
cisco
Cisco live!
July 10–14, 2016 · Las Vegas, NV
Your Time Is Now

Intro to IOS XR for Enterprises and Service Providers

Brad Edgeworth, CCIE#31574, Systems Engineer
@BradEdgeworth

BRKARC-1008

Cisco live!

Introduction

Housekeeping

- Make sure you are in the correct session: BRKARC-1008
- Please silence cell phones
- If you are following along with the PDF version.

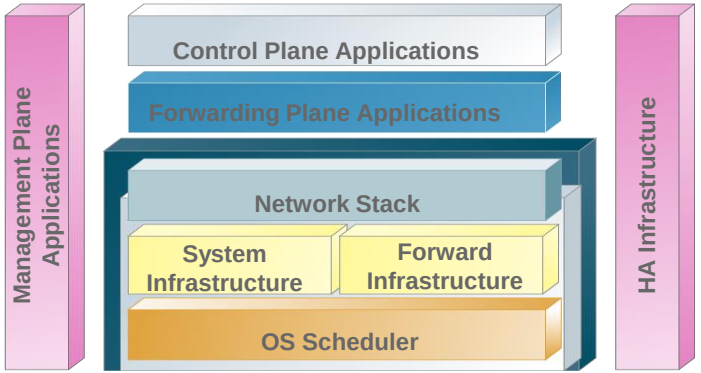
Indicates that this is a reference slide.
(Will not be covered in the presentation).

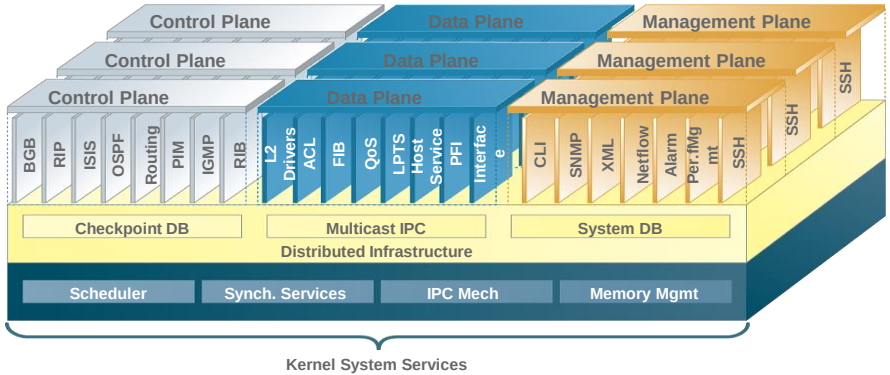
Agenda

- Introduction
- IOS XR Architecture
  - Modularity, Scalability, Stability, Security, Software Package
  - Classic vs XR 6.0
- Command Line Interface
- Configuration Management
- Monitoring and Troubleshooting tools
- Example Configurations
- Route Policy Language (RPL)

IOS XR Architecture :
Modularity & Software Packages

Evolution of Router OS





IOS

- Monolithic Kernel
- Run to Completion Scheduler
- Centralized Infrastructure
- Centralized applications
- Everything has hardware access

IOS XR

- Micro Kernel
- Preemptive Multitasking
- Distributed Infrastructure
- Distributed applications
- Limited access to hardware

