshold.

E.g., Baseline memory usage 40%. Start troubleshooting when the memory goes above 70% and constantly keeps increasing without adding any new configuration.
## Memory Utilization (RAM)

### Why is memory utilization high?

<table>
<thead>
<tr>
<th>Common Cause</th>
<th>Recommended Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensive Config</td>
<td>Reduce configuration to supported scale</td>
</tr>
<tr>
<td>Excessive memory allocated to trace buffers</td>
<td>Reset trace buffers to default sizes</td>
</tr>
<tr>
<td>DoS Attack/Punted traffic causing buffer depletion</td>
<td>Identify packets and block them using an ACL</td>
</tr>
<tr>
<td>Protocol flaps/re-convergence causing high transient memory utilization</td>
<td>Identify reason for network instability</td>
</tr>
<tr>
<td>Memory Leak caused by software bug</td>
<td>Open a Service Request</td>
</tr>
</tbody>
</table>

**Set trace control <-> buffer default**
### Troubleshooting Memory Utilization

#### Which Process is holding most of the memory?

Switch1# **show processes memory sorted**

System memory: 3930916K total, 1118032K used, 2812884K free, 221968K kernel reserved

Lowest (b): 2252987972

<table>
<thead>
<tr>
<th>PID</th>
<th>Text</th>
<th>Data</th>
<th>Stack</th>
<th>Heap</th>
<th>RSS</th>
<th>Total</th>
<th>Process</th>
<th>Total Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>10623</td>
<td>56892</td>
<td>36452</td>
<td>92</td>
<td>5400</td>
<td>196116</td>
<td>336728</td>
<td>iosd</td>
<td>336728</td>
</tr>
<tr>
<td>5534</td>
<td>8716</td>
<td>31116</td>
<td>92</td>
<td>4620</td>
<td>136908</td>
<td>562460</td>
<td>fed</td>
<td>562460</td>
</tr>
<tr>
<td>10619</td>
<td>21976</td>
<td>555372</td>
<td>88</td>
<td>13980</td>
<td>102320</td>
<td>723240</td>
<td>wcm</td>
<td>723240</td>
</tr>
<tr>
<td>6032</td>
<td>4</td>
<td>97708</td>
<td>116</td>
<td>91996</td>
<td>99044</td>
<td>116676</td>
<td>idope.py</td>
<td>116676</td>
</tr>
<tr>
<td>12259</td>
<td>4</td>
<td>193244</td>
<td>236</td>
<td>38244</td>
<td>73672</td>
<td>299464</td>
<td>wnweb_paster.py</td>
<td>299464</td>
</tr>
<tr>
<td>5536</td>
<td>660</td>
<td>163524</td>
<td>88</td>
<td>4332</td>
<td>55968</td>
<td>336496</td>
<td>stack-mgr</td>
<td>336496</td>
</tr>
<tr>
<td>6057</td>
<td>3532</td>
<td>137308</td>
<td>88</td>
<td>2200</td>
<td>54200</td>
<td>311676</td>
<td>ffm</td>
<td>311676</td>
</tr>
<tr>
<td>6076</td>
<td>112</td>
<td>160908</td>
<td>88</td>
<td>6764</td>
<td>44728</td>
<td>233548</td>
<td>cli_agent</td>
<td>233548</td>
</tr>
<tr>
<td>6058</td>
<td>1232</td>
<td>287972</td>
<td>88</td>
<td>8112</td>
<td>38352</td>
<td>438040</td>
<td>eicored</td>
<td>438040</td>
</tr>
</tbody>
</table>
### Troubleshooting Memory Utilization - End

**Drill Down deeper - Which Process within IOSd?**

Switch1# `show processes memory detailed process iosd sorted`

<table>
<thead>
<tr>
<th>PID</th>
<th>TTY</th>
<th>Allocated</th>
<th>Freed</th>
<th>Holding</th>
<th>Getbufs</th>
<th>Retbufs</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>164</td>
<td>0</td>
<td>1534944</td>
<td>0</td>
<td>1558112</td>
<td>907264</td>
<td>0</td>
<td>NGWC DOT1X Proce</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>984492</td>
<td>0</td>
<td>0</td>
<td><em>MallocLite</em></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>657344</td>
<td>1544</td>
<td>678968</td>
<td>0</td>
<td>0</td>
<td>Chunk Manager</td>
</tr>
<tr>
<td>276</td>
<td>0</td>
<td>925564</td>
<td>297800</td>
<td>563696</td>
<td>0</td>
<td>0</td>
<td>os_info_p provid</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>415892</td>
<td>1856</td>
<td>376480</td>
<td>0</td>
<td>0</td>
<td>IPC Seat RX Cont</td>
</tr>
</tbody>
</table>

