Introduction to Cisco IOS XR Cisco Next-Generation Operating System
BRKARC-3980
Mukhtiar A. Shaikh
Agenda

- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
- Protocol Configuration and Verification
  - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDR</td>
<td>Secure Domain Router</td>
</tr>
<tr>
<td>RPL</td>
<td>Route Policy Language</td>
</tr>
<tr>
<td>SMU</td>
<td>Software Maintenance Unit</td>
</tr>
<tr>
<td>LPTS</td>
<td>Local Packet Transport Protocol</td>
</tr>
<tr>
<td>IMDB</td>
<td>In Memory Data Base</td>
</tr>
<tr>
<td>DRP</td>
<td>Distributed Route Processor</td>
</tr>
<tr>
<td>AIB</td>
<td>Adjacency Information Base</td>
</tr>
<tr>
<td>IDB</td>
<td>Interface Data base</td>
</tr>
<tr>
<td>ISSU</td>
<td>In Service Software Upgrade</td>
</tr>
<tr>
<td>IFIB</td>
<td>Internal Forwarding Information Base</td>
</tr>
<tr>
<td>pIFIB</td>
<td>Pre-IFIB</td>
</tr>
<tr>
<td>PIE</td>
<td>Package Installation Envelope</td>
</tr>
<tr>
<td>IPFRR</td>
<td>IP Fast Reroute</td>
</tr>
<tr>
<td>LFA</td>
<td>Loop Free Alternate</td>
</tr>
<tr>
<td>NSR</td>
<td>Non-Stop Forwarding</td>
</tr>
<tr>
<td>BCDL</td>
<td>Bulk Content Downloader</td>
</tr>
<tr>
<td>dSDRSC</td>
<td>Designated SDR Shelf Controller</td>
</tr>
</tbody>
</table>
Cisco IOS Software portfolio and OS Harmonization

Agenda
- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
- Protocol Configuration and Verification
  - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
Cisco Software Portfolio

- **NXOS Data Center Transformation**
- **IOS and IOS Evolution Foundation for Borderless Network Platform & Services**
- **IOS-XR Carrier Class IP NGN**

Application and Network Service Consolidation

“The Network As The Platform”
Software. For Your Network, And Beyond.

One PI Development Team for all Platforms:

- IOS & IOS XE
- NX OS
- IOS XR

Specific areas of focus:

- Behavioral consistency
- Operational consistency (Feature and System)
- Release Timing and Lifecycle
- Programmability, Integration and Investment Protection

Goal: Provide a Consistent Networking Software Strategy and User Experience for all Cisco Products
IOS-XR Architecture and Feature Overview

Agenda
- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
- Protocol Configuration and Verification
  - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
Router OS Evolution

- Monolithic Kernel
  - Centralized Infrastructure
  - Integrated Network stack
  - Centralized applications

- Micro Kernel
  - Distributed Infrastructure
  - Independent Network stack
  - Distributed applications
Upgrade specific packages/Composites

- Across Entire system
  - Useful once a feature is qualified and you want to roll it without lot of cmd
- Targeted Install to specific cards
  - Useful while a feature is being qualified
  - Reduces churn in the system to card boundary

Point Fix for software faults
Distributed Control Plane

- Routing protocols and signaling protocols can run in one or more (D)RP
- Each (D)RP can have redundancy support with standby (D)RP
- Out of resources handling for proactive planning
Distributed Forwarding Infrastructure

Single stage forwarding
- Single global Adjacency Information Base (AIB) distributed to all line cards
- Single global Interface Management DB distributed to all line cards
- Only Ingress FIB – forces forwarding features to be run in RP

Two stage forwarding
- Each line card has independent AIB only for local interfaces
- Each line card has independent Interface DB for local interfaces
- Both Ingress and Egress FIB – allows forwarding features to be independently run in LCs
Process Restartability

- Used for small/contained faults (individual or small groups of process failures)
- Processes support restarting with dynamic state recovery
  - Mirrored State via checkpoint or synchronization with peer
- First line of defense- All Processes are restartable for fault recovery
- Certain processes are ‘mandatory’ – must always be running. Failure of mandatory processes can cause RP failover
- Second line of defense - Card-level Redundancy is used when Process Restart fails-
Protected Process Memory Space

- Each process has a virtual memory space
  - Kernel/MMU maps virtual address to physical address (at page level)
  - Threads share the memory space

- One process cannot corrupt another’s memory
  - Process can only access virtual space
  - In IOS – all processes shared same virtual space

- Communication between processes via controlled APIs

- Limited use of shared memory
Process Restart

Microkernel Architecture Enables Restart of Most Processes

- Microkernel includes minimal functionality
- Non-kernel processes can be restarted
- Critical to HA, ISSU, and MDR functions
- Restarting many processes can be tricky
  - Dependent processes may also need to restart

Green areas cannot restart

- Microkernel includes minimal functionality
- Non-kernel processes can be restarted
- Critical to HA, ISSU, and MDR functions
- Restarting many processes can be tricky
  - Dependent processes may also need to restart
Protection Against DDOS

- Layered Control Plane protection using multiple policers
  - DOS Filter using L2 Congestion Control Mode
  - Line rate ACL filtering
  - Control Plane Session Validation using Pre-filter mechanisms
  - Adjustable performance for trusted control plane session treatment
  - Multiple Queues to CPU
### LTPS Stats

```
RP/0/2/CPU0:PE1#show lpts pifib hardware police location 0/1/cpu0
```

FT - Flow type ID; PPS - Packets per second configured rate

<table>
<thead>
<tr>
<th>FT Flow type</th>
<th>Rate (PPS)</th>
<th>Accept/Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 unconfigured-default</td>
<td>99</td>
<td>0/0</td>
</tr>
<tr>
<td>1 Fragment</td>
<td>1000</td>
<td>0/0</td>
</tr>
<tr>
<td>2 OSPF-mc-known</td>
<td>1500</td>
<td>0/0</td>
</tr>
<tr>
<td>3 OSPF-mc-default</td>
<td>250</td>
<td>0/0</td>
</tr>
<tr>
<td>4 OSPF-uc-known</td>
<td>2000</td>
<td>0/0</td>
</tr>
<tr>
<td>5 OSPF-uc-default</td>
<td>250</td>
<td>0/0</td>
</tr>
<tr>
<td>6 ISIS-known</td>
<td>1500</td>
<td>0/0</td>
</tr>
<tr>
<td>7 ISIS-default</td>
<td>250</td>
<td>0/0</td>
</tr>
<tr>
<td>8 BGP-known</td>
<td>2000</td>
<td>3511/0</td>
</tr>
<tr>
<td>9 BGP-cfg-peer</td>
<td>1500</td>
<td>33/0</td>
</tr>
<tr>
<td>10 BGP-default</td>
<td>500</td>
<td>55/0</td>
</tr>
<tr>
<td>11 PIM-mcast</td>
<td>1500</td>
<td>0/0</td>
</tr>
<tr>
<td>12 PIM-ucast</td>
<td>1500</td>
<td>0/0</td>
</tr>
</tbody>
</table>
```

......
Local Packet Transport Protocol (LPTS)

Interoffice Mail for Data Plane

- Enables delivery of data to distributed processes across the system hardware (RPs, DRPs)
- Used for ‘for_us’ packet prioritization and filtering
  - Sends ‘for us’ packets only to the nodes that want them
  - Uses HW policers to throttle “for us” traffic
  - Applies to data plane traffic, not IPC
- Integral firewall to protects router resources - Packet forwarding executed in HW - no impact on LC CPU
- Dynamic adjustment - Flow types reflect both application type (OSPF, BGP, ...) and trust (established, configured or unknown peer) - Additional bandwidth allowed once neighbor up
**LPTS: Dynamic Control Plane Protection**

- DCoPP is an automatic, built-in firewall for control plane traffic.
- Every Control and Management packet from the line card is rate limited in hardware to provide flood protection at RP.

### LC 1 IFIB TCAM HW Entries

<table>
<thead>
<tr>
<th>Local</th>
<th>Port</th>
<th>Remote</th>
<th>Port</th>
<th>Rate</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>ICMP</td>
<td>ANY</td>
<td>ANY</td>
<td>1000</td>
<td>low</td>
</tr>
<tr>
<td>any</td>
<td>179</td>
<td>any</td>
<td>any</td>
<td>100</td>
<td>medium</td>
</tr>
<tr>
<td>any</td>
<td>179</td>
<td>202.4.48.99</td>
<td>any</td>
<td>1000</td>
<td>medium</td>
</tr>
<tr>
<td>200.200.0.2</td>
<td>13232</td>
<td>200.200.0.1</td>
<td>646</td>
<td>100</td>
<td>medium</td>
</tr>
</tbody>
</table>

### LC 2 IFIB TCAM HW Entries ...

```sql
router bgp
  neighbor 202.4.48.99
ttl_security
! mpls ldp
...!
```

**TCP Handshake**

```sql
ttl 255
```

**DcoPP is an automatic, built-in firewall for control plane traffic.**

**Every Control and Management packet from the line card is rate limited in hardware to provide flood protection at RP.**
What Is a Secure Domain Router—SDR?

- Independent/isolated physical routing instance within a common (multi-) chassis
- Each RP & LC in chassis uniquely allocated to a specific LR
- Resource sharing between LRs is limited to fabric, power, cooling
- Acts as an independent router
  - Processors not shared – CPU resources not in contention
  - Memory not shared – Memory leaks can only affect that SDR
- Hardware Failures Isolated

Diagram:

Moving to Secure Domain Routers to Complete Physical & Logical (Memory Protected) Separation
Resource Allocation for SDR Creation

- Route Processor[s] (RPs) and Line Card[s]
- Slot level granularity
- Configure and communicate via Admin Plane
- All routing apps run in LR Plane.
  - Can not communicate with processes outside their LR.

Control Plane Separation -> Data Plane Separation -> No inter-SDR communication (apart some exceptions) -> Simple -> fault isolation
IOS-XR Boot Images and Installation

Agenda
- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
- Protocol Configuration and Verification
  - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
IOS-XR Software Packages

- **IOS-XR Image:**
  - **Base Image:**
    - Includes the following components:
      - OS
      - Admin
      - Forwarding (IPv4 / IPv6 Unicast)
  - **PIEs (Package Installation Envelope):**
    - Unique PIE for each feature including
      - MPLS, Multicast, Manageability and Security

**IOS-XR Modular Packages**

**Core**

- **Base**
  - Interface manager,
  - System database, checkpoint services
  - Configuration management, etc.

- **OS:** Kernel, file system, memory management, and other slow changing core

- **Forwarding**
  - Platform independent
  - FIB, ARP, QoS, ACL, etc.

- **Line Card**
  - Platform Dependent
  - LC ucode & drivers

- **Admin**
  - Resource Management:
    - Rack, Fabric, LR management

- **Routing:**
  - RIB, BGP, ISIS, OSPF, RPL

**Optional**

- **MPLS**
  - MPLS, UCP

- **Multicast**
  - PIM, MFIB, IGMP

- **Security:**
  - IPSec, Encryption, Decryption

- **Manageability:**
  - ORB, XML, Alarms management

**Multicast / MPLS / Security and Manageability PIEs**

- **Optional**

- **Mandatory**
Software Install Terminology
Software Maintenance Upgrade

- Provides timely temporary point fixes for urgent issues for a given package version
- Fix integrated into the subsequent IOS XR maintenance release.
- Implementation changes only. No interface changes (no changes to CLI, APIs, IPC etc.) or new feature content
- Ideally not traffic impacting (Hitless, non traffic impacting)
- SMU is named by release and bugid - Examples - hfr-rout-3.2.2.CSCei63263.pie
PIE: Package Installation Envelope

- PIEs are a **delivery mechanism** for packages
  - Used to deliver
    - Major release – New functionality (3.8, 3.9, 4.0..)
    - Maintenance release – SW fixes (3.8.1, 3.8.2….)
    - SMU – Fix for a specific bug
- Includes authentication info
- Installed from IOS XR admin mode

- .vm files are the other delivery mechanism
  - .vm files are bootable images
  - Used as the Initial Install for GSR migration
PIE Installation Concepts

- PIE install used once system is operational
- Packages can be added or upgraded
- System performs sanity checks
- 3 phase install
  - Add – Copy package and unpack
  - Activate – Restart processes/nodes with new code
  - Commit – Lock activated packages through reload

  • The mini does not have the following functionality:
    • MPLS, Multicast, Security & Manageability through XML/CWI
  • PIEs are installed from Admin mode
  • Following actions can be performed on PIEs:
    • Add / Remove
    • Activate / De-activate
install add Command
Copy Image to Disk, Verify, and Unpack

RP/0/0/CPU0:P4(admin)#install add tftp://172.21.116.8/c12k-mcast.pie-3.2.85.3I

Install: The idle timeout on this line will be suspended for synchronous install operations
Install: Starting install operation. Do not insert or remove cards until the operation completes.
RP/0/0/CPU0:P4(admin)#
Install: Now operating in asynchronous mode. Do not attempt subsequent install operations until this operation is complete.
Install 3: [ 0%] Install operation 'add /tftp://172.21.116.8/c12k-mcast.pie-3.2.85.3I to disk0:' assigned request id: 3
Install 3: [ 1%] Downloading PIE file from /tftp://172.21.116.8/c12k-mcast.pie-3.2.85.3I
Install 3: [ 1%]     Transferred 3298994 Bytes
Install 3: [ 1%] Downloaded the package to the router
Install 3: [ 1%] Verifying the package
Install 3: [ 1%] [OK]
Install 3: [ 1%] Verification of the package successful [OK]
Install 3: [ 95%] Going ahead to install the package...
Install 3: [ 95%] Add of '/tftp://172.21.116.8/c12k-mcast.pie-3.2.85.3I' completed.
Install 3: [100%] Add successful.
Install 3: [100%] The following package(s) and/or SMU(s) are now available to be activated:
Install 3: [100%]     disk0:c12k-mcast-3.2.85
Install 3: [100%] Please carefully follow the instructions in the release notes when activating any software
Install 3: [100%] Idle timeout on this line will now be resumed for synchronous install operations
Install activate Command

Begin Executing New Software

RP/0/0/CPU0:P4(admin)#install activate disk0:c12k-mcast-3.2.85
Install: The idle timeout on this line will be suspended for synchronous install operations
Install: Starting install operation. Do not insert or remove cards until the operation...
RP/0/0/CPU0:P4(admin)#
Install: Now operating in asynchronous mode. Do not attempt subsequent install operations until this operation is complete.
Install 3: [ 0%] Install operation 'activate disk0:c12k-mcast-3.2.85' assigned request id: 3
Install 3: [ 1%] Performing Inter-Package Card/Node/Scope Version Dependency Checks
Install 3: [ 1%] [OK]
Install 3: [ 1%] Checking API compatibility in software configurations...
Install 3: [ 1%] [OK]
Install 3: [ 10%] Updating software configurations.
Install 3: [ 10%] RP,DRP:
Install 3: [ 10%] Activating c12k-mcast-3.2.85
Install 3: [ 10%] Checking running configuration version compatibility with newly activated...
Install 3: [ 10%] No incompatibilities found between the activated software and router configuration.

RP/0/0/CPU0:Nov 12 14:24:01.249 : instdir[181]: %INSTMGR-6-SOFTWARE_CHANGE_END : Software change transaction 3 is COMPLETE.
Install 3: [100%] Performing software change
Install 3: [100%] Activation operation successful.
Install 3: [100%] NOTE: The changes made to software configurations will not be persistent across RP reloads. Use the command 'install commit' to make changes persistent.
Install 3: [100%] Idle timeout on this line will now be resumed for synchronous install operations
install commit Command
Lock In Activated Software Across Reload

RP/0/0/CPU0:P5(admin)#install commit
Install: The idle timeout on this line will be suspended for synchronous install operations
Install 5: [ 1%] Install operation 'commit' assigned request id: 5
Install 5: [100%] Committing uncommitted changes in software configurations.
Install 5: [100%] Commit operation successful.
Install 5: [100%] Idle timeout on this line will now be resumed for synchronous operations
Deactivating Packages