Established 2004

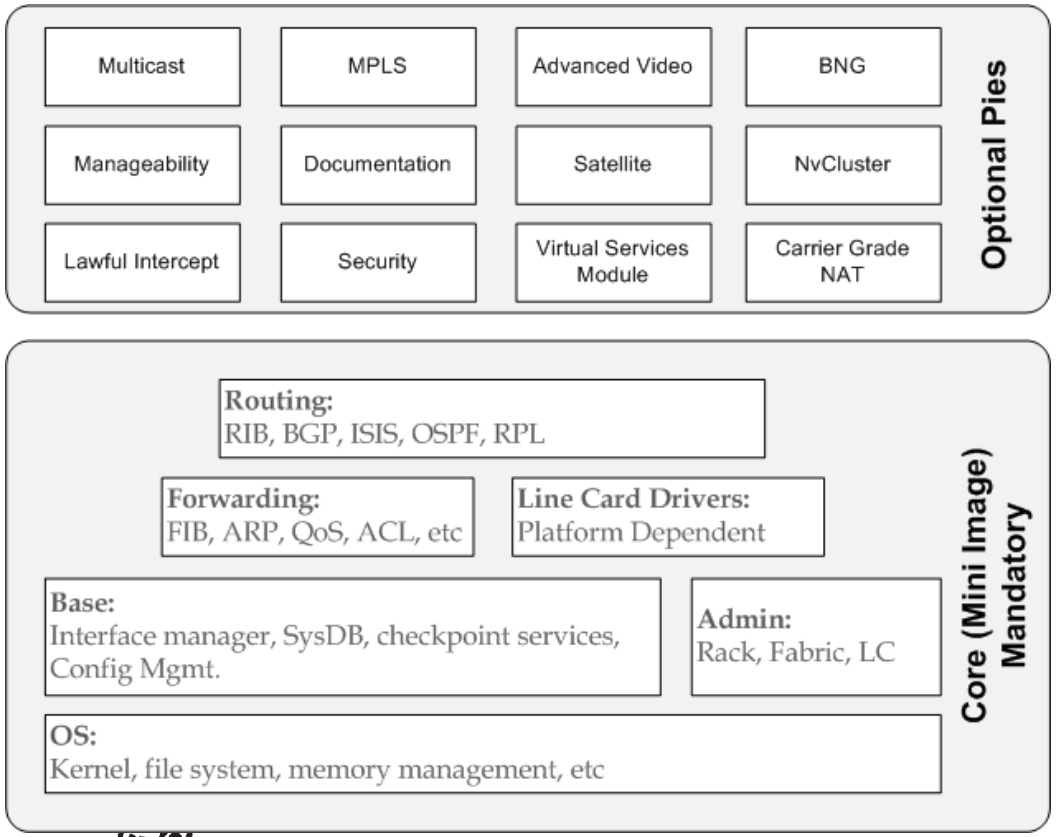
IOS XR's Software Packaging

- Does not use a single file. Uses multiple DLLs that make up a software feature.
- DLLs provide a mechanism for memory sharing and are loaded & unloaded as needed.
- Package Installation Envelopes (PIEs) are a delivery mechanism for packages.
- The *'mini'* is the core composite package that contains mandatory software to boot IOS XR

PIE?

Mini?

IOS-XR Software Packages



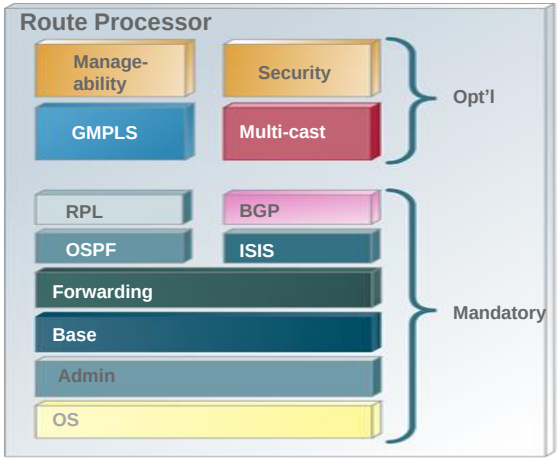
IOS XR Mini:

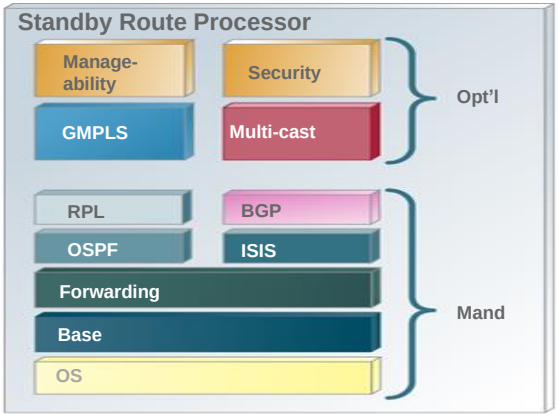
- **Includes the following components:**
  - **OS**
  - **Base**
  - **Admin**
  - **Line Card Drivers**
  - **Forwarding Logic**
  - **Routing Protocols**