**Is holding memory increasing?**

Memory leak due to 802.1X process
# Command Summary - Memory

<table>
<thead>
<tr>
<th>Troubleshooting Steps</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check memory usage on system</td>
<td>show processes memory sorted</td>
</tr>
<tr>
<td>Check memory usage of a particular process</td>
<td>show processes memory detailed process fed</td>
</tr>
<tr>
<td>Check memory usage of IOSd</td>
<td>show processes memory detailed process iosd</td>
</tr>
<tr>
<td>Check allocators of memory within IOSd</td>
<td>show memory detailed process iosd allocating-process totals</td>
</tr>
</tbody>
</table>
Agenda

• Product Overview
• Image Management
• Troubleshooting Memory and CPU Resources
• Troubleshooting Stacking & High Availability
• Troubleshooting Hardware Forwarding
• Qos Implementation and Troubleshooting
• A glimpse at the future, IOS XE 16.X
• Summary
In this section, you will learn about ...

- 3x50 Stacking Architecture
  - Stacking Show commands
  - Troubleshooting failure to form a stack
- 3x50 HA Architecture
  - Election of Active and Standby
  - Show commands for checking HA states
Which Stacking Technology?

StackWise-480

- 3850 StackWise-480 is a new generation of Catalyst 3850 stacking
  - 240Gbps of bandwidth (120Gbps TX & 120Gbps RX per connector)
  - Similar to previous stacking implementations, ring redundancy is achieved via ring-wrap capabilities provided in hardware
  - NOT backward compatible with currently fielded stacking technologies, most notably StackWise Plus
Stack cables

• StackWise-480 currently supports 3 cables
  ➢ STACK-T1-50CM = 0.5m cable
  ➢ STACK-T1-1M = 1m cable
  ➢ STACK-T1-3M = 3m cable

• All StackWise-480 cables include ACT II chips for counterfeit protection
StackWise-160 & cables

- 3650 StackWise-160 is a new generation of Catalyst 3650 stacking
  - 160Gpbs stacking bandwidth
  - NOT backward compatible with currently fielded stacking technologies, most notably StackWise Plus
  - Stack cable can NOT be used on 3850
  - Stack cables are 50cm, 1m, and 3m in length
How many Stack Ring in my stack?

• 6 rings in total
• 3 rings go East
• 3 rings go West
• Each ring is 40G
• Total Stack BW = 240G
• With Spatial Reuse = 480G

Stack Interface of UADP

ASIC

Is math really an opinion?

Assuming 4 x 24-port 3850 Switches

Packets are segmented/reassembled in HW (256 byte segments)
Understanding Spatial Reuse
Doubling the capacity of my stack

Assuming
4 x 24-port
3850 Switches

Destination
Stripping
Packet travels
½ the rings.
Taken out of
stack by
destination
What is the status of my stack?

Switch# `show switch detail`

Switch/Stack Mac Address: `6400.f124.df80` - Local Mac Address

Mac persistency wait time: Indefinite

<table>
<thead>
<tr>
<th>Switch#</th>
<th>Role</th>
<th>Mac Address</th>
<th>Priority</th>
<th>Version</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>*1</td>
<td>Active</td>
<td>6400.f124.df80</td>
<td>10</td>
<td>0</td>
<td>Ready</td>
</tr>
<tr>
<td>2</td>
<td>Standby</td>
<td>6400.f124.de80</td>
<td>1</td>
<td>0</td>
<td>Ready</td>
</tr>
</tbody>
</table>

Stack Port Status

<table>
<thead>
<tr>
<th>Switch#</th>
<th>Port 1</th>
<th>Port 2</th>
<th>Neighbors</th>
<th>Port 1</th>
<th>Port 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OK</td>
<td>OK</td>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>OK</td>
<td>OK</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Priority, followed by MAC Address determines which switch gets elected as Active.
What is the status of my stacking ports?

Look for any anomaly in this output

```
Switch# show switch stack-ports summary
```

<table>
<thead>
<tr>
<th>Sw#/Port#</th>
<th>Port Status</th>
<th>Neighbor</th>
<th>Cable Length</th>
<th>Link OK</th>
<th>Link Active</th>
<th>Sync OK</th>
<th>#Changes to LinkOK</th>
<th>In Loopback</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1</td>
<td>OK</td>
<td>2</td>
<td>50cm</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>1/2</td>
<td>OK</td>
<td>2</td>
<td>Unknown</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>2/1</td>
<td>OK</td>
<td>1</td>
<td>100cm</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>2/2</td>
<td>OK</td>
<td>1</td>
<td>50cm</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>1</td>
<td>No</td>
</tr>
</tbody>
</table>

Cable with corrupted EEPROM
What happens When Image Version Mismatches?