```
RP/0/0/CPU0:P5(admin)#install deactivate disk0:c12k-rp-mgb1-3.2.85
Install: The idle timeout on this line will be suspended for synchronous install operations
Install: Starting install operation. Do not insert or remove cards until the operation completes.
RP/0/0/CPU0:P5(admin)#
Install: Now operating in asynchronous mode. Do not attempt subsequent install operations until this operation is complete.
Install 8: [ 0%] Install operation 'deactivate disk0:c12k-mgb1-3.2.85' assigned request id: 8
Install 8: [ 1%] Package 'disk0:c12k-mgb1-3.2.85' is not active and cannot be deactivated.
Install 8: [ 1%] Idle timeout on this line will now be resumed for synchronous install operations
```

Package features no longer available
Package still installed
Package can be reactivated
Config Management Using CLI

Agenda
- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
- Protocol Configuration and Verification
  - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
IOS-XR and IOS Config Differences

- IOS-XR configuration is held in binary form which is quicker to parse and process - ‘show running-configuration’ is just an ASCII representation of the binary data extracted from all nodes in the system
- There is no concept of a startup configuration like in IOS
- If one copies the running config to startup, a backup config with the name “startup” is created
- Router config is based on two stage config model.
- “running” or “active” config can not be modified directly.
- Instead, user config first enters a staging area (first stage)
- Must be explicitly promoted to be part of active config (second stage).

<table>
<thead>
<tr>
<th>IOS-XR</th>
<th>IOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration changes do NOT take place after &lt;CR&gt;</td>
<td>Configurations take place immediately after &lt;CR&gt;</td>
</tr>
<tr>
<td>Configuration changes must be ‘committed’ before they take effect</td>
<td>No commit</td>
</tr>
<tr>
<td>Allows you to verify your configuration before applying it</td>
<td>No verification required</td>
</tr>
<tr>
<td>Two stage configuration model</td>
<td>Not available</td>
</tr>
<tr>
<td>Configuration rollback</td>
<td>Not available</td>
</tr>
<tr>
<td>Provision to pre-configure</td>
<td>Not available</td>
</tr>
<tr>
<td>New config plane – Admin mode</td>
<td>Not available</td>
</tr>
<tr>
<td>Feature centric</td>
<td>Interface centric</td>
</tr>
</tbody>
</table>
IOS-XR CLI: Two Stage Config Model

User establishes config session

Target Config

= 

Config Change

Adds/deletes/modifies configuration; these changes:
- Are entered in the staging area
- Are validated for syntax and authorized
- Can be reviewed and modified

Second Stage

Commit

Active Config

Promotes the changes to active configuration; these changes:
- Are verified for semantic correctness
- Are check-pointed on the router
New CLI reflects the HW position in the system
- Introduces the Hierarchical location scheme
- Each linecard has three-level identification: Shelf/Slot/cpu #
- Interfaces have the Shelf/Slot/Bay/Interface scheme

CRS-1 is designed to scale 72 linecard chassis with a potential of 1296 linecard and RP slots
- Location identifiers use R/S/M/I format
  - R = Rack (applicable in multi-chassis systems)
  - S = Slot (physical slot the module is in)
  - M = module (0 for ‘fixed’ PLIMs, n for SPAs)
  - I = Interface

Protocol referenced by address family type – v4/v6

Backward compatible command-set with IOS

```
RP/0/0/CPU0:Router-1#show ipv4 interface brief

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>MgmtEth0/0/CPU0/0</td>
<td>10.23.1.69</td>
<td>Up</td>
<td>Up</td>
</tr>
<tr>
<td>MgmtEth0/0/CPU0/1</td>
<td>unassigned</td>
<td>Shutdown</td>
<td>Down</td>
</tr>
<tr>
<td>MgmtEth0/0/CPU0/2</td>
<td>unassigned</td>
<td>Shutdown</td>
<td>Down</td>
</tr>
<tr>
<td>GigabitEthernet0/2/0/0</td>
<td>100.12.1.1</td>
<td>Up</td>
<td>Up</td>
</tr>
</tbody>
</table>
```
IOS-XR CLI: New CLI Format and Configuration Modes

- Config modes include:
  - Privileged exec mode
  - Global config mode
  - Config sub-mode
  - Admin mode

- Admin mode is newly introduced compared to IOS
- Admin mode allows viewing / configuring shared resources
  - Fabric
  - Logical Router
  - Package installation

```bash
RP/0/0/CPU0:ios#config t
RP/0/0/CPU0:ios(config)#interface MgmtEth 0/0/CPU0/0
RP/0/0/CPU0:ios(config-if)#
RP/0/0/CPU0:ios#admin
RP/0/0/CPU0:ios(admin)#
```
IOS-XR CLI: Config Commits

RP/0/0/CPU0:iosxr1#show run int gi0/2/0/0
% No such configuration item(s)

RP/0/0/CPU0:iosxr1#conf t
RP/0/0/CPU0:iosxr1(config)#interface gig0/2/0/0
RP/0/0/CPU0:iosxr1(config-if)#ipv4 address 100.12.1.1/24
RP/0/0/CPU0:iosxr1(config-if)#commit
RP/0/0/CPU0:Apr 24 00:49:28.119 : config[65691]: %MGBL-CONFIG-6-DB_COMMIT: Configuration committed by user 'root'. Use 'show configuration commit changes 1000000036' to view the changes.
RP/0/0/CPU0:iosxr1(config-if)#end

RP/0/0/CPU0:iosxr1#
RP/0/0/CPU0:iosxr1#show run int gigabitEthernet 0/2/0/0
interface GigabitEthernet0/2/0/0
  ipv4 address 100.12.1.1 255.255.255.0
IOS-XR CLI: Config Commit

- Commit keyword writes config into Active Config

- Supplies a commit ID to help in Config Rollback
  - 1000000036 is the commit ID in previous illustration

- List of commits can be viewed
  - History list is maintained

- Commits can be labeled with user-friendly ‘tags’
  - Eliminates the cumbersome IDs

- Config restrictions can be imposed based on user
  - In previous illustration, the user “root” is indicated

<table>
<thead>
<tr>
<th>SNo.</th>
<th>Label/ID</th>
<th>User</th>
<th>Line</th>
<th>Client</th>
<th>Time Stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000000037</td>
<td>root</td>
<td>con0_0_CPU</td>
<td>CLI</td>
<td>01:39:03 UTC Mon Apr 24 2006</td>
</tr>
<tr>
<td>2</td>
<td>1000000036</td>
<td>root</td>
<td>con0_0_CPU</td>
<td>CLI</td>
<td>01:18:10 UTC Mon Apr 24 2006</td>
</tr>
<tr>
<td>3</td>
<td>1000000035</td>
<td>root</td>
<td>con0_0_CPU</td>
<td>CLI</td>
<td>01:00:54 UTC Mon Apr 24 2006</td>
</tr>
</tbody>
</table>
IOS-XR CLI: Config Rollback

RP/0/0/CPU0:iosxr1#conf t
RP/0/0/CPU0:iosxr1(config)#hostname iox-CL12
RP/0/0/CPU0:iosxr1(config)#commit
RP/0/0/CPU0:Apr 24 01:00:55.302 : config[65691]: %MGBL-CONFIG-6-DB_COMMIT : Configuration committed by user 'root'. Use 'show configuration commit changes 1000000034' to view the changes.
RP/0/0/CPU0:iox-CL12(config)#end
RP/0/0/CPU0:iox-CL12#
RP/0/0/CPU0:iox-CL12#rollback configuration to 1000000033
Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing.
3 items committed in 1 sec (2)items/sec
Updating.RP/0/0/CPU0:Apr 24 01:01:07.143 : config_rollback[65691]: %MGBL-CONFIG-6-DB_COMMIT : Configuration committed by user 'root'. Use 'show configuration commit changes 1000000035' to view the changes.

Updated Commit database in 1 sec
Configuration successfully rolled back to '1000000033'.
RP/0/0/CPU0:iosxr1#
RP/0/0/CPU0:iosxr1#
IOS-XR CLI: Pre-Config Capabilities

- Pre-config feature allows configuring physical interfaces before they are inserted into the router.
- Preconfigured interfaces are not verified or applied until the actual interface with the matching location
- Allows reduction down time and helps improve operational tasks
- Prior to the LC being inserted
  - Select the interface
  - Configure the timing (e.g. for SONET controller)
  - Configure the framing
  - Configure the IP address

```
RP/0/0/CPU0:IOX-4(config)#interface preconfigure POS 0/4/1/0
RP/0/0/CPU0:IOX-4(config-if-pre)#ip address 1.1.1.1 255.255.255.0
RP/0/0/CPU0:IOX-4(config-if-pre)#encapsulation ppp
RP/0/0/CPU0:IOX-4(config)#controller preconfigure sonet 0/4/0/0
RP/0/0/CPU0:IOX-4(config-sonet)#clock source line
```
IOS-XR CLI: Config Error Handling

- Two levels of config error handling
- Parser/Syntax error
  - Identified by the parser when the <return> key is entered
- Commit error
  - Syntactically correct but erroneous from config commit standpoint
  - Error details viewed through “show configuration failed” command
  - Common reasons for this error include:
    - Non-atomic config sequence
    - Lack of predecessor config

```sh
RP/0/0/CPU0:ios#conf t
RP/0/0/CPU0:ios(config)#policy p1
RP/0/0/CPU0:ios(config-pmap)#class c0
RP/0/0/CPU0:ios(config-pmap-c)#set precedence 0
RP/0/0/CPU0:ios(config-pmap-c)#commit
% Failed to commit one or more configuration items during an atomic operation, no changes have been made.
Please use 'show configuration failed' to view the errors

RP/0/0/CPU0:ios(config-pmap-c)#show configuration failed
!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
policy-map p1
  class c0
    set precedence routine
!!% Class-map not configured: c0
```
RPL - Route Policy Language

Agenda
- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
  - Protocol Configuration and Verification
    - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
RPL Motivation and Basic Building Blocks

- **Scaling**
  - Using route-maps could lead to 100k – 1M lines of configuration (e.g. 1000s of BGP peers).

- **Modularity**
  - Exploit modularity to reuse common portions of configuration.

- **Parameterization**
  - For elements which are not exact copies of each other we can add parameterization (think variables) to get further re-use.

- **Improved Clarity**
  - No Silently skipped statements.
RPL Syntax: General Structure

- RPL is used in 2 steps:
  - Define the policy in configuration mode:

    Route-policy <Policy Name>
    
    statement A
    
    statement B
    
    End-policy

  - Use the policy in BGP body:

    Router bgp 99
    
    neighbor 2.2.2.2 remote-as 1
    
    address-family ipv4 unicast
    
    route-policy <policy Name> in/out
RPL - Conditional Statements – if and if-then-else

An “if” statement uses a conditional expression to decide which actions or dispositions should be taken for the given route.

If as-path in as-path-set-1 then
  drop
endif

The “if” statement also permits an “else” clause, which is executed if the expression is false.

if med eq 150 then
  set local-preference 10
elseif med eq 200 then
  set local-preference 60
else
  set local-preference 0
endif

The statements within an if statement may themselves be if statements

if community matches-every(12:34,56:78) then
  if med eq 8 then
    drop
  endif
  set local-preference 100
endif
RPL: Boolean Expressions

- Boolean expressions evaluate as either true or false.
- Compound conditions by means of boolean operators: negation (`not`), conjunction (`and`), disjunction (`or`).