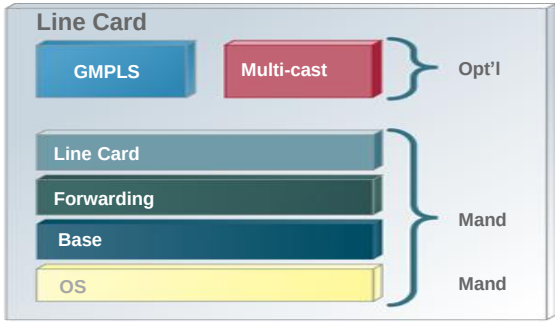
IOS XR Versioning

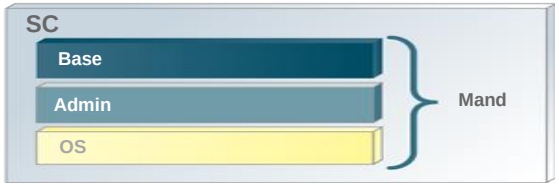
- **Major release**
  New functionality (4.2, 4.3, 5.1, 5.2, etc.)
- **Maintenance release**
  SW fixes (5.2.0, 5.2.1, 5.2.2)
- There are not any special S, T, J, or XT trains; special functionality is added through packages.

IOS XR Modular Packaged Software







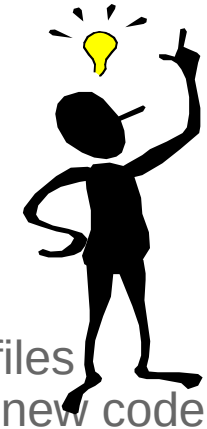


Software Maintenance Upgrade (SMUs)

- Patches for urgent issues for a specific code release and platform
- Fix integrated into the subsequent IOS XR maintenance release.
- Corrects defects only. No interface changes (no changes to CLI, APIs, IPC etc.) or new feature content. Also shrinks your (customer) validation time because only that feature needs to be validated
- SMU is named by release and bugid - Examples - **hfr-px-5.2.1.CSCue55783.pie**

SMU?

PIE Installation Concepts

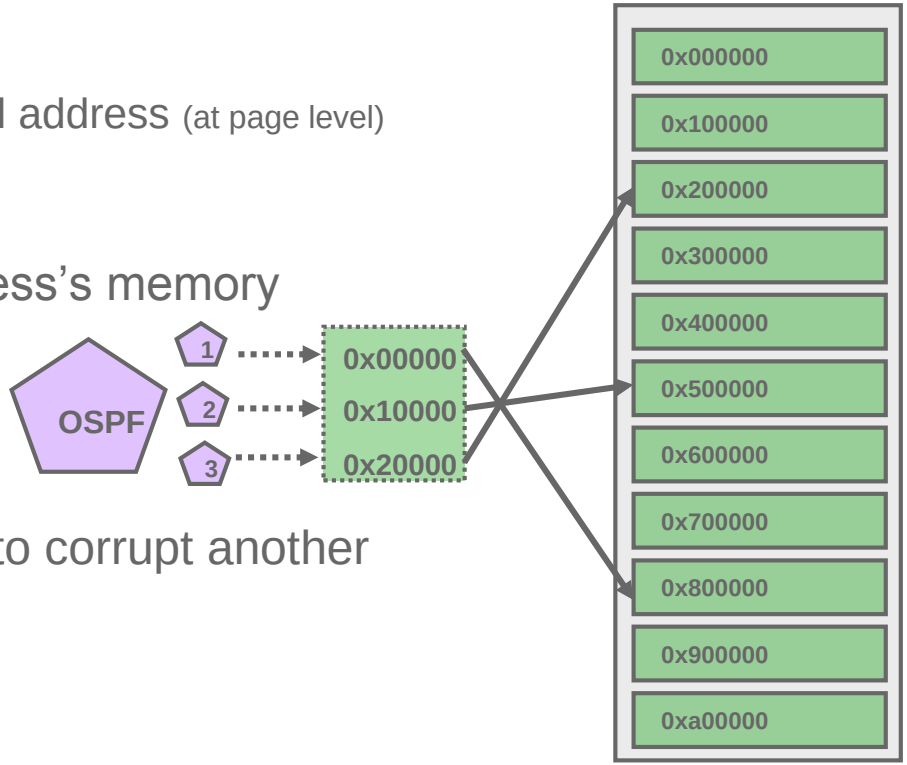


- PIEs can be added, upgraded, or removed.
- Three phases of PIE installation
  - **Add** – Copies the package file to the local storage and unpacks the files
    Activate – Installs the new code by restarting processes/nodes with new code
    Commit – Locks the activated code to sustain reloads
- Three phases of removing a PIE
  - **DeActivate** – Removes the code from runtime
  - **Commit** – Locks the code change to sustain reloads
  - **Delete** – Removes unused packages from the local storage