• If the switches are in version mismatch state, they will not stack

• Debugging:
  • If they do not match, upgrade the switch to the Active’s version

Switch# show switch

<table>
<thead>
<tr>
<th>Switch#</th>
<th>Role</th>
<th>Mac Address</th>
<th>Priority</th>
<th>Version</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>*1</td>
<td>Active</td>
<td>6400.f125.1480</td>
<td>1</td>
<td>V01</td>
<td>Ready</td>
</tr>
<tr>
<td>2</td>
<td>Standby</td>
<td>6400.f125.2680</td>
<td>1</td>
<td>V01</td>
<td>Ready</td>
</tr>
<tr>
<td>3</td>
<td>Member</td>
<td>6400.f125.2500</td>
<td>1</td>
<td>0</td>
<td>V-Mismatch</td>
</tr>
<tr>
<td>4</td>
<td>Member</td>
<td>6400.f125.2480</td>
<td>1</td>
<td>0</td>
<td>V-Mismatch</td>
</tr>
</tbody>
</table>

Switch(config)# software auto-upgrade enable
Any newly added member automatically upgraded. Reload only new switch
What happens when there is License Mismatch?

Member switch will not stack

License mismatch. Switch will not stack

license right-to-use deactivate ipservices
license right-to-use activate ipbase acceptEULA
Reload switch
What happens when there is License Mismatch?

Command output

```
Switch# show license right-to-use slot 1

<table>
<thead>
<tr>
<th>Slot#</th>
<th>License name</th>
<th>Type</th>
<th>Count</th>
<th>Period left</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ipbase</td>
<td>permanent</td>
<td>N/A</td>
<td>Lifetime</td>
</tr>
<tr>
<td>1</td>
<td>lanbase</td>
<td>permanent</td>
<td>N/A</td>
<td>Lifetime</td>
</tr>
<tr>
<td>1</td>
<td>apcount</td>
<td>adder</td>
<td>4</td>
<td>Lifetime</td>
</tr>
</tbody>
</table>
```