```plaintext
if med eq 42 and next-hop in (1.1.1.1) then
if med eq 10 and not destination in (10.1.3.0/24) or community is (56:78)
if med eq 10 and (not destination in (10.1.3.0/24)) or community is (56:78)
```

RPL - Hierarchical Policy

```plaintext
route-policy Inner
set weight 100
end-policy

route-policy Outer
apply Inner
set community (2:666) additive
end-policy
```
Protocol Configuration and Verification
OSPF Configuration and CLI Comparison

**OSPF IOS**

```
router ospf 99
router-id 1.1.1.1
log-adjacency-changes
area 0
! Interface GigabitEthernet2/0/0
ip address 201.1.1.2 255.255.255.0
ip ospf cost 20
```

**OSPF IOS XR**

```
router ospf 99
nsr
router-id 1.1.1.1
area 0
interface GigabitEthernet0/2/0/0
cost 20
!
interface FastEthernet0/6/2/0
cost 20
```

- Although you can configure OSPF before you configure an IP address, no OSPF routing will occur until at least one IP address is configured.

```
show run router ospf
```
OSPF LSA/SPF Timers

Routing Process "ospf 100" with ID 10.0.0.1

NSR (Non-stop routing) is Enabled
Supports only single TOS(TOS0) routes
Supports opaque LSA

Router is not originating router-LSAs with maximum metric

Initial SPF schedule delay 50 msecs
Minimum hold time between two consecutive SPFs 200 msecs
Maximum wait time between two consecutive SPFs 5000 msecs
Initial LSA throttle delay 50 msecs
Minimum hold time for LSA throttle 200 msecs
Maximum wait time for LSA throttle 5000 msecs
Minimum LSA interval 200 msecs. Minimum LSA arrival 100 msecs

<snip>
**OSPF Traces**

RP/0/2/CPU0:PE1#show ospf trace

OSPF Trace Summary (100, RP/0/2/CPU0:PE1, 2048M)

<table>
<thead>
<tr>
<th>Trace Name</th>
<th>Size</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. adj</td>
<td>2048</td>
<td>1109</td>
<td>adjacency</td>
</tr>
<tr>
<td>2. adj_cycle</td>
<td>2048</td>
<td>79968</td>
<td>dbd/flood events/pkts</td>
</tr>
<tr>
<td>3. config</td>
<td>512</td>
<td>49</td>
<td>config events</td>
</tr>
<tr>
<td>4. errors</td>
<td>2048</td>
<td>1</td>
<td>errors</td>
</tr>
<tr>
<td>5. events</td>
<td>4096</td>
<td>54</td>
<td>mda/rtrid/bfd/vrf</td>
</tr>
<tr>
<td>6. ha</td>
<td>1024</td>
<td>275</td>
<td>startup/HA/NSF</td>
</tr>
<tr>
<td>7. hello</td>
<td>2048</td>
<td>1300864</td>
<td>hello events/pkts</td>
</tr>
<tr>
<td>8. idb</td>
<td>2048</td>
<td>198</td>
<td>interface</td>
</tr>
<tr>
<td>9. pkt</td>
<td>2048</td>
<td>503752</td>
<td>I/O packets</td>
</tr>
<tr>
<td>10. rib</td>
<td>2048</td>
<td>286</td>
<td>rib batching</td>
</tr>
<tr>
<td>11. spf</td>
<td>1024</td>
<td>1014</td>
<td>spf/topology</td>
</tr>
<tr>
<td>12. spf_cycle</td>
<td>2048</td>
<td>1003</td>
<td>spf/topology detail</td>
</tr>
<tr>
<td>13. te</td>
<td>2048</td>
<td>1838</td>
<td>mpls-te</td>
</tr>
<tr>
<td>14. test</td>
<td>1024</td>
<td>0</td>
<td>testing info</td>
</tr>
<tr>
<td>15. mq</td>
<td>256</td>
<td>13</td>
<td>message queue info</td>
</tr>
</tbody>
</table>

RP/0/2/CPU0:PE1#show ospf trace hello

Traces for OSPF 100 (Fri Jun 24 19:52:41)

Traces returned/requested/available: 2048/2048/2048

Trace buffer: hello

1 Jun 24 18:58:43.304 ospf_send_hello: area 0.0.0.0 intf PO0/7/0/1 from 100.14.0.1

2 Jun 24 18:58:45.737 ospf_rcv_hello: intf PO0/7/0/0 area 0.0.0.0 from 10.0.0.2 100.12.0.2

3 Jun 24 18:58:45.737 ospf_check_hello_events: intf PO0/7/0/0 area 0.0.0.0 from 100.12.0.1

4 Jun 24 18:58:48.189 ospf_rcv_hello: intf PO0/7/0/1 area 0.0.0.0 from 100.14.0.4

5 Jun 24 18:58:48.189 ospf_check_hello_events: intf PO0/7/0/1 area 0.0.0.0 from 100.14.0.1
Comparison of Cisco IOS Static Route and Cisco IOS XR Static Route

### Static Route IOS

IOS#sh run | beg ip route 192.1.1.0
ip route 192.1.1.0 255.255.255.0 g4/0
ip route 223.255.254.0 255.255.255.0 10.13.0.1

### Static Route IOS XR

RP/0/1/CPU0:IOS-XR#sh run router static

**router static**
- **address-family ipv4 unicast**
  - 43.43.44.0/24 Serial0/5/3/3/0:2
  - 192.1.1.0/24 Gigabitethernet0/4/0/0
  - 223.255.254.0/32 MgmtEth0/1/CPU0/0
- **address-family ipv6 unicast**
  - 5301::1111/128 Serial0/5/3/3/0:0
- !
**ISIS CLI Differences**

- New hierarchical CLI configuration structure
- Allows for Multiple Independent ISIS instances
- IOS XR supports IP but not OSI Connectionless Network Service (CLNS) routing
- SNP packet by default are authenticated if passwords are defined.

**IOS ISIS Configuration:**

```plaintext
router isis ios
net 47.1111.1111.0001.0000.0c00.0006.00
domain-password mydomainpasswd authenticate snp validate
area-password myareapasswd authenticate snp validate
metric-style wide
log-adjacency-changes
nsf ietf
maximum-paths 6
!
interface POS1/0/0
ip address 201.1.1.2 255.255.255.0
ip router isis ios
isis metric 30
```

**IOS XR ISIS Configuration:**

```plaintext
router isis iosxr
net 47.1111.1111.0001.0000.0c00.0007.00
nsf ietf
log adjacency changes
lsp-password mydomainpasswd level 1 send-only snp send-only
lsp-password myareapasswd level 2 send-only snp send-only
address-family ipv4 unicast
metric-style wide
maximum-paths 6
!
interface POS0/2/0/1
address-family ipv4 unicast
metric 30
```
Comparison of Cisco IOS BGP and Cisco IOS XR BGP

- New BGP configuration fundamentals consisting of hierarchical CLI.
- Distributed BGP
- Route Policy Language (RPL)

**IOS BGP Configuration**

```
routerrbgp300
  nobgpdedefaultipv4-unicast
bgplog-neighbor-changes
neighbor1.1.1.1remote-as400
neighbor1.1.1.1update-sourceLoopback0!
address-familyipv4
neighbor1.1.1.1activate
noautosummary
nosynchronization
```

**IOS XR BGP Configuration**

```
RP/0/1/CPU0:IOS-XR#sh run router bgp
routerbgp300
bgprouter-id2.2.2.2
  address-familyipv4unicast!
neighbor1.1.1.2
  remote-as400
  address-familyipv4unicast
    route-policypolicyin
  route-policypolicyout!
```
BGP: Show Commands (1/2)

RP/0/1/CPU0:Router-6# sh bgp ipv4 unicast summary
BGP router identifier 2.2.2.2, local AS number 300
BGP generic scan interval 60 secs
BGP table state: Active
BGP main routing table version 101
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.

<table>
<thead>
<tr>
<th>Process</th>
<th>RecvTblVer</th>
<th>bRIB/RIB</th>
<th>LabelVer</th>
<th>ImportVer</th>
<th>SendTblVer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>Spk</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>St/PfxRcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.1.1.2</td>
<td>0</td>
<td>400</td>
<td>2451</td>
<td>2453</td>
<td>101</td>
<td>0</td>
<td>0</td>
<td>00:24:33</td>
<td>100</td>
</tr>
</tbody>
</table>

Show ip bgp summary (IOS)
Show bgp ipv4 unicast summary (IOS XR)
Distributed BGP: Show Commands (2/2)

show bgp summary

BGP router identifier 10.0.0.1, local AS number 1
BGP main routing table version 45966
BGP scan interval 60 secs
BGP is operating in DISTRIBUTED mode.

<table>
<thead>
<tr>
<th>Process</th>
<th>RecvTblVer</th>
<th>bRIB/RIB</th>
<th>SendTblVer</th>
</tr>
</thead>
<tbody>
<tr>
<td>bRIB</td>
<td>196349</td>
<td>45966</td>
<td>45966</td>
</tr>
<tr>
<td>Speaker 1</td>
<td>113097</td>
<td>112388</td>
<td>13187</td>
</tr>
<tr>
<td>Speaker 2</td>
<td>90942</td>
<td>90379</td>
<td>13187</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>Spr</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>St/PfxRcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.101.1</td>
<td>1</td>
<td>1</td>
<td>46286</td>
<td>7</td>
<td>13187</td>
<td>982</td>
<td>0</td>
<td>00:04:27</td>
<td>47550</td>
</tr>
<tr>
<td>10.0.101.2</td>
<td>1</td>
<td>1</td>
<td>47119</td>
<td>7</td>
<td>13187</td>
<td>988</td>
<td>0</td>
<td>00:04:27</td>
<td>48384</td>
</tr>
<tr>
<td>10.0.101.3</td>
<td>1</td>
<td>1</td>
<td>46973</td>
<td>7</td>
<td>13187</td>
<td>1001</td>
<td>0</td>
<td>00:04:28</td>
<td>48237</td>
</tr>
<tr>
<td>10.0.101.4</td>
<td>1</td>
<td>1</td>
<td>46806</td>
<td>7</td>
<td>13187</td>
<td>1153</td>
<td>0</td>
<td>00:04:28</td>
<td>48071</td>
</tr>
<tr>
<td>10.0.101.5</td>
<td>2</td>
<td>1</td>
<td>34810</td>
<td>7</td>
<td>13187</td>
<td>0</td>
<td>0</td>
<td>00:04:28</td>
<td>35813</td>
</tr>
<tr>
<td>10.0.101.6</td>
<td>2</td>
<td>1</td>
<td>46801</td>
<td>7</td>
<td>13187</td>
<td>900</td>
<td>0</td>
<td>00:04:28</td>
<td>48066</td>
</tr>
</tbody>
</table>

- When to Use Distributed BGP
  - Scaling needs exceed abilities of a single RP
  - Process Isolation

- Note: There is significant overhead involved with Distributed BGP
Hmmm- No BGP Routes!!
Check Route-Policy

RP/0/2/CPU0:PE1#show bgp summary

BGP router identifier 10.0.0.1, local AS number 65518

Neighbor Spk AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down St/PfxRcd
10.0.0.10 0 65518 1397 1376 1 0 0 22:46:55 0
10.0.0.11 0 65518 1397 1376 1 0 0 22:46:54 0
20.20.1.2 0 10 5 3 1 0 0 00:00:11 0!

RP/0/2/CPU0:PE1#

RP/0/2/CPU0:Jul 5 17:28:13.829 : bgp[123]: %ROUTING-BGP-5-ADJCHANGE : neighbor 20.20.1.2 Up

RP/0/2/CPU0:Jul 5 17:28:13.829 : bgp[123]: %ROUTING-BGP-4-NBR_NOPOLICY : No inbound IPv4 Unicast policy is configured for eBGP neighbor 20.20.1.2. No IPv4 Unicast prefixes will be accepted from the neighbor until inbound policy is configured.

RP/0/2/CPU0:Jul 5 17:28:13.829 : bgp[123]: %ROUTING-BGP-4-NBR_NOPOLICY : No outbound IPv4 Unicast policy is configured for eBGP neighbor 20.20.1.2. No IPv4 Unicast prefixes will be sent to the neighbor until outbound policy is configured.
BGP Timers

RP/0/2/CPU0:PE1#show bgp neighbor 20.20.1.2

BGP neighbor is 20.20.1.2
Remote AS 10, local AS 65518, external link
Remote router ID 28.28.28.1
BGP state = Established, up for 00:00:20
Last read 00:00:20, hold time is 180, keepalive interval is 60 seconds

<snip>
Sent 3 messages, 0 notifications, 0 in queue
Minimum time between advertisement runs is 30 seconds
Minimum time between advertisement runs is 0 seconds
For Address Family: IPv4 Unicast
BGP neighbor version 1
Update group: 0.2
eBGP neighbor with no inbound or outbound policy; defaults to 'drop'

IOS Default
iBGP Adv: 5,1,0
eBGP Adv: 30 Sec

IOS-XR eBGP Adv.: 30Sec
IOS-XR iBGP Adv.: 0 Sec
Establish IBGP with Route-Reflector Scalable Way

- Peer-Groups in IOS reduce repetitive configuration
- IOS-XR does not use Peer Groups but provides more scalable and granular configuration mechanism via
  - Session Groups
  - Address Family Groups
  - Neighbor Groups
- Allows to stack and inherit very large configuration which makes it easy to read, apply and manage.

- **Session Groups**: Set of commands applicable to BGP session itself.
- **Address-Family Group**: Contents of the neighbor session, Information exchanged in the BGP session
- **Neighbor Group**: Bundling of different Session group and AF-group forms a neighbour group.
Establish BGP using Session, AF and Neighbor Groups