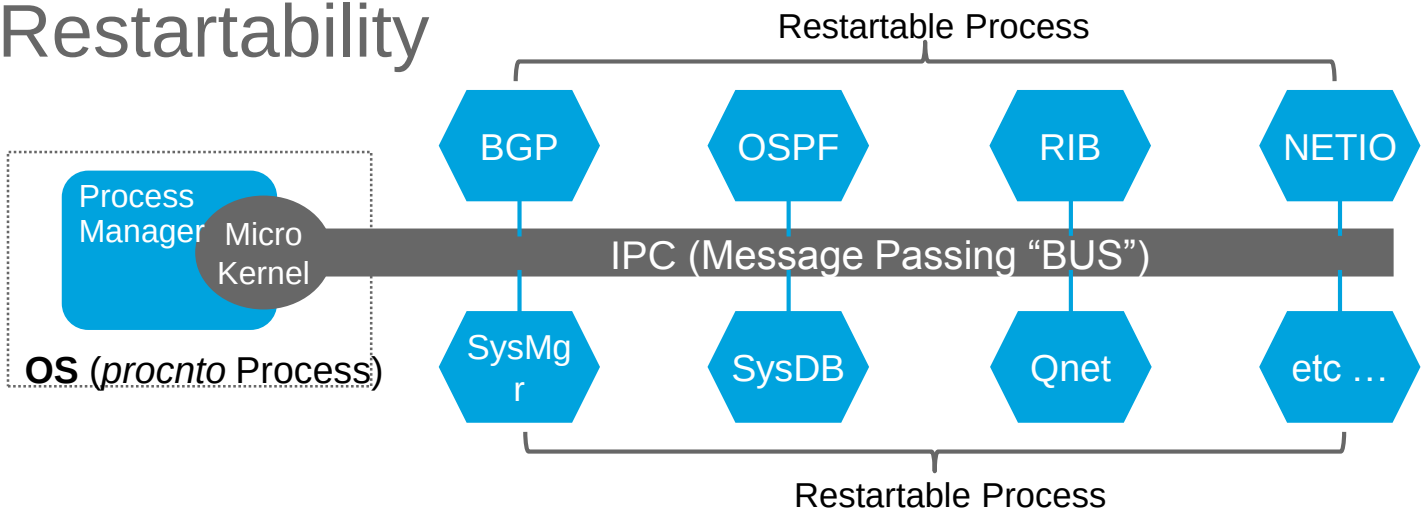
IOS XR Architecture : Stability

Protected Memory Space For Processes

- 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 process's memory
  - Process can only access virtual space
- IOS processes share same memory space through aliasing. Possible for one process to corrupt another process's memory