Lanbase license is permanent

```
Switch# show license right-to-use mismatch

<table>
<thead>
<tr>
<th>Slot#</th>
<th>License Name</th>
<th>Adder AP Count</th>
<th>Base AP Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ipservices</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```
HA Redundancy on 3x50

A major Shift from 3750-X

Catalyst 3750-X – StackWise-Plus
- Hybrid control-plane processing
- N:1 stateless control-plane redundancy
- Distributed L2/L3 Forwarding Redundancy
- Stateless L3 protocol Redundancy

Catalyst 3850 – StackWise-480
- Centralized control-plane processing
- 1+1 Stateful redundancy (SSO)
- Distributed L2/L3 Forwarding Redundancy
- IOS HA Framework alignment for L3 protocol
HA SSO Architecture

Feature State is synced between Active and Standby Member in stack

Feature States are inactive on Standby Member
HA—Roles and Definitions

- **Route Processor Domain** – a set of SW processes (e.g. IOSd, WCM) that implement the centralized Active and Standby portions of the stack control plane.
- **Line Card Domain** – a set of SW processes (e.g. FED, Platform Manager) that implement the distributed Line Card portions of the stack control plane.
- **Infra Domain** – Support SW for the RP and LC Domains.
- **Active Switch** – supports the Active RP Domain, a LC Domain and Infra Domain.
- **Standby Switch** – supports the Standby RP Domain, a LC Domain and Infra Domain.
- **Member Switch** – supports a LC Domain and Infra Domain.
- **Election** – assigning roles or functions within the stack.
Catalyst 3x50 – HA State Machine

- Active starts RP Domain (IOSd, WCM, etc) locally
- Programs hardware on all LC Domains
- Traffic resumes once hardware is programmed
- Starts 2min Timer to elect Standby in parallel
- Active elects Standby
- Standby starts RP Domain locally
- Starts Bulk Sync with Active RP
- Standby reaches “Standby Hot”
Show switch with SSO

Switch# show switch
Switch/Stack Mac Address: 2037.06cf.0e80

<table>
<thead>
<tr>
<th>Switch#</th>
<th>Role</th>
<th>Mac Address</th>
<th>Priority</th>
<th>Version</th>
<th>Current State</th>
</tr>
</thead>
<tbody>
<tr>
<td>*1</td>
<td>Active</td>
<td>2037.06cf.0e80</td>
<td>10</td>
<td>PP</td>
<td>Ready</td>
</tr>
<tr>
<td>2</td>
<td>Standby</td>
<td>2037.06cf.3380</td>
<td>8</td>
<td>PP</td>
<td>Ready</td>
</tr>
<tr>
<td>3</td>
<td>Member</td>
<td>2037.06cf.1400</td>
<td>6</td>
<td>PP</td>
<td>Ready</td>
</tr>
<tr>
<td>4</td>
<td>Member</td>
<td>2037.06cf.3000</td>
<td>4</td>
<td>PP</td>
<td>Ready</td>
</tr>
</tbody>
</table>

* Indicates which member is providing the “stack Identity” (aka “stack MAC”)

Mac Address doesn't change for stack duration
Did I reach full SSO state?

Switch# `show redundancy states`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>my state</td>
<td>13 -ACTIVE</td>
</tr>
<tr>
<td>peer state</td>
<td>8 -STANDBY HOT</td>
</tr>
<tr>
<td>Mode</td>
<td>Duplex</td>
</tr>
<tr>
<td>Unit ID</td>
<td>2</td>
</tr>
</tbody>
</table>

Redundancy Mode (Operational) = SSO
Redundancy Mode (Configured) = SSO
Redundancy State = SSO
Manual Swact = enabled

Communications = **Up**

client count = 76
client_notification_TMR = 360000 milliseconds
keep_alive TMR = 9000 milliseconds
keep_alive count = 0
keep_alive threshold = 9
RF debug mask = 0

Terminal state for SSO. If “peer state” is stuck in any other state for more than 10 minutes, open a service request with TAC.

If Communication channel is not Up, there might be a problem with stack connectivity. Check stack cable.
Agenda

• Product Overview
• Image Management
• Troubleshooting Memory and CPU Resources
• Troubleshooting Stacking & High Availability
• Troubleshooting Hardware forwarding
• Qos Implementation and Troubleshooting
• A glimpse at the future, IOS XE 16.X
• Summary
In this section, you will learn about ...

- Importance of the TCAM (Ternary Content Addressable Memory)
- MAC Address learning troubleshooting scenario
TCAM on 3x50

What is it used for?

• Several features that need forwarding at line rate have entries stored in the TCAM
• It is partitioned in several banks and regions
• Features use a Hash Table Manager (HTM) to select and configure which region to use
• Entries wrongly programmed in TCAM will lead to wrong or unexpected forwarding decisions
### What features are using the TCAM?

#### Establish a Baseline

Switch1# `show platform tcam utilization asic all`

**CAM Utilization for ASIC# 0**

<table>
<thead>
<tr>
<th>Table</th>
<th>Max Values</th>
<th>Used Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast MAC addresses</td>
<td>32768/512</td>
<td>82/22</td>
</tr>
<tr>
<td>Directly or indirectly connected routes</td>
<td>32768/8192</td>
<td>7/89</td>
</tr>
<tr>
<td>IGMP and Multicast groups</td>
<td>8192/512</td>
<td>0/16</td>
</tr>
<tr>
<td>Security Access Control Entries</td>
<td>3072</td>
<td>173</td>
</tr>
<tr>
<td>QoS Access Control Entries</td>
<td>2816</td>
<td>52</td>
</tr>
<tr>
<td>Netflow ACEs</td>
<td>1024</td>
<td>15</td>
</tr>
<tr>