```bash
RP/0/7/CPU0:D3(config-bgp)#session-group ebgp
RP/0/7/CPU0:D3(config-bgp-sngrp)#remote-as 11
RP/0/7/CPU0:D3(config-bgp-sngrp)#password cisco
RP/0/7/CPU0:D3(config-bgp-sngrp)#description ebgp
RP/0/7/CPU0:D3(config-bgp-sngrp)#exit
RP/0/7/CPU0:D3(config-bgp)#af-group permit-ipv4 address-family ipv4 unicast
RP/0/7/CPU0:D3(config-bgp-afgrp)#route-policy ebgp in
RP/0/7/CPU0:D3(config-bgp-afgrp)#route-policy ebgp out
RP/0/7/CPU0:D3(config-bgp-afgrp)#maximum-prefix 1000
RP/0/7/CPU0:D3(config-bgp-afgrp)#exit
RP/0/7/CPU0:D3(config-bgp)#neighbor-group ebgp-11
RP/0/7/CPU0:D3(config-bgp-nbrgrp)#use session-group ebgp
RP/0/7/CPU0:D3(config-bgp-nbrgrp)#address-family ipv4 unicast
RP/0/7/CPU0:D3(config-bgp-nbrgrp-af)#use af-group permit-ipv4
RP/0/7/CPU0:D3(config-bgp-nbrgrp-af)#exit
RP/0/7/CPU0:D3(config-bgp-nbrgrp)#exit
RP/0/7/CPU0:D3(config-bgp)#neighbor 192.168.1.2
```
IP Fast Re-Route (IPFRR)
IP Fast Re-Route: Introduction

- IGP computes primary paths
- IGP pre-computes backup paths
- FIB pre-installs backup paths in data-plane
- Upon link failure, all traffic going over the failed link is diverted to a backup link
- Traffic is guaranteed to be diverted to backup link within 50ms after the primary link failure
- After the final IGP convergence traffic is moved to the new best path. This operation is hitless. New primary path can be equal to the original backup path, in which case traffic stays on this path.
Loop Free Alternate Fast Reroute per Prefix/Link LFA

- A per-link LFA is a neighbor that can be safely (in a loop free manner) used as an alternate next-hop for all traffic normally destined on the protected link.

- IGP computes a primary paths and also pre-computes backup paths.
- FIB pre-installs backup paths in data-plane.
- Upon link failure, all traffic going over the failed link is diverted to a backup link within 50msec.
- Upon IGP convergence, traffic may move to a new best path in a hitless manner- New path may be same as the backup path.

- A per-link LFA is a neighbor that can be safely used as an alternate next-hop for all traffic normally destined on the protected link.
- Offers prefix independent convergence.
- Backup path pre-computed- No new protocol extension required.
- Per-hop behavior (no Network-wide deployment requirement).
- Simple configuration of ISIS or OSPF per-Link or per prefix LFA FRR.
LFA FRR (Topology Dependent)

- Availability of the backup NH is dependent on the topology and link metric assignments
- Triangles are usually good
- All depends on metric assignment

Route D
NH: F
LFA: no

Route D
NH: S

RP/0/0/CPU0:ospf-3-2(config)#router ospf 1
RP/0/0/CPU0:ospf-3-2(config-ospf)#area 0
RP/0/0/CPU0:ospf-3-2(config-ospf-ar)#int pos 0/3/0/0
RP/0/0/CPU0:ospf-3-2(config-ospf-ar-if)#fast-reroute per-link enable
RP/0/0/CPU0:ospf-3-2(config-ospf-ar-if)# fast-reroute per-link exclude interface POS0/3/0/1
ISIS IPFRR Configuration

- IPFRR needs to be configured only under ISIS to compute the LFA

Sample Config –

```
router isis CL12
  interface GigabitEthernet0/1/2/1
    address-family ipv4 unicast
      ipfrr lfa level 2
```
### ISIS: “show” Commands

RP/0/RP1/CPU0:router#show isis neighbor detail

<table>
<thead>
<tr>
<th>IS-IS merv neighbors:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Id</strong></td>
<td><strong>Interface</strong></td>
<td><strong>SNPA</strong></td>
<td><strong>State</strong></td>
<td><strong>Holdtime</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Hurricane</td>
<td>Gi0/1/2/1</td>
<td><em>PtoP</em></td>
<td>Up</td>
<td>26</td>
<td>L2</td>
</tr>
<tr>
<td>Area Address(es):</td>
<td>49.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv4 Address(es):</td>
<td>10.26.48.3*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topologies:</td>
<td>IPv4 Unicast</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uptime:</td>
<td>00:32:43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hurricane</td>
<td>PO0/1/0/2</td>
<td><em>PtoP</em></td>
<td>Up</td>
<td>23</td>
<td>L2</td>
</tr>
<tr>
<td>Area Address(es):</td>
<td>49.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv4 Address(es):</td>
<td>10.22.48.3*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topologies:</td>
<td>IPv4 Unicast</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uptime:</td>
<td>00:32:43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IPFRR: LFA Neighbor: PE1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFA IPv4 address:</td>
<td>10.26.48.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFA Router address:</td>
<td>3.3.3.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFA Interface:</td>
<td>Gi0/1/2/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total neighbor count: 2**
Non-Stop Routing (NSR)
What about BGP Graceful Restart- GR?

- Requires software support/upgrade on all routers
- Requires the operators to manually tune various timers—if not correctly determined, GR may not come into effect
- Adds load on the peering routers which could cause instability
- NSF procedures add CPU load on neighbors/protocol peers
- Scalability is limited—extensive post switchover protocol activity (NSF/GR procedures)
- Not all vendors have implemented GR
Overview of Non-Stop Routing (NSR) Operation

- Unlike GR, NSR is a self-contained solution to maintain the routing topology across HA events
- TCP connections and the routing protocol sessions are migrated from the active RP to standby RP without letting the peers knowing about the switchover
- Does not depend on any protocol extensions—relies on forwarding-plane’s NSF capability
- Neighbors/protocol peers and rest of the network do not notice that an OSPF/LDP/BGP process went through a restart
  - Minimal LSA/Route information re-flooded during NSR recovery
  - Overall CPU usage greatly reduced during NSR recovery

**Improves reliability of the overall system**
NSR Configuration and Verification

```
RP/0/0/CPU0:R2#show running-config router bgp
router bgp 1
  nsr
  bgp router-id 3.3.3.3
```

```
RP/0/0/CPU0:R2#show bgp summary
BGP router identifier 3.3.3.3, local AS number 1
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0xe0000000
BGP main routing table version 561
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.

<table>
<thead>
<tr>
<th>Process</th>
<th>RecvTblVerbRIB/RIB</th>
<th>LabelVer</th>
<th>ImportVer</th>
<th>SendTblVer</th>
<th>StandbyVer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker</td>
<td>561</td>
<td>561</td>
<td>561</td>
<td>561</td>
<td>561</td>
</tr>
<tr>
<td>Neighbor</td>
<td>Spk AS MsgRcvdMsgSentTblVerInQOutQ Up/Down St/PfxRcd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.0.101.1</td>
<td>0 1 1068 1036 561 0 0 14:35:30</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
IOS XR: MPLS LDP, TE, and VPN Configuration and Verification

Agenda
- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
- Protocol Configuration and Verification
  - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
LDP

- Like other protocols in IOS XR, LDP has its own configuration submode
- All LDP configuration takes place in this submode
- There is no support for TDP in IOS XR

Rp/0/0/CPU0:Rtr(config)#mpls ldp
Rp/0/0/CPU0:Rtr(config-ldp)#?

backoff Configure session backoff parameters
discovery Configure discovery parameters
explicit-null Configure explicit-null advertisement
graceful-restart Configure LDP graceful restart feature
holdtime Configure session holdtime
interface Enable LDP on an interface and enter interface submode
label Configure label advertisement control
log Configure log information
neighbor Configure neighbor parameters
router-id Configure router Id
signalling Configure LDP signalling parameters

LDP is enabled on an interface via the interface submode underneath the LDP submode

Rp/0/0/CPU0:Rtr(config-ldp)#interface pos0/6/0/2
Rp/0/0/CPU0:Rtr(config-ldp-if)#?

discovery Configure interface LDP discovery parameter
### “show mpls forwarding”/”show mpls” Interfaces

**RP/0/0/CPU0:Rtr#** `show mpls forwarding`

<table>
<thead>
<tr>
<th>Local Label</th>
<th>Outgoing Label or ID</th>
<th>Outgoing Prefix</th>
<th>Outgoing Interface</th>
<th>Next Hop</th>
<th>Bytes</th>
<th>Switched</th>
<th>T</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Pop Label</td>
<td>192.168.1.1/32</td>
<td>PO0/6/0/2</td>
<td>192.168.6.1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Pop Label</td>
<td>192.168.1.3/32</td>
<td>PO0/6/0/0</td>
<td>192.168.7.3</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Unlabelled</td>
<td>192.168.1.4/32</td>
<td>PO0/6/0/1</td>
<td>192.168.8.5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Unlabelled</td>
<td>192.168.1.5/32</td>
<td>PO0/6/0/1</td>
<td>192.168.8.5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>29</td>
<td>192.168.1.6/32</td>
<td>PO0/6/0/2</td>
<td>192.168.6.1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>30</td>
<td>192.168.1.7/32</td>
<td>PO0/6/0/0</td>
<td>192.168.7.3</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unlabelled</td>
<td>192.168.1.7/32</td>
<td>PO0/6/0/1</td>
<td>192.168.8.5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RP/0/0/CPU0:Rtr#** `show mpls interfaces`

<table>
<thead>
<tr>
<th>Interface</th>
<th>LDP</th>
<th>Tunnel</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS0/6/0/0</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>POS0/6/0/1</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>POS0/6/0/2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### “show mpls ldp”

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backoff</td>
<td>Session Backoff table information</td>
</tr>
<tr>
<td>bindings</td>
<td>Label Information Base (LIB) information</td>
</tr>
<tr>
<td>discovery</td>
<td>LDP Discovery Hello Information</td>
</tr>
<tr>
<td>forwarding</td>
<td>Forwarding entries information</td>
</tr>
<tr>
<td>graceful-restart</td>
<td>Graceful Restart feature information</td>
</tr>
<tr>
<td>neighbor</td>
<td>Neighbor information</td>
</tr>
<tr>
<td>parameters</td>
<td>Configuration parameter information</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistics information</td>
</tr>
<tr>
<td>summary</td>
<td>Summarized information</td>
</tr>
<tr>
<td>tech-support</td>
<td>Output show commands of interest for MPLS LDP debugging</td>
</tr>
</tbody>
</table>
LDP Inbound Label Filtering

- Controls receipt of incoming labels (remote bindings) for a set of prefixes from a peer
- The configuration is specified using prefix-acl per-nbr

```
RP/0/0/CPU0:Rtr(config-ldp)# label accept for prefix-acl from peer-ip-address
```

```
RP/0/0/CPU0:Rtr# show run mpls ldp label
mpls ldp
  label
    accept
      for pfx-acl1 from 1.1.1.1
      for pfx-acl3 from 3.3.3.3
!
```
LDP Local Label Allocation Control

- Controls allocation of local labels for set of prefixes learnt from routing
- Provides savings on (control/forwarding plane) CPU/memory resource usage, as well network updates
- Supersedes outbound label filtering rules
- Useful in typical L3VPN setup to allocate/advertise labels for “only” PE’s loopback /32 addresses (i.e., BGP nexthops)
- New ACL-based configuration CLI/XML.

```
RP/0/0/CPU0:Rtr(config-ldp)# label allocate for prefix-acl
RP/0/0/CPU0:Rtr# show run mpls ldp label
mpls ldp
  label
    allocate for pfx-acl1
```

LDP Session Protection

- **Problem:**
  - Link up: IP converges much faster/earlier than MPLS (LDP).
  - Link up, MPLS traffic loss until MPLS converges.
  - Link flap: LDP session also flaps.

- **Solution:**
  - Protect an LDP (link) session by means of “parallel” source of targeted discovery/hello.
  - Given IP connectivity, LDP session is kept alive and neighbor label bindings are maintained while link is down.
  - Minimizes traffic loss as well as enabling faster MPLS convergence on link coming up.
LDP Session Protection (Cont.)