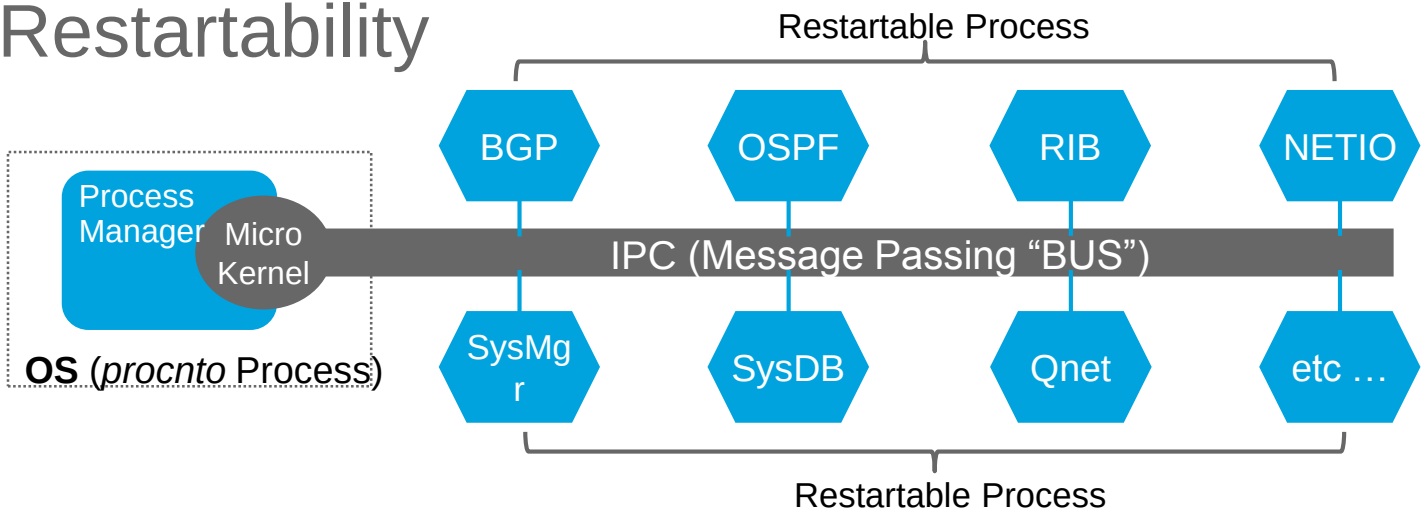


Process Restartability



- Used for containing small faults or process failures
- All processes outside the microkernel process manager are individually restartable.
- If any of the processes, including SysMgr, SysDB, BGP, or Qnet, is restarted it does not cause the entire system to reload.
- Certain processes are '**mandatory**' – must always be running. Failure of mandatory processes can cause RP failover

Process Restartability



- Processes support restarting with dynamic state recovery
- Processes store will try to store information in a peer process during normal runtime.
- In the event of a process restarts, it recovers its states from checkpoint of persistent storage or peer processes.

IOS XR Architecture : Security

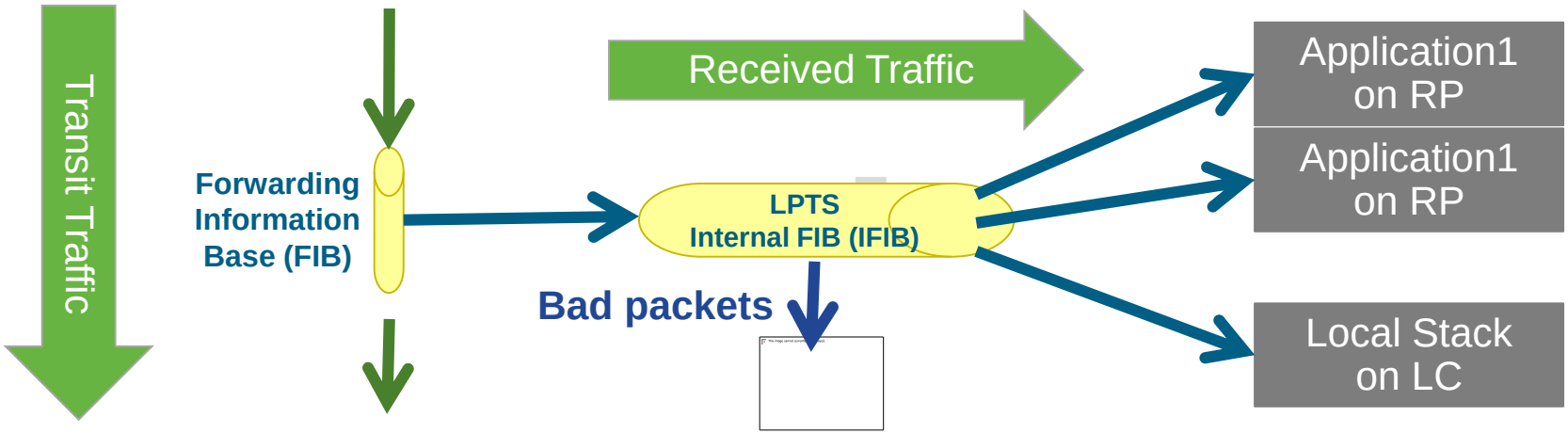
Local Packet Transport Services (LPTS)

- Equivalent to CoPP but MUCH better
- Responsible for delivery of data destined for a router.
- Does not apply to transit or IPC traffic
- Enables delivery of data to distributed processes across the system hardware (RPs, LCs)

Interoffice Mail for Data Plane

LC/RP CPU guard