<td>Input Microflow policer ACEs</td>
<td>256</td>
<td>7</td>
</tr>
<tr>
<td>Output Microflow policer ACEs</td>
<td>256</td>
<td>7</td>
</tr>
<tr>
<td>Control Plane Entries</td>
<td>512</td>
<td>187</td>
</tr>
<tr>
<td>Policy Based Routing ACEs</td>
<td>1024</td>
<td>9</td>
</tr>
</tbody>
</table>

<Snip>
Mac Address Learning issue

Network diagram layout

Switch 1

Gig 1/0/1

Gig 1/0/4

Gig 1/0/5

Gig 2/0/10

Host 1
MAC: 001.001.001

Host 2
MAC: 002.002.002

Host 3
MAC: 003.003.003

Host 4
MAC: 004.004.004

Switch 2

Gig 2/0/2

Gig 1/0/10

Gig 2/0/10

P01

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Public
Mac Address Learning issue

How does learning happens?

• Supports Up to 32000 Mac Addresses in TCAM
• Hardware assisted Software learning
• Port ASIC learns MAC Address and puts it into a Learning Cache (Mac Address Table Manager MATM)
• FED reads MATM Table and programs Entry in TCAM
## Mac Address Learning issue

### Looking at the MATM Table

<table>
<thead>
<tr>
<th>MAC address Table for Vlan: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Why is Host 3 associated with a different type?
Mac Address Learning issue

The Meaning of Type

- A MAC Address is aged out only on the switch where it is first learned
- Other switches learn through Notifications

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>Seq#</th>
<th>macHandle</th>
<th>siHandle</th>
<th>diHandle</th>
<th>inactTime(s)</th>
<th>ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0001.0001.0001</td>
<td>1</td>
<td>33</td>
<td>0X60F4EBC0</td>
<td>0X60F4A550</td>
<td>0X0000F001</td>
<td>4</td>
<td>Gi1/0/1</td>
</tr>
</tbody>
</table>

Switch1# show platform matm macTable vlan 1 Switch 2
Mac address Table for Vlan: 1
<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>Seq#</th>
<th>macHandle</th>
<th>siHandle</th>
<th>diHandle</th>
<th>inactTime(s)</th>
<th>ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0001.0001.0001</td>
<td>257</td>
<td>33</td>
<td>0X5DA57F58</td>
<td>0X5DA57EB88</td>
<td>0X0000F001</td>
<td>0</td>
<td>Gi1/0/1</td>
</tr>
</tbody>
</table>

Host 1 is a dynamic entry on Switch 1 that will be aged on this switch

Host 1 was a dynamic entry on Switch 2 that cannot be aged out on this switch

Host 1 was learned with Seq# 33
Mac Address Learning issue

The Meaning of sequence #

- Unique number assigned by system every time the MAC address is learned
- What does it mean if it keeps changing?

Switch1# show platform matm macTable vlan 1 Switch1
Mac address Table for Vlan: 1

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC</th>
<th>Type</th>
<th>Seq#</th>
<th>macHandle</th>
<th>siHandle</th>
<th>diHandle</th>
<th>inactTime(s)</th>
<th>ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0001.0001.0001</td>
<td>1</td>
<td>33</td>
<td>0X5DC15FC0</td>
<td>0X5DC15A48</td>
<td>0X0000F001</td>
<td>0</td>
<td>Gi1/0/1</td>
</tr>
</tbody>
</table>

Switch1# clear mac address-table dynamic
Switch1# show platform matm macTable vlan 1 Switch1
1 0001.0001.0001 1 79 0X5DC17938 0X5DC15E98 0X0000F001 0 Gi1/0/1

Switch1# clear mac address-table dynamic
Switch1# show platform matm macTable vlan 1 Switch1
1 0001.0001.0001 1 85 0X5DC16068 0X5DC15FC0 0X0000F001 0 Gi1/0/1

Sequence number changing after every Clear – Unstable environment
Mac Address Learning issue

Learning issue on a Port Channel

- Ensure all physical interfaces are rightly bundled
- First step is to look at the following output

Switch1# show etherchannel 1 summary
< ..> Snip

Number of channel-groups in use: 1
Number of aggregators: 1

<table>
<thead>
<tr>
<th>Group</th>
<th>Port-channel</th>
<th>Protocol</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po1(SU)</td>
<td>-</td>
<td>Gi1/0/4(P)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Gi1/0/5(P)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Gi2/0/10(P)</td>
</tr>
</tbody>
</table>
Mac Address Learning issue

Learning issue on a Port Channel

- IIF-ID (Interface Identifier Factory ID) unique for each interface in the system
- Not programmed in Hardware, Software construct

Switch1# show platform port-asic ifm mappings etherchannel
Mappings Table

<table>
<thead>
<tr>
<th>Channel</th>
<th>Interface</th>
<th>IIF-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po1</td>
<td>0x01090040000000fa</td>
</tr>
</tbody>
</table>

This is the IIF-ID of Po1
Mac Address Learning issue

Learning issue on a Port Channel

- We now need to find IIF-id of all interfaces associated with the Port Channel IIF-ID

Switch1# show platform port-asic ifm iif-id 0x0109004000000fa
Interface Table
< Snip>
Interface Type : ETHERCHANNEL
Port Type : SWITCH PORT
Channel Number : 1
SNMP IF Index : 123
# Of Active Ports : 3
Base GPN : 584
Index[0] : 0x01006d8000000091
Index[6] : 0x0100490000000013
Index[7] : 0x0102468000000015

IIF-ID of Po1
List of associated IIF-ID
Mac Address Learning issue

Learning issue on a Port Channel

- Let's now use the IIF-ID to find the corresponding interfaces and verify their state