- Enable LDP session protection for all peers or set of peers (as specified by ACL) and configure session protection holdup duration

  ```
  RP/0/0/CPU0:Rtr(config-ldp)#
  session protection [for peer-acl] [duration seconds]
  ```

- Enable logging for session protection events (Initiated, Recovered, Failed)

  ```
  RP/0/0/CPU0:Rtr(config-ldp)#
  log session-protection
  ```

- When log session-protection is configured, following type of log notices are printed:

  ```
  mpls_ldp[315]: %LDP-5-SESSION_PROTECTION : Session hold up initiated for peer 3.3.3.3:0
  mpls_ldp[315]: %LDP-5-SESSION_PROTECTION : Session recovery succeeded for peer 3.3.3.3:0
  mpls_ldp[315]: %LDP-5-SESSION_PROTECTION : Session recovery failed for peer 3.3.3.3:0
  ```

- New debug to debug session protection events:

  ```
  debug mpls ldp session-protection [ peer-acl acl ]
  ```
LDP IGP Sync

- **Problem:**
  - Traffic hit on link up when IGP converges before MPLS (LDP)
  - Traffic loss when no LDP session on outbound interfaces

- **Solution:**
  - Makes sure that no traffic is routed towards links on which MPLS (LDP) is not yet converged.
  - Synchronize IGP with LDP so that LDP controls IGP metric for given link, depending on LDP state on given link.
  - A link is advertised by IGP with max metric if LDP session is not yet up or not yet converged (label bindings exchange).
LDP IGP Sync (Cont.)

- IGP sync feature enabled only under IGP
- Enabling IGP sync feature for ISIS

```
RP/0/0/CPU0:Rtr(config-isis-if-af)# mpls ldp sync [ level <1-2> ]
```

- Enabling IGP sync feature for OSPF

```
RP/0/0/CPU0:Rtr(config-isis-if-af)# mpls ldp sync
```

- To delay declaring sync up, a delay time can be configured under LDP:

```
RP/0/0/CPU0:Rtr(config-ldp)# igp sync delay seconds
```

- New debug to debug LDP IGP sync events:

```
debug mpls ldp igp sync [ interface intf-name ]
```
MPLS OAM

RP/0/2/CPU0:PE1(config)#mpls oam

RP/0/2/CPU0:PE1#ping mpls ipv4 10.0.0.3 255.255.255.255

Sending 5, 100-byte MPLS Echos to 10.0.0.3/32, timeout is 2 seconds, send interval is 0 msec:

<snip>
Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/8/10 ms

RP/0/2/CPU0:PE1#

RP/0/2/CPU0:PE1#traceroute mpls ipv4 10.0.0.3 255.255.255.255

Tracing MPLS Label Switched Path to 10.0.0.3/32, timeout is 2 seconds

<snip>
Type escape sequence to abort.

0 100.12.0.1 MRU 4470 [Labels: 16005 Exp: 0]
L 1 100.12.0.2 MRU 4470 [Labels: implicit-null Exp: 0] 9 ms
! 2 100.23.0.3 9 ms

RP/0/2/CPU0:PE1#
Traffic Engineering, FRR Configuration, and Verification
Traffic Engineering Configuration

- TE has its own submode
- RSVP also has its own (separate) submode
- TE submode controls everything that doesn’t directly relate to the signalling protocol used
- This includes link management, reoptimization, various internal timers
- TE submode also has an interface component to it
- If rsvp submode not explicitly configure reasonable defaults are used, including advertising a link with an available bandwidth of 0
TE Submode

RP/0/0/CPU0:Rtr(config)# mpls traffic-eng
RP/0/0/CPU0:Rtr(config-mpls-te)#?

bfd Configure BFD parameters
fast-reroute Fast-reroute config parameters
interface Enable MPLS-TE on an interface
link-management MPLS Link Manager subcommands
maximum Maximum number of configurable tunnels
path-selection Path Selection Configuration
reoptimize Reoptimize timers frequency
signalling Signalling options
topology Topology Database Configuration

RP/0/0/CPU0:Rtr(config-mpls-te)# interface POS0/6/0/0
RP/0/0/CPU0:Rtr# show running-config mpls traffic-eng

mpls traffic-eng
  interface POS0/6/0/0
  !
  interface POS0/6/0/1
  !
  interface POS0/6/0/2
Tunnel Head-End and RSVP Configuration

- **Tunnel head-end configuration**
  1. Create a ‘tunnel-te <num>’ interface
  2. Under this, add your head-end interface configuration

```plaintext
interface tunnel-te6
  ipv4 unnumbered Loopback0
  autoroute announce
  destination 192.168.1.6
  path-option 5 explicit name foo
  path-option 10 dynamic
!
  explicit-path name foo
  index 1 next-address ipv4 unicast 192.168.1.3
  index 2 next-address ipv4 unicast 192.168.1.8
  index 3 next-address ipv4 unicast 192.168.1.6
!
```

- **For GRE Tunnel:**
  Create ‘tunnel-ip <num>’ interface

- **Basic RSVP Submode config:**

```plaintext
RP/0/0/CPU0:Rtr# show running-config rsvp
rsvp
  interface POS0/6/0/0
    bandwidth 155520
    !
  interface POS0/6/0/1
    bandwidth 155520
    !
  interface POS0/6/0/2
    bandwidth 155520
    !
  !
```
FRR and BFD Configuration Commands

- Configure a backup-path on the interface you want to protect:

```
RP/0/0/CPU0:Rtr(config) #
mpls traffic-eng
    interface POS0/6/0/0
    backup-path tunnel-te 103
```

- XR Supports *GigE interfaces protected by a backup tunnel

```
RP/0/0/CPU0:Rtr(config) #
mpls traffic-eng interface GigabitEthernet 0/6/0/0
    bfd fast-detect
```

- XR Supports BFD running over *GigE interfaces

```
mls traffic-eng bfd ?
    minimum-interval Hello interval
    multiplier Detect multiplier
```

- User can configure BFD Parameters for FRR. Global config for the node.
FRR and BFD Configuration Commands

- Configure a backup-path on the interface you want to protect:

```plaintext
mpls traffic-eng
  interface POS0/6/0/0
    backup-path tunnel-te 103
```

- XR Supports *GigE interfaces protected by a backup tunnel

```plaintext
RP/0/0/CPU0:Rtr(config)#
mpls traffic-eng interface GigabitEthernet 0/6/0/0
  Backup tunnel 10
```

- XR Supports BFD running over *GigE interfaces

```plaintext
RP/0/0/CPU0:Rtr(config)#
mpls traffic-eng interface GigabitEthernet 0/6/0/0
  bfd fast-detect
```

```plaintext
RP/0/0/CPU0:Rtr(config)#
```

```plaintext
mpls traffic-eng bfd ?
  minimum-interval Hello interval
  multiplier Detect multiplier
```
FRR Using BFD “show” Commands

RP/0/0/CPU0:Rtr# show mpls traffic-eng link-management bfd

- Displays the BFD sessions and states created by TE

RP/0/0/CPU0:Rtr# show mpls traffic-eng link-management bfd

Link ID:: GigabitEthernet 0/6/0/0
BFD Neighbor Address: 7.3.3.1, State: Up
BFD Neighbor Address: 7.3.3.5, State: Up

Link ID:: POS0/0/0/1
No BFD Neighbor

Link ID:: POS0/0/0/2
BFD Neighbor Address: 7.4.4.1, State: Down
IOS XR L3VPN
L3VPN Features

- **Basic L3VPN**
  - PE-CE protocols: eBGP, OSPF, EIGRP, RIP, Static
  - MPLS Transport for the core: LDP, RSVP-TE

- **Inter-AS**
  - All options: Option A, B, & C

- **Label Allocation Schemes**
  - Per-prefix, Per-CE, Per-VRF
  - Per-prefix is the default allocation mode.

- **Site Of Origin**
  - as-override
  - remove private-as
  - VRF prefix-limit
  - allow-as in
  - retain route-target
  - ...

---

BRKARC-3980 © 2012 Cisco and/or its affiliates. All rights reserved. Cisco Public
Other Distinguishing Features of IOS XR

- Event based import processing
- Auto-RD—to automatically allocate RD
- Label allocation schemes
- Distributed BGP support
- Automatic trigger of Route-Refresh upon RT config change

- Integration of NHT
- Back to back VRFs
- MP-eBGP for VPNv4 address-family
- ipv4 labeled-unicast address-family
- next-hop-unchanged
- allocate-label
L3VPN - VRF Configuration

Configuring PE/RR VPNV4 iBGP Neighbors
router bgp 100
    bgp router-id 100.100.100.100
    address-family vpnv4 unicast
    neighbor 168.1.1.1
        remote-as 100
        address-family vpnv4 unicast

Configuring the VRF in the Global Mode
vrf vpn1
    description foo
    router-id 1.1.1.1
    address-family ipv4 unicast
        import route-target 100:1
        export route-target 200:1
        import route-policy vpn1-import
        export route-policy vpn1-export

Assigning interface to VRF
interface g0/1/0/2
    vrf vpn1
        ipv4 address 1.1.1.2/24
router bgp 100
  bgp router-id 100.100.100.100
  address-family vpnv4 unicast
  vrf vpn1
    rd [auto | 100:1]
    label-allocation-mode [per-ce | per-vrf]
  address-family ipv4 unicast
  neighbor 1.1.1.1
    remote-as 65523
  address-family ipv4 unicast
    route-policy vpn1-in in
    route-policy vpn1-out out

router eigrp 100
  vrf vpn1
    address-family ipv4
      router-id 100.100.100.100
      redistribute bgp 100 route-policy policy1
      interface g0/1/0/2
        site-of-origin 100:1

router ospf 100
  vrf vpn1
    router-id 100.100.100.100
    domain-id type 0005 value 000102030405
    domain-tag 101
    redistribute bgp 100 route-policy policy1
    area 0
  interface g0/1/0/2

router rip
  vrf vpn1
    redistribute bgp 100 route-policy policy1
    interface g0/1/0/2
      site-of-origin 100:1

router static
  vrf vpn1
    address-family ipv4 unicast
      redistribute bgp 100 route-policy policy1
      interface g0/1/0/2
        site-of-origin 100:1
        10.1.1.1/32 g0/1/0/2
BGP Configuration

Configuring EBGP ipv4+label/Inter-AS Neighbors

```
router bgp 100
  bgp router-id 100.100.100.100
  address-family vpnv4 unicast
  neighbor 168.1.1.1
    remote-as 100
    address-family vpnv4 unicast
  neighbor 145.1.1.1
    remote-as 101
    address-family ipv4 label-unicast
      send-label route-policy label-policy

router bgp 100
  bgp router-id 100.100.100.100
  address-family ipv4 unicast
    allocate-label route-policy vpn1-label
  neighbor 1.1.1.1
    remote-as 65523
    address-family ipv4 labeled-unicast
      route-policy vpn1-label-in in
    address-family ipv4 unicast
      route-policy vpn1-in in
      route-policy vpn1-out out
```
IOS XR Multicast Forwarding Architecture
Multicast Routing Information Base (MRIB)

- A centralized database of mroutes and attributes
- A communication medium between protocols and forwarding, e.g.,
  - IGMP and PIM
  - MSDP and PIM
  - PIM and MFIB (Multicast forwarding Information Base)
- Does not contain internal protocol state, e.g., PIM timers
Cisco IOS vs IOS XR Multicast

IOS Multicast Configuration

```plaintext
ip multicast-routing distributed
...
...
ip pim rp-address A.B.C.D

Int pos2/0/0
  ip igmp version 3
  ip pim sparse-mode
Pos5/1/0
...
```

IOS XR Multicast Configuration

```plaintext
multicast-routing address-family ipv4
  interface POS0/2/0/0
    enable
    !
  interface POS0/5/1/0
    enable
    !
  interface Serial0/5/3/3/0:0
    enable

router igmp
  interface POS0/2/0/0
    version 3
    !
  interface POS0/5/1/0
    version 3
    !
  interface Serial0/5/3/3/0:0
    version 3

router pim address-family ipv4
  rp-address A.B.C.D
  interface POS0/2/0/0
    !
  interface POS0/5/1/0
    !
  interface Serial0/5/3/3/0:0
    !
```
IOS-XR Operation and Monitoring Tools

Agenda
- Cisco IOS Software portfolio and OS Harmonization
- IOS-XR Architecture and Feature Overview
  - Modularity, Distributed Architecture, HA, Security, Manageability, SDRs
- IOS-XR Boot images and Installation
- Config Management using CLI
- Protocol Configuration and Verification
  - RPL, OSPF, Static, BGP, MPLS, Mcast
- IOS XR Operation and Monitoring Tools
Process Management

• Process
  – An executable portion of code run within its own memory space

• Threads
  – A process may contain one or more threads or a ‘sub-process’ e.g. OSPF process has a thread which handles ‘hellos’
  – A thread may only run when the parent process is allocated runtime by the system scheduler.

• Job ID (JID) and Process ID (PID)
  – Each process is allocated a Job ID# or JID when it is first run. Remains associated with the process even if the process is stopped & restarted - The PID changes if the process is stopped and restarted

• Thread ID# (TID)
  – If a process contains threads, each is assigned a TID# associated with the PID/JID.