```
Switch1# show platform port-asic ifm iif-id 0x01006d8000000091
Interface IIF-ID : 0x01006d8000000091
Interface Name   : Gi2/0/10
Interface State  : READY

Switch1# show platform port-asic ifm iif-id 0x0100490000000013
Interface IIF-ID : 0x0100490000000013
Interface Name   : Gi1/0/4
Interface Block Pointer : 0x5e2de0d0
Interface State  : READY

Switch1# show platform port-asic ifm iif-id 0x0102468000000015
Interface IIF-ID : 0x0102468000000015
Interface Name   : Gi1/0/5
Interface Block Pointer : 0x5e2e3848
Interface State  : READY
```

BRKCRS-3146 © 2016 Cisco and/or its affiliates. All rights reserved. Cisco Public
Embedded Wireshark

Overview

• Allows for packet data to be captured at various points in the packet processing path; flowing through, to and from Catalyst 3850/3650 switch.

• Requires IPBase or IPServices license.

• No need to have physical access to the switch or a separate computer (unlike SPAN)

• During a Wireshark packet capture, hardware forwarding happens concurrently.

• Capture can be saved and viewed on switch itself, or can be exported as a .pcap file to be viewed on a computer.

• In our example, GigabitEthernet1/0/1 is injected with the Address Resolution Protocol (ARP) request for 10.10.10.1. The host is configured as 10.10.10.10.
Embedded Wireshark

Configuration Steps

C3850# monitor capture TESTCAP GigabitEthernet1/0/1 both
C3850# monitor capture TESTCAP file location bootflash:MY_CAP.pcap
C3850# monitor capture TESTCAP match ipv4 proto tcp eq 80
C3850# monitor capture TESTCAP start
C3850# monitor capture TESTCAP stop

Display options

C3850#show monitor capture file bootflash:MY_CAP.pcap
brief brief display
detailed detailed
display-filter Display filter
dump for dump
| Output modifiers

Define a capture point. It could be an interface, vlan or control-plane

Inline filter matching with protocol type and L4 port number. Could also use an access-list

Start/stop the capture

Wireshark supports “brief”, “detail” and “dump” options
3850# show monitor capture file flash:MY_CAP.pcap

1  0.000000 10.10.10.10 -> 10.10.10.1 IP Unknown (0xff)
2  0.000992 10.10.10.10 -> 10.10.10.1 IP Unknown (0xff)
3  0.000992 10.10.10.10 -> 10.10.10.1 IP Unknown (0xff)

You can TFTP/FTP the pcap file off of the switch and view the capture file in Wireshark
Agenda

• Product Overview
• Image Management and Licensing Scheme
• Troubleshooting Memory and CPU Resources
• Troubleshooting Stacking & High Availability
• Troubleshooting Hardware Forwarding
• Qos Implementation and Troubleshooting
• A glimpse at the future, IOS XE 16.X
• Summary
In this section, you will learn about ...

- QOS implementation
- QOS Troubleshooting examples
QoS – What’s New with 3x50

- Modular QoS based CLI (MQC)
  - Alignment with 4500E series (Sup6, Sup7)
  - Class-based Queuing, Policing, Shaping, Marking

- More Queues
  - Up to 2P6Q3T queuing capabilities
  - Standard 3750X provides 1P3Q3T
  - Not limited to 2 queue-sets
  - Flexible MQC Provisioning abstracts queuing hardware

- Per SSID Bandwidth Management

```plaintext
Policy-map PER-PORT-POLICING
  Class VOIP
    set dscp ef
    police 128000 conform-action transmit
    exceed-action drop
  Class VIDEO
    set dscp cs4
    police 384000 conform-action transmit
    exceed-action drop
  Class SIGNALING
    set dscp cs3
    police 32000 conform-action transmit exceed-action drop
  Class TRANSACTIONAL-DATA
    set dscp af21
  Class class-default
    set dscp default
```

Flexible MQC Provisioning abstracts queuing hardware.
QoS – What’s New with 3x50

Default Behavior Change

- **3750**, With “mls qos” enabled at global level all the ports are untrusted and DSCP/precedence/COS of the incoming packets are reset to 0
- **3750**, “mls qos trust” is needed at the interface level to change the trust mode
- **3850**, port is trusted by default, DSCP/precedence/COS values are retained
# 3750 MLS QoS vs. 3850 MQC QoS

<table>
<thead>
<tr>
<th></th>
<th>3750</th>
<th>3850</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Structure</strong></td>
<td>MLS</td>
<td>MQC</td>
</tr>
<tr>
<td><strong>Global Config</strong></td>
<td>Support mls qos</td>
<td>No mls qos support</td>
</tr>
<tr>
<td></td>
<td>Support some of MQC at ingress</td>
<td>Support MQC [class-map, policy-map]</td>
</tr>
<tr>
<td><strong>Interface Config</strong></td>
<td>Support mls qos config and some of MQC cli at ingress</td>
<td>Attach the policy to the interface</td>
</tr>
<tr>
<td><strong>Port Ingress</strong></td>
<td>Classification/Policing/Marking/Queuing</td>
<td>Classification/Policing/Marking</td>
</tr>
<tr>
<td><strong>Port Egress</strong></td>
<td>Queing</td>
<td>Classification/Policing/Marking/Queuing</td>
</tr>
<tr>
<td><strong>SVI Ingress</strong></td>
<td>Classification/Policing/Marking</td>
<td>Classification/Marking</td>
</tr>
<tr>
<td><strong>SVI Egress</strong></td>
<td>None</td>
<td>Classification/Marking</td>
</tr>
</tbody>
</table>

QoS Example
Verify Default trust mode on 3x50

3560-1
Ip 192.168.30.1 Vlan 30
Gi0/7 trunk switchport

Gig0/7

3560-1# ping 192.168.30.2 repeat 8 tos 40

3560-1
Ip 192.168.30.1 Vlan 30
Gi0/7 trunk switchport

Gig0/7

3850
Gi2/0/7 and Gi2/0/5 Trunk Switchport

Gig2/0/7

Access List QOS
Permit icmp host 192.168.30.1 host 192.168.30.2 dscp af11

3850
AF11=DSCP10=TOS 40

Gig2/0/5

3560-2
Ip 192.168.30.2 Vlan 30
Gi0/5 Trunk Switchport

Gig0/5

3560-2# show access-lists QOS
Extended IP access list QOS
10 permit icmp host 192.168.30.1 host 192.168.30.2 dscp af11 [8 matches]
QoS Example
Marking of packets to Af11

Class-map QOS
Match access-group name TEST

3850-1
Access-list TEST
permit icmp host 192.168.30.1 host 192.168.30.2
permit icmp host 192.168.30.2 host 192.168.30.1

3850-2
Ip 192.168.30.2 Vlan 30
Gi0/5 Trunk Swithport

AF11=DSCP10=TOS 40

3560-1
Ip 192.168.30.1 Vlan 30
Gi0/7 trunk swithport

3560-2
Ip 192.168.30.2 Vlan 30
Gi0/5 Trunk Swithport

3850#
show policy-map QOS
Policy Map QOS
Class QOS
set dscp af11
Interface gig2/0/7
service-policy input QOS
QoS Example
Marking of packets to Af11 - End

3560-1
Ip 192.168.30.1 Vlan 30
Gi0/7 trunk switchport

3560-1# ping 192.168.30.2 repeat 10

TOS = 0

3850

3560-2
Ip 192.168.30.2 Vlan 30
Gi0/5 Trunk Switchport

3560-2# show access-lists QOS
Extended IP access list QOS
10 permit icmp host 192.168.30.1 host 192.168.30.2 dscp af11 (10 matches)

3850# show platform qos policy hw_state target gigabitEthernet 2/0/7
Input policy :QOS
H/W programming State: INSTALLED IN HW
Output policy :Not attached
QoS Example
Verify incoming and corresponding outgoing packet Marking

Access-list TEST
permit icmp host 192.168.30.1 host 192.168.30.2 dscp af11
3850# show policy-map QOS
Policy Map QOS
  Class QOS
    set dscp af33
    Class class-default
    set dscp af21

3850# show platform qos dscp-cos counters gigabitEthernet 2/0/7 | in DSCP10
Ingress DSCP10 15 0
Egress  DSCP10  0  0

3850# show platform qos dscp-cos counters gigabitEthernet 2/0/5 | in DSCP30
Ingress  DSCP30  0  0
Egress    DSCP30 15  0
QoS Example
Verify incoming and corresponding outgoing packet Marking - End

3560-1
Ip 192.168.30.1 Vlan 30
Gi0/7 trunk switchport

3560-1
Ip 192.168.30.1 Vlan 30
Gi0/7 trunk switchport

3560-2
Ip 192.168.30.2 Vlan 30
Gi0/5 Trunk Switchport

AF11 = DSCP 10 = TOS 40
AF33 = DSCP30
AF21 = DSCP18

3850#
show platform qos dscp-cos counters gigabitEthernet 2/0/7 | in DSCP0
Ingress DSCP0 25 0
Egress DSCP0 0 0

3850#
show platform qos dscp-cos counters gigabitEthernet 2/0/5 | in DSCP18
Ingress DSCP18 0 0
Egress DSCP18 25 0

3560-1# ping 192.168.30.2 repeat 25
Agenda

- Product Overview
- Image Management and Licensing Scheme
- Troubleshooting Memory and CPU Resources
- Troubleshooting Stacking & High Availability
- Troubleshooting Hardware Forwarding
- Qos Implementation and Troubleshooting
- A glimpse at the future, IOS XE 16.X
- Summary
The Vision

Newly revamped IOS XE Operating system

- Converged OS providing feature consistency across Enterprise products
- Architectural Innovation and feature velocity
- Operational efficiencies with modularized independent upgrades
- New technologies such as Software Defined Network (SDN) and Network Function Virtualization (NFV)
- Ease-of-Troubleshooting
Overview of IOS XE 16.1.1 (Denali)

Manageability
- Prime Infra.
- APIC-EM
- WebUI

Operating System
Unified Software Stack (IOS-XE 16.x.x)
- Switches
- Wireless
- Routers
- IoX

Platform ASICs/CPU
- CPP
- UADP
- CPU

One Single Release for all Enterprise platforms
One Operating System

Where do we stand today?

Catalyst 3850/3650
- IOS-XE 16.1.1 (Nov 2015)
- 3850/3650 mGig, G12, G24

ASR1K
- IOS-XE 16.1.3 (May 2016)
- ASR1K series routers

Additional platforms

IOS-XE 16.x
Modularized Upgrades

**Challenges:**
- Support for newer APs require IOS upgrade
- Increase in OpEx for IOS re-certification

**Benefit:**
- Sub-package upgrade reduces testing cycle
- Allows newer AP deployment w/o full Image upgrade - saves OpEx
16.X upgrade failures

Upgrade from older IOS-XE version to 16.X fails

• Booting 16.X image via TFTP without upgrading old boot loader image may fail.

    switch: boot tftp://172.16.1.1/16X_image_name.bin
Reading full image into memory.................................done
ERROR: Full package not loaded, l=418381817, f=468734554, d=50352737
Not a Valid Image

• Check release notes & remember to upgrade the boot loader version before upgrading to 16.X.

• **Solution:** Copy Polaris image to flash and boot from flash instead of TFTP. Problem not seen if boot loader is upgraded prior to upgrading to 16.X
16.X upgrade failures - End
Upgrade from older IOS-XE version to Denali 16.X fails

• Software install to upgrade from old IOS XE version to 16.X may fail if you missed “new” and “force” options.

3850#software install file flash:cat3k_caa-universalk9.16.01.01.SPA.bin
Preparing install operation ...
[1]: Starting install operation
[1]: Expanding bundle flash:cat3k_caa-universalk9.16.01.01.SPA.bin
[1]: Finished expanding bundle flash:cat3k_caa-universalk9.16.01.01.SPA.bin
[1]: Verifying and copying expanded package files to flash:
[1]: Verified and copied expanded package files to flash:
[1]: Starting compatibility checks
[1]: % Candidate package compatibility checks failed because the following package dependencies were not satisfied. Operation aborted.
[1]: % An internal error was encountered. Operation aborted.

• Solution: Re-issue the command with “new” and “force” options
Vision – Faster Troubleshooting

Contextual Troubleshooting isolates network issues faster

**Challenges:**
- User unsure of which process/feature to debug
- User ends up enabling debugging for all flows

**Answer:**
- Radioactive Tracing helps Conditional Logging across Features & Processes
Radio Active Tracing

A typical example