- Basic command
  show process

- Process restart-ability
  process restart <process name | number> <option>

- Monitor commands:
  monitor processes
  monitor threads (or top)

- Troubleshooting commands:
  show exception, exception dumpcore
  show context, clear context
  process core
  follow process
'show process' Command

RP/0/RP1/CPU0:equinox# show process snmpd

Job Id: 288
PID: 143532

Executable path:/disk0/hfr-base-3.2.85/bin/snmpd

Instance #: 1
Version ID: 00.00.0000
Respawn: ON
Respawn count: 1
Max. spawns per minute: 12
Last started: Mon May 9 15:32:22 2005
Process state: Run
Package state: Normal

Started on config: cfg/gl/snmp/admin/community/ww
core: TEXT SHARDEMEM MAINMEM
Max. core: 0
startup_path: /pkg/startup/snmpd.startup
Ready: 11.636s

Process cpu time: 45.821 user, 5.058 kernel, 50.879 total

<table>
<thead>
<tr>
<th>JID</th>
<th>TID</th>
<th>Stack</th>
<th>pri</th>
<th>state</th>
<th>HR:MM:SS:MSEC NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>288</td>
<td>1</td>
<td>96K</td>
<td>10</td>
<td>Condvar</td>
<td>0:00:29:0500 snmpd</td>
</tr>
<tr>
<td>288</td>
<td>2</td>
<td>96K</td>
<td>10</td>
<td>Receive</td>
<td>0:00:00:0049 snmpd</td>
</tr>
<tr>
<td>288</td>
<td>3</td>
<td>96K</td>
<td>10</td>
<td>Receive</td>
<td>0:00:01:0427 snmpd</td>
</tr>
<tr>
<td>288</td>
<td>4</td>
<td>96K</td>
<td>10</td>
<td>Receive</td>
<td>0:00:04:0505 snmpd</td>
</tr>
<tr>
<td>288</td>
<td>5</td>
<td>96K</td>
<td>10</td>
<td>Condvar</td>
<td>0:00:00:0000 snmpd</td>
</tr>
<tr>
<td>288</td>
<td>6</td>
<td>96K</td>
<td>10</td>
<td>Receive</td>
<td>0:00:09:0788 snmpd</td>
</tr>
<tr>
<td>288</td>
<td>7</td>
<td>96K</td>
<td>10</td>
<td>Condvar</td>
<td>0:00:00:0042 snmpd</td>
</tr>
<tr>
<td>288</td>
<td>8</td>
<td>96K</td>
<td>10</td>
<td>Receive</td>
<td>0:00:00:0011 snmpd</td>
</tr>
</tbody>
</table>

SNMP Process contains 8 threads which operate under JID 288
“monitor process” Command

• Command provide Unix ‘top’ like information
• Automatically updates every 10 seconds
• Can specify the location of the node that you wish to monitor, for example 0/RP0/CPU0 or 0/2/CPU0

233 processes; 788 threads; 4663 channels, 5906 fds
CPU states: 94.8% idle, 4.1% user, 1.0% kernel
Memory: 4096M total, 3599M avail, page size 4K

<table>
<thead>
<tr>
<th>JID</th>
<th>TIDS</th>
<th>Chans</th>
<th>FDS</th>
<th>Tmrs</th>
<th>MEM</th>
<th>HH:MM:SS</th>
<th>CPU</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26</td>
<td>236</td>
<td>183</td>
<td>1</td>
<td>0</td>
<td>67:18:56</td>
<td>1.06%</td>
<td>procnto-600-smp-cisco...</td>
</tr>
<tr>
<td>256</td>
<td>5</td>
<td>39</td>
<td>21</td>
<td>4</td>
<td>292K</td>
<td>0:02:44</td>
<td>0.79%</td>
<td>packet</td>
</tr>
<tr>
<td>69</td>
<td>10</td>
<td>454</td>
<td>9</td>
<td>3</td>
<td>2M</td>
<td>0:33:07</td>
<td>0.62%</td>
<td>qnet</td>
</tr>
<tr>
<td>331</td>
<td>8</td>
<td>254</td>
<td>21</td>
<td>13</td>
<td>2M</td>
<td>0:15:20</td>
<td>0.52%</td>
<td>wdsysmon</td>
</tr>
<tr>
<td>55</td>
<td>11</td>
<td>23</td>
<td>15</td>
<td>6</td>
<td>36M</td>
<td>0:31:18</td>
<td>0.50%</td>
<td>eth_server</td>
</tr>
<tr>
<td>241</td>
<td>12</td>
<td>96</td>
<td>83</td>
<td>13</td>
<td>1M</td>
<td>0:04:54</td>
<td>0.37%</td>
<td>netio</td>
</tr>
<tr>
<td>171</td>
<td>15</td>
<td>97</td>
<td>44</td>
<td>9</td>
<td>2M</td>
<td>0:03:33</td>
<td>0.12%</td>
<td>gsp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JID</th>
<th>TIDS</th>
<th>Chans</th>
<th>FDS</th>
<th>Tmrs</th>
<th>MEM</th>
<th>HH:MM:SS</th>
<th>CPU</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>55</td>
<td>11</td>
<td>23</td>
<td>15</td>
<td>6</td>
<td>36M</td>
<td>0:00:00</td>
<td>0.00%</td>
<td>eth_server</td>
</tr>
<tr>
<td>155</td>
<td>1</td>
<td>7</td>
<td>18</td>
<td>4</td>
<td>12M</td>
<td>0:00:00</td>
<td>0.00%</td>
<td>fgid_server</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>11</td>
<td>16</td>
<td>3</td>
<td>11M</td>
<td>0:00:00</td>
<td>0.00%</td>
<td>fgid_aggregator</td>
</tr>
<tr>
<td>257</td>
<td>8</td>
<td>16</td>
<td>36</td>
<td>3</td>
<td>8M</td>
<td>0:00:00</td>
<td>0.00%</td>
<td>parser_server</td>
</tr>
<tr>
<td>65554</td>
<td>7</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>7M</td>
<td>0:00:00</td>
<td>0.00%</td>
<td>devb-ata</td>
</tr>
<tr>
<td>53</td>
<td>5</td>
<td>237</td>
<td>633</td>
<td>0</td>
<td>4M</td>
<td>0:00:00</td>
<td>0.00%</td>
<td>dllmgr</td>
</tr>
<tr>
<td>121</td>
<td>11</td>
<td>48</td>
<td>67</td>
<td>19</td>
<td>3M</td>
<td>0:00:00</td>
<td>0.00%</td>
<td>bgp</td>
</tr>
</tbody>
</table>

Similar concept for ‘Monitor threads’ command
Memory Management

- Global memory:
  - show memory summary

- Process memory
  - show memory
  - show process memory
  - show dll memory
  - show memory heap
  - show memory compare

- How to troubleshoot memory leaks – Use commands:
  - monitor process
  - show process memory
  - show memory
  - follow
  - show memory compare
# Global Memory

Physical Memory: 4096M total
Application Memory: 3951M (3600M available)
Image: 16M (bootram: 16M)
Reserved: 128M, IOMem: 2028M, flashfsys: 0
Total shared window: 8M

- Physical Memory - amount of physical memory installed on the device
- Application Memory - memory available for the system to use (total memory minus image size, reserved, and flashfsys)
- Image - size of the bootable image
- Reserved - amount of space reserved for packet memory
- IOMem - IO memory—currently used as a backup for packet memory
- Flashfsys - flash file system memory

```
RP/0/RP1/CPU0:equinox# show memory summary

Physical Memory: 4096M total
Application Memory: 3951M (3600M available)
Image: 16M (bootram: 16M)
Reserved: 128M, IOMem: 2028M, flashfsys: 0
Total shared window: 8M
```

<table>
<thead>
<tr>
<th>Address</th>
<th>Bytes</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>480b9000</td>
<td>4096</td>
<td>Program Stack (pages not allocated)</td>
</tr>
<tr>
<td>480b7000</td>
<td>122880</td>
<td>Program Stack (pages not allocated)</td>
</tr>
<tr>
<td>4820000</td>
<td>126976</td>
<td>Allocated Memory</td>
</tr>
<tr>
<td>4820100</td>
<td>290816</td>
<td>Shared Memory</td>
</tr>
<tr>
<td>4824700</td>
<td>4096</td>
<td>Program Text or Data</td>
</tr>
<tr>
<td>4825a000</td>
<td>4096</td>
<td>Program Text or Data</td>
</tr>
<tr>
<td>4825b000</td>
<td>16384</td>
<td>Allocated Memory</td>
</tr>
<tr>
<td>6010000</td>
<td>401408</td>
<td>Shared Memory</td>
</tr>
<tr>
<td>6022e000</td>
<td>49152</td>
<td>Shared Memory</td>
</tr>
<tr>
<td>6023a000</td>
<td>4096</td>
<td>Shared Memory</td>
</tr>
<tr>
<td>6034f000</td>
<td>966656</td>
<td>Reserved Memory</td>
</tr>
<tr>
<td>6043b000</td>
<td>126976</td>
<td>Shared Memory</td>
</tr>
<tr>
<td>001b000</td>
<td>4096</td>
<td>DLL Data</td>
</tr>
<tr>
<td>001c2000</td>
<td>24576</td>
<td>DLL Text</td>
</tr>
<tr>
<td>001c000</td>
<td>4096</td>
<td>DLL Data</td>
</tr>
</tbody>
</table>

Starting address in memory
Size of memory allocated
“show memory compare” Command

• Process how to use the command:
  1. show memory compare start
     Takes the initial snapshot of heap usage
  2. show memory compare end
     Takes the second snapshot of heap usage
  3. show memory compare report
     Displays the heap memory comparison report

RP/0/RP1/CPU0:equinox# show memory compare start
Successfully stored memory snapshot /harddisk:/malloc_dump/memcmp_start.out

RP/0/RP1/CPU0:equinox# show memory compare end
Successfully stored memory snapshot /harddisk:/malloc_dump/memcmp_end.out

RP/0/RP1/CPU0:equinox# show memory compare report

<table>
<thead>
<tr>
<th>JID</th>
<th>name</th>
<th>mem before</th>
<th>mem after</th>
<th>difference</th>
<th>mallocs</th>
<th>restart</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>i2c_server</td>
<td>11756</td>
<td>11916</td>
<td>160</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>283</td>
<td>shelfmgr</td>
<td>273508</td>
<td>273460</td>
<td>-48</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>bgp</td>
<td>2522256</td>
<td>2522208</td>
<td>-48</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>qnet</td>
<td>2013844</td>
<td>2013196</td>
<td>-648</td>
<td>-27</td>
<td></td>
</tr>
<tr>
<td>234</td>
<td>lpts_pa</td>
<td>408536</td>
<td>407632</td>
<td>-904</td>
<td>-14</td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>isis</td>
<td>3089108</td>
<td>3087900</td>
<td>-1208</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>314</td>
<td>tcp</td>
<td>247196</td>
<td>245740</td>
<td>-1456</td>
<td>-9</td>
<td></td>
</tr>
<tr>
<td>241</td>
<td>netio</td>
<td>808136</td>
<td>806464</td>
<td>-1672</td>
<td>-46</td>
<td></td>
</tr>
</tbody>
</table>

You are now free to remove snapshot memcmp_start.out and memcmp_end.out under /harddisk:/malloc_dump.
System Monitoring
System Monitoring

■ Commands
  - show tech
  - show system verify
  - monitor interface
  - monitor controller
‘show tech’ Command

- Traditional dump of configuration and show command outputs but can be focused on functional areas

```
RP/0/RP1/CPU0:CRS_IOX# show tech-support ?

bcdl       Output show commands of interest for bcdl debugging
cef        Output show commands of interest for CEF debugging
file       Specify a valid file name (e.g. disk0:tmp.log)
gsp        Output show commands of interest for gsp debugging
install    Show install information for Tech-support
isis       Show IS-IS-related diagnostics for Tech-support
mpls       Output show commands of interest for MPLS debugging
multicast  Show multicast related information
ospfv3     Output show commands for OSPFv3 debugging
password   Include password in output
placement  Gather lots of information about process placement
platform   show tech-support platform output
rib        Show IP RIB related information
routing    Routing show tech-support output
terminal   Send output to terminal

RP/0/RP1/CPU0:CRS_IOX# show tech-support cef ipv4 location ?
0/2/CPU0   Fully qualified location specification
0/3/CPU0   Fully qualified location specification
0/5/CPU0   Fully qualified location specification
0/RP0/CPU0 Fully qualified location specification
0/RP1/CPU0 Fully qualified location specification
WORD      Fully qualified location specification
```
Trace Functionality