```
3850# debug platform condition mac 0017.59BE.3A32
```

Enable Granular debugging on MAC Address D46D.50AF.8DBD Across CPU and Process boundaries

```
3850# show debug
<snip>
Conditional Debug Global State: Start
  Conditions
  Direction
  ------------------------------  ------------------------------
  MAC Address  0017.59BE.3A32  N/A
```

Traces automatically generated

```
01/27 11:48:14.082 [dot1x] [17810]: UUID: 9800000000067, [0017.59BE.3A32]: Gi2/0/14] New client detected, sending session start event for 0017.59BE.3A32
```

```
01/27 11:48:14.082 [sadb-attr] [17810]: UUID: 9800000000067, : No record found for aaa_type: 42, data: 0017.59be.3a32
01/27 11:48:14.082 [auth-mgr] [17810]: UUID: 9800000000067, : [0000.0000.0000:unknown] Record not found for attr_type 42
01/27 11:48:14.082 [auth-mgr] [17810]: UUID: 9800000000067, [0017.59BE.3A32]: Gi2/0/14] Session Start event called with conn_hdl 6, vlan: 0, identity: 0x76000051d
```

verify condition is set and started for a given Mac Address
Agenda

• Product Overview
• Image Management and Licensing Scheme
• Troubleshooting Memory and CPU Resources
• Troubleshooting Stacking & High Availability
• Troubleshooting Hardware Forwarding
• Qos Implementation and Troubleshooting
• A glimpse at the future, IOS XE 16.X
• Summary
Recommended Material

- **BRKCRS-2700** – Evolution of the Enterprise Network
- **BRKCRS-2888** - Advanced Enterprise Campus Design: Converged Access
- **BRKARC-3438** - Cisco Catalyst 3850 and 3650 Series Switching Architecture
- **BRKCRS-3438** - High Availability in the Access


Summary

• Do you have a better understanding of:
  • Key differences between 3x50 and 3750X
  • Importance of running recommended release
  • How to establish the switch baseline

• Would you like to see:
  • More/Less of any particular topic
  • More topics
  • Longer session
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 from the Session Catalog on CiscoLive.com/us.

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Call to Action - Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Lunch & Learn
- Meet the Engineer 1:1 meetings
Closing Statement

Catalyst 3650
Deploy

Catalyst 3850
Deploy

The End
Thank you