- Many major functions have ‘trace’ functionality to show the last actions it conducted – a form of ‘always-on’ debug

```
RP/0/RP1/CPU0:CRS1_1# sh cef trace tailf
4 unique entries (128 possible, 0 filtered)May 19 23:37:04.172 ipv4_fib/ipv4_fib_mgr_unique 0/RP1/CPU0 5# IP-CEF platform creates hash set to TRUE.
May 20 07:04:41.574 ipv4_fib/ipv4_fib_mgr_unique 0/RP1/CPU0 5# IP-CEF-Error: Doesn't own external adjacency for 1.1.1.100/32 path 0
May 20 07:04:41.574 ipv4_fib/ipv4_fib_mgr_unique 0/RP1/CPU0 5# IP-CEF-Error: Doesn't own external adjacency for 10.1.2.0/24 path 0
May 20 07:04:41.574 ipv4_fib/ipv4_fib_mgr_unique 0/RP1/CPU0 5# IP-CEF-Error: Doesn't own external adjacency for 10.1.3.0/24 path 0
May 19 23:37:04.174 ipv4_fib/ipv4_fib_mgr_low 0/RP1/CPU0 t1  IP-CEF-Event: Spawning CEF periodic thread
May 19 23:37:04.174 ipv4_fib/ipv4_fib_mgr_low 0/RP1/CPU0 t1  IP-CEF-Event: Spawning CEF main thread
May 19 23:37:04.199 ipv4_fib/ipv4_fib_mgr_low 0/RP1/CPU0 t3  IP-CEF-Event: Connected to IMP
May 19 23:37:04.207 ipv4_fib/ipv4_fib_mgr_hi 0/RP1/CPU0 t1  IP-CEF-Event: Connected to netio
May 19 23:37:04.218 ipv4_fib/ipv4_fib_mgr_errors 0/RP1/CPU0 t1 IP-FIB: fib_pfi_if_connect pfi_ifh_bind failed : 0x434e8c00, 'pfi-ifh' detected the 'warning' condition 'Failed to contact the IFH server'
```

```
RP/0/RP1/CPU0:CRS1_1# sh sysmgr trace tailf
May 19 23:33:17.912 sysmgr/global 0/RP1/CPU0 1# t1  SYSMGR_INIT jid=0 UNKNOWN
May 20 00:27:50.879 sysmgr/global 0/RP1/CPU0 4# t13 [unknown 0x10a/1] 0x00000005
May 20 05:02:59.839 sysmgr/global 0/RP1/CPU0 1# t11  FAILOVER jid=0 UNKNOWN
May 20 05:03:15.473 sysmgr/global 0/RP1/CPU0 105# t11  PROC_NOTIFY jid=206 ipv4_smiap
May 20 05:04:15.583 sysmgr/global 0/RP1/CPU0 20# t13  PROC_EXIT jid=312 sysmgr_preload_dll
May 20 07:04:41.509 sysmgr/global 0/RP1/CPU0 61# t14  PROC_AVAILABLE jid=340 mpls_ldp
May 20 14:32:53.625 sysmgr/global 0/RP1/CPU0 2# t11  PROC_ABORT jid=288 snmpd status=139
May 20 14:32:53.996 sysmgr/global 0/RP1/CPU0 213# t11  PROC_START jid=288 snmpd
May 20 14:32:55.637 sysmgr/global 0/RP1/CPU0 209# t11  PROC_READY jid=288 snmpd
```
### “monitor interface” Command

- per interface reporting (requires MGBL package)

```
RP/0/RP1/CPU0:CRS1_1# monitor interface mgmtEth 0/rp1/CPU0/0

CRS1_1               Monitor Time: 00:00:52               SysUptime: 16:33:16
MgmtEth0/RP1/CPU0/0 is up, line protocol is up
Encapsulation ARPA

<table>
<thead>
<tr>
<th>Traffic Stats: (5 minute rates)</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Packets:</td>
<td>313326</td>
</tr>
<tr>
<td>Input pps:</td>
<td>5</td>
</tr>
<tr>
<td>Input Bytes:</td>
<td>34467898</td>
</tr>
<tr>
<td>Input Kbps:</td>
<td>3</td>
</tr>
<tr>
<td>Output Packets:</td>
<td>37633</td>
</tr>
<tr>
<td>Output pps:</td>
<td>0</td>
</tr>
<tr>
<td>Output Bytes:</td>
<td>2034463</td>
</tr>
<tr>
<td>Output Kbps:</td>
<td>0</td>
</tr>
</tbody>
</table>

Errors Stats:

<table>
<thead>
<tr>
<th>Errors Stats:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Total:</td>
</tr>
<tr>
<td>Input CRC:</td>
</tr>
<tr>
<td>Input Frame:</td>
</tr>
<tr>
<td>Input Overrun:</td>
</tr>
<tr>
<td>Output Total:</td>
</tr>
<tr>
<td>Output Underrun:</td>
</tr>
</tbody>
</table>

Quit='q', Freeze='f', Thaw='t', Clear='c', Interface='i',
Detail='d', Brief='b', Next='n', Prev='p'
```
“monitor interface” Command

- or all interfaces in the system

```plaintext
RP/0/RP1/CPU0:CRS1_1# monitor interface all

CRS1_1                  Monitor Time: 00:00:55                  SysUptime: 16:35:50

<table>
<thead>
<tr>
<th>Interface</th>
<th>Encap</th>
<th>Input pps</th>
<th>Output pps</th>
<th>Input Kbps</th>
<th>Output Kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>MgmtEth0/RP0/CPU0/0</td>
<td>ARPA</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>MgmtEth0/RP1/CPU0/0</td>
<td>ARPA</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>POS0/4/0/0</td>
<td>HDLC</td>
<td>1230</td>
<td>0</td>
<td>542210</td>
<td>0</td>
</tr>
<tr>
<td>POS0/4/0/3</td>
<td>HDLC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POS0/4/0/6</td>
<td>PPP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POS0/4/0/7</td>
<td>HDLC</td>
<td>0</td>
<td>1230</td>
<td>0</td>
<td>542210</td>
</tr>
</tbody>
</table>

Quit='q', Freeze='f', Thaw='t', Clear='c', Next set='n', Prev set='p'
```
### “monitor controller sonet” Command

```
RP/0/RP1/CPU0:CRS1_1# monitor controller sonet 0/2/0/0

equinox Monitor Time: 00:00:47 SysUptime: 172:31:04

Controller for SONET0_2_0_0

<table>
<thead>
<tr>
<th>Controller Stats</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path LOP</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Path AIS</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Path RDI</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Path BIP</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Path FEBE</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Path NEWPTR</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Path PSE</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Path NSE</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Line AIS</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Line RDI</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Line BIP</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Line FEBE</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Section LOS</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Section LOF</td>
<td>0 (0 per-sec)</td>
</tr>
<tr>
<td>Section BIP</td>
<td>0 (0 per-sec)</td>
</tr>
</tbody>
</table>

Quit='q', Freeze='f', Thaw='t', Clear='c', Select controller='s'
```
“describe” Command

Details of a command and associated process/files

RP/0/RP1/CPU0:equinox# describe show controllers pse summary

The command is defined in metro_driver.parser

Node 0/RP1/CPU0 has file metro_driver.parser for boot package /disk0/hfr-os-mbi-3.2.90/mbihfr-rp.vm from hfr-1c

Package:

```
  hfr-1c
  hfr-1c V3.2.90[3I]  linecard package for ppc
  Vendor : Cisco Systems
  Desc  : linecard package for ppc
  Build : Built on Tue May 24 23:46:10 CEST 2005
  Source : By edde-bld1 in /vws/afz/prodaction/3.2.90.3I/hfr/workspace for c2.95.3-p8
```

Component:

```
hfr-metro-driver V0.0.0[main/204]  Driver for Metro ASIC
```

File: metro_driver.parser

User needs ALL of the following taskids:

- interface (READ)
- drivers (READ)

It will take the following actions:

Spawn the process:
```
  metro_cli -t 0x1
```

Version of component code

Permissions required for execution of command

KSH command that CLI is calling – only visible to 'cisco-support' users from 3.2.0
Online Manuals: man

- Integrated commands reference with various search options:

RP/0/RP1/CPU0:ios# man keyword mpls

Following Commands Matched for 'mpls'

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>is-is</td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>is-is</td>
<td>mpls traffic-eng router-id (is-is)</td>
</tr>
<tr>
<td>is-is</td>
<td>show isis mpls traffic-eng adjacency-log</td>
</tr>
<tr>
<td>is-is</td>
<td>show isis mpls traffic-eng advertisements</td>
</tr>
<tr>
<td>is-is</td>
<td>show isis mpls traffic-eng tunnel</td>
</tr>
<tr>
<td>ospf</td>
<td>mpls traffic-eng area (ospf)</td>
</tr>
<tr>
<td>ospf</td>
<td>mpls traffic-eng router-id (ospf)</td>
</tr>
<tr>
<td>ospf</td>
<td>show ospf mpls traffic-eng</td>
</tr>
<tr>
<td>quality-of-service</td>
<td>match mpls experimental topmost</td>
</tr>
<tr>
<td>quality-of-service</td>
<td>set mpls experimental topmost</td>
</tr>
</tbody>
</table>

RP/0/RP1/CPU0:ios# man command show cef ipv4

COMMAND

    show cef ipv4

DESCRIPTION

To display the IPv4 Cisco Express Forwarding (CEF) table, use the show cef ipv4 command in EXEC mode.

    show cef ipv4 [prefix [mask] | type instance] [detail] [location node-id]

!
Announcing the New IOS XR Specialist Certification!

- Is there a training course I can take?
  - Yes! There is a New course: Implementing and Maintaining Cisco Technologies Using IOS XR (IMTXR)
  - [https://learningnetwork.cisco.com/community/certifications/service_provider/ios_xr_specialist](https://learningnetwork.cisco.com/community/certifications/service_provider/ios_xr_specialist)

- When can I get certified?
  - Launch date is June 12, 2012.
  - The IOS XR Specialist Certification exam number is 644-906.

- How can I get access to IOS XR to prepare?
  - Customer & Partners: Cisco Learning Labs (July 2012)
  - Partners: SE Gold Labs (July 2012)

- Is there a Cisco Press book?

Please spread the word about this new Cisco Specialist Certification!
Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Passport points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

Don’t forget to activate your Cisco Live Virtual account for access to all session material, communities, and on-demand and live activities throughout the year. Activate your account at the Cisco booth in the World of Solutions or visit www.ciscolive.com
Q and A
Visit the Cisco Store for Related Titles
http://theciscostores.com
Miscellaneous
Logging Archive

• use disk devices to store syslogs

RP/0/RP1/CPU0:ios(config)#logging archive ?

archive-length  The maximum no of weeks of log to maintain
archive-size    The total size of the archive
device          Configure the archive device
file-size       The maximum file size for a single log file.
frequency      The collection interval for logs
severity        The minimum severity of log messages to archive
<cr>

RP/0/RP1/CPU0:ios(config)#logging archive device ?

disk0  Use disk0 as the archive device
disk1  Use disk1 as the archive device
harddisk Use harddisk as the archive device

• Options must be set for function to work

• recommended defaults:
  • archive-length = 4 weeks
  • archive-size = 300 MB
  • device = harddisk
  • file size = 5 MB
  • frequency = daily
  • severity = informational

RP/0/8/CPU0:PE1#sh run logging archive
logging archive
device harddisk
severity informational
file-size 5
frequency daily
archive-size 300
archive-length 4
Online Manuals: man

- Integrated commands reference with various search options:

```
RP/0/RP1/CPU0:ios# man keyword mpls
```

Following Commands Matched for 'mpls'

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>is-is</td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>is-is</td>
<td>mpls traffic-eng router-id (is-is)</td>
</tr>
<tr>
<td>is-is</td>
<td>show isis mpls traffic-eng adjacency-log</td>
</tr>
<tr>
<td>is-is</td>
<td>show isis mpls traffic-eng advertisements</td>
</tr>
<tr>
<td>is-is</td>
<td>show isis mpls traffic-eng tunnel</td>
</tr>
<tr>
<td>ospf</td>
<td>mpls traffic-eng area (ospf)</td>
</tr>
<tr>
<td>ospf</td>
<td>mpls traffic-eng router-id (ospf)</td>
</tr>
<tr>
<td>ospf</td>
<td>show ospf mpls traffic-eng</td>
</tr>
<tr>
<td>quality-of-service</td>
<td>match mpls experimental topmost</td>
</tr>
<tr>
<td>quality-of-service</td>
<td>set mpls experimental topmost</td>
</tr>
</tbody>
</table>

```
RP/0/RP1/CPU0:ios# man command show cef ipv4
```

**COMMAND**

```
show cef ipv4
```

**DESCRIPTION**

To display the IPv4 Cisco Express Forwarding (CEF) table, use the `show cef ipv4` command in EXEC mode.

```
show cef ipv4 [prefix [mask] | type instance] [detail] [location node-id]
```
Single ASIC reference name across platforms

RP/0/RP1/CPU0:CRS1_1#sh controllers ?
MgmtEth  Ethernet/IEEE 802.3 interface(s)
POS      Packet over SONET network interface(s)
SONET    SONET/SDH Port controller(s)
asic     Select ASIC to peek & poke.
asic-scan Generic ASIC Scan
backplane HFR control plane
cpuctrl  Show commands for the cpuctrl ASIC
egressq Egress Queue Manager information
fabricq  Fabric queue ASIC show screens.
fdi-lib  Show fdi info.
fia      Display Fabric Interface ASIC (FIA) information
hfrpm    Show commands for HFR Platform-mgr.
ingressq Show commands for the ingressq queueing ASIC.
null     show commands for null interface
plim     PLIM Information
pse      Packet Switching Engine information
switch   Show BCM registers
system   System controller information

• Single ASIC reference name across platforms

<table>
<thead>
<tr>
<th>Abstraction name</th>
<th>Actual name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cpuctrl</td>
<td>Squid</td>
</tr>
<tr>
<td>Egressq</td>
<td>Sharq</td>
</tr>
<tr>
<td>Fabricq</td>
<td>Sponge</td>
</tr>
<tr>
<td>Ingressq</td>
<td>Sprayer</td>
</tr>
<tr>
<td>Pse</td>
<td>Metro</td>
</tr>
</tbody>
</table>

• 12k

<table>
<thead>
<tr>
<th>Abstraction name</th>
<th>Actual name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egressq</td>
<td>Conga/Stingray</td>
</tr>
<tr>
<td>Fia</td>
<td>Fuscili/Superfish+</td>
</tr>
<tr>
<td>Ingressq</td>
<td>Radar/Piranha</td>
</tr>
<tr>
<td>Pse</td>
<td>Alpha/Wahoo</td>
</tr>
</tbody>
</table>

CRS

Abstraction name    Actual name
------------------    -----------
“describe” Command

Details of a command and associated process/files

RP/0/RP1/CPU0:equinox# describe show mpls label table summary
The command is defined in mpls_lsd.parser
Node 0/RP1/CPU0 has file mpls_lsd.parser for boot package /disk0/hfr-os-mbi-3.2.90/mbihfr-rp.vm from hfr-mls

Package:
  hfr-mls
    hfr-mls V3.2.90[3I] MPLS Package
    Vendor : Cisco Systems
    Desc   : MPLS Package
    Build  : Built on Tue May 24 23:53:24 CEST 2005
    Source : By edde-bld1 in /vws/afz/production/3.2.90.3I/hfr/workspace for c2.95.3-p8

Component:
  mpls-lsd V0.0.0[main/155] Label Switching Database of MPLS Forwarding Infrastructure

File: mpls_lsd.parser

User needs ALL of the following taskids:
  mpls-te (READ) or mpls-ldp (READ) or mpls-static (READ)

It will take the following actions:
  Spawn the process:
    mpls_lsd_command -t 0x0 -s
QoS and ACLs
QoS Configuration

- No major difference in CLI configuration between IOS and IOS XR.
- New CLI command in IOS XR is “show qos interface “

```
RP/0/RP0/CPU0:IOS-XR# sh run policy-map NW
policy-map NW
  class P1
    police rate 2000 kbps
    priority
  !
  class P2
    police rate 2000 kbps burst 1000 bytes
    !
  !

RP/0/RP0/CPU0:IOS-XR# show run class-map P1
class-map match-any P1
  match precedence priority
  !

RP/0/RP0/CPU0:IOS-XR# show run class-map P2
class-map match-any P2
  match precedence immediate
  !
```
"show qos" Interface

RP/0/RP0/CPU0:IOS-XR# show qos interface poS 0/6/0/0 output

Interface POS0_6_0_0 -- output policy
Total number of classes: 3

LEVEL1 class: classid = 0x1
class name = P1
No explicit weight assigned for this class
Sharq Queue ID = 10
This Q belongs to Group = 9
Queue Max. BW. = 2396160 kbps
TailDrop Threshold(bytes)= 2995200
WRED not configured for this class
Policer slot # = 225
Policer avg. kbps = 1953 kbps
Policer peak kbps = 0 kbps
Policer conform burst configured = 0 Kbits
Policer conform burst programmed = 62496 bytes
Policer conform action = Just TX
Policer conform action value = 0
Policer exceed action = DROP PKT
Policer exceed action value = 0

<output omitted>
Base Security Using ACLs

- Creating access lists

```
RP/0/0/0:RP-POD1#conf t
RP/0/0/0:RP-POD1(config)#ip access-list Cisco
RP/0/0/0:RP-POD1(config-ipv4-acl)#
```

- Numbering statements (ACEs)

```
RP/0/0/0:RP-POD1(config-ipv4-acl)#10 permit ip
```
ACL Editing

- Adding a line (ACE)
  ```
  (config-ipv4-acl)# 30 deny udp any eq netbios-dgm any
  ```

- Removing a line (ACE)
  ```
  (config-ipv4-acl)# no 20
  ```

- Re-sequencing ACL
  ```
  resequence ip access-list <tag>
  ```

- Sample re-sequenced ACL
  ```
  ip access-list Ethernet_In
  10 deny udp any eq netbios-ns any
  20 deny udp any host 255.255.255.255 eq tftp
  30 permit any
  ```

- Copying ACL
  ```
  RP/0/0/0:rp-router#copy ipv4 access-list pod6 pod6copy
  ```
Secure Domain Routers
(a.k.a., Logical Routers)
What Is a Secure Domain Router—SDR?

- Independent/isolated physical routing instance within a common (multi-) chassis
- Each RP & LC in chassis uniquely allocated to a specific LR
- Resource sharing between LRs is limited to fabric, power, cooling
- Acts as an independent router
  - Processors not shared – CPU resources not in contention
  - Memory not shared – Memory leaks can only affect that SDR
- Hardware Failures Isolated
Resource Allocation for SDR Creation

- Route Processor[s] (RPs) and Line Card[s]
- Slot level granularity
- Configure and communicate via Admin Plane
- All routing apps run in LR Plane.
  - Can not communicate with processes outside their LR.

Control Plane Separation -> Data Plane Separation -> No inter-SDR communication (apart some exceptions) -> Simple -> fault isolation
Default/Owner SDR

- Handles inventory of unassigned cards
- When IOX router boots first time
  - All cards belongs to default LR
  - User configures to assign to diff LR (SDR)
- No configuration needed
- Head of Admin Plane - DSC
- Head of LR Plane : dLRSC/dSDRSC
Secure Domain Router Architecture
Admin vs. SDR Planes

- **SDR/LR Plane**: Everything involved with running a router. Generally more platform independent.
- **Admin Plane**: Everything involved with keeping the hardware running. More platform dependent. Also, this is where SDRs are created.
- SDRs operate like physically separate routers
- Admin and SDR planes provide Fault isolation through IPC Partitioning
Secure Domain Router Architecture Configuration Partitioning

- System owner configures fabric/chassis
- SDR owner configures LC/(D)RPs
- Fault Isolation through Config Partitioning

- SDR Plane Config
  - Local config stored within (D)RP
  - Interfaces, applications, (D)RP pairing
- Must log into SDR to configure it

- Admin Plane Config
  - Shared resource config stored in separate file
  - Accessible to CRS/GSR owner only
- Scoped within the Admin plane (RPs, SPs and SCs)
How to Create a Secure Domain Router

- There Are Three Levels of Commands…
- Admin conf t –
  - Make changes to admin plane config
- Logical-router <lr name>
  - Enter configure LR submode, name must be <= 32 chars
- Location <nodeid>
  - Specifies what node you wish to add to the LR
Entering Admin Configuration

```
RP/0/0/0:ownerLR#admin config
RP/0/0/0:ownerLR (admin-config)# sdr foo
RP/0/0/0:ownerLR (admin-config-sdr:foo)# location 0/2/*
RP/0/0/0:ownerLR (admin-config-sdr:foo)# location 0/6/*
RP/0/0/0:ownerLR (admin-config-sdr:foo)# location 0/7/*
RP/0/0/0:ownerLR (admin-config-sdr:foo)# exit
RP/0/0/0:ownerLR (admin-config)# exit

Uncommitted changes found, commit them? [yes]:

RP/0/0/0:ownerLR (admin)# sh running
Building configuration...
sdr foo
  location 0/2/0
  location 0/6/0
  location 0/7/0
```

- Assignment is much like an OIR
- Card is removed (virtually) from physical router
- (owner SDR) and added to SDR foo Reload
DRP and DRP Pairing

- SDR should be carved using DRP Pair and Bunch of LC
- Provides software redundancy between 2 DRPs
- The two paired DRPs can be in any LC slot within or even across racks (CRS)
- Provides HA capability that can survive rack loss
- Pairing is through config only
DRP Pairing

```
RP/0/0/0:ownerLR (admin-config)# pairing <drp_pair_name>
RP/0/0/0:ownerLR (admin-config)# location 0/2/* 0/5/* ===== ➔ only wild card instances allowed

RP/0/0/0:ownerLR (admin)# show running
!
pairing <drp_pair_name>
location 0/2/* 0/5/*
!
RP/0/0/0:ownerLR (admin)# show redundancy summary | inc 0/2
  0/2/CPU0     0/5/CPU0     (Ready)
  0/2/CPU1     0/5/CPU1     (Ready)
```

- DRP pair will reset on commit of pairing config and will boot up as Active/Standby pair
- Switchover can be performed on drp-pair using “redundancy switchover location <primary-drp-node>” cli
Carving SDR

- Named SDR carved by allocating nodes to SDR through configuration
- After SDR is carved, all nodes assigned will reload and come up in new SDR

```bash
RP/0/RP0/CPU0:ios(admin-config)#sdr sdr1
RP/0/RP0/CPU0:ios(admin-config-lr:sdr1)#pair drp-pair primary ↔ DRP Pair
RP/0/RP0/CPU0:ios(admin-config-lr:sdr1)#location 0/2/* ↔ DRP
RP/0/RP0/CPU0:ios(admin-config-lr:sdr1)#location 0/3/* ↔ LC
RP/0/RP0/CPU0:ios(admin-config-lr:sdr1)#exit
RP/0/RP0/CPU0:ios(admin-config)#commit
```
Logging into SDR

- A newly carved SDR has no SDR scoped configuration
- This means that a local user has to be created before SDR can be used
- CRS owner (root-system user) logs in remotely first-time into SDR and creates local SDR users
  - Username: lab@admin
  - Password: lab
- To login remotely, remote authentication should be enabled on owner SDR
  - (admin-config) aaa authentication remote login local
- Root-system users cannot be created in SDR
- Highest privilege level for SDR users: root-lr

```
RP/0/RP0/CPU0:ios=admin)#show running-config
Building configuration...
sdr sdr1
  pair drp-pair primary
  location 0/0/*
!
sdr sdr2
  location 1/RP*/primary
```

```
RP/0/0/0:ownerLR#sh run
!
hostname ownerSDR
line console
  exec-timeout 0 0
!
service timestamps
interface Loopback0
  ipv4 address 192.168.1.1 255.255.255.0
```
### SDR Verification: “show platform “ (Admin/SDR Mode)

#### In admin mode, shows inventory of entire system

```
RP/0/RP0/CPU0:ios(admin)#show platform
```

<table>
<thead>
<tr>
<th>Node</th>
<th>Type</th>
<th>PLIM</th>
<th>State</th>
<th>Config State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0/SP</td>
<td>DRP(SP)</td>
<td>N/A</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/0/CPU0</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/0/CPU1</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/3/SP</td>
<td>DRP(SP)</td>
<td>N/A</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/3/CPU0</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/3/CPU1</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/RP0/CPU0</td>
<td>RP(Active)</td>
<td>N/A</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/RP1/CPU0</td>
<td>RP(Standby)</td>
<td>N/A</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
</tbody>
</table>

#### In SDR mode, shows SDR scoped inventory

```
DRP/0/0/CPU0:ios#show platform
```

<table>
<thead>
<tr>
<th>Node</th>
<th>Type</th>
<th>PLIM</th>
<th>State</th>
<th>Config State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0/CPU0</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/0/CPU1</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/3/CPU0</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>0/3/CPU1</td>
<td>DRP(Active)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>1/3/CPU0</td>
<td>DRP(Standby)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
<tr>
<td>1/3/CPU1</td>
<td>DRP(Standby)</td>
<td>DRP-ACC</td>
<td>IOS XR RUN</td>
<td>PWR,NSHUT,MON</td>
</tr>
</tbody>
</table>
SDR and Process Placement

- Placement Daemon process is responsible for placing placeable processes on available nodes
- Placement provides
  - Load Balance
  - Better Convergence
  - High Availability
- Process affinity configurable by
  - Node Set
  - Node type
  - User defined policy
- DRPs support process placement
- Process Placement only within SDR boundaries
Track 1: Harmonizing the Customer Experience

Northbound interfaces

- Redundancy minimized & mapped
- Inconsistency identified & eliminated

Consistent layered interfaces specified

Protocols & Services

Common Northbound Mgmt / APIs

Component A

Component B

Human
CLI
HTML

Machine
REST
SNMP
Netflow

Common component functions are invoked independent of accessing API
Technologies Driving Management Convergence

**APIs**
- C, JAVA, REST Program
- API Presentation
- API Infrastructure
- Platform

**Manageability**
- Common Presentation
- SDK
- NETCONF
- Switch Router
- YANG

**Programmability**
- Orchestration or Control
- Controller
- Switch

**Consistent cross-platform interface for programmers**
- Innovate new features quickly
- Deploy cross-platform
- Leverage apps from developer’s community
- Put services closer to your users

**Monitor and control network operation and data**
- Consistent interfaces and standards across all platforms
- Standards-based information models
- Alternative to CLIs and screen-scraping

**Enable flow control and orchestration via open protocols**
- Program virtualized network resources
- Provide logical separation of control and data plane
- Abstract network functions such as security or routing
Track 2: Expand Infrastructure for Software Sharing

- **IOS & IOS XE**
  - Virtualized Network Operating System
  - Virtualized Services (vNIC, FAL)
  - Legacy Forwarding Infra, SW Drivers
  - Platform Specific HW
  - Shared Infra

- **IOS XR**
  - Virtualized Network Operating System
  - Virtualized Services (vNIC, FAL)
  - Common Forwarding Infra, Shared SW Drivers
  - Shared/Custom Hardware
  - Shared Infra

- **NX OS**
  - Virtualized Network Operating System
  - Virtualized Services
  - Admin Plane, VDC Mgmt.
  - Virtualization Services
  - Shared Infra

Areas of Investment:
- Virtualized Services (vNIC, FAL)
- Common Forwarding Infra, Shared SW Drivers
- Shared/Custom Hardware
- Platform Infra (Chassis + IO + Environmental)
- Linux Kernel + Distribution
- Processor Complex
Track 3: Drive Common Networking Protocols and Applications

Each product in the IOS Software Family is built from a common set of Networking technologies and innovations
Key Takeaways
Cisco’s Network Software Strategy

Harmonize the User Experience
Common Human Interfaces for all products
Common Machine Interfaces for all Products
Common Software Release Strategies

Infrastructure for Software Sharing
Virtualization Technologies for leveraging existing Software
Evolved Application Runtime Environment
Uniform Baseline

Common Protocols and Applications
Shared components: consistency across all platforms
Standardized releases across platforms
Integrated Software based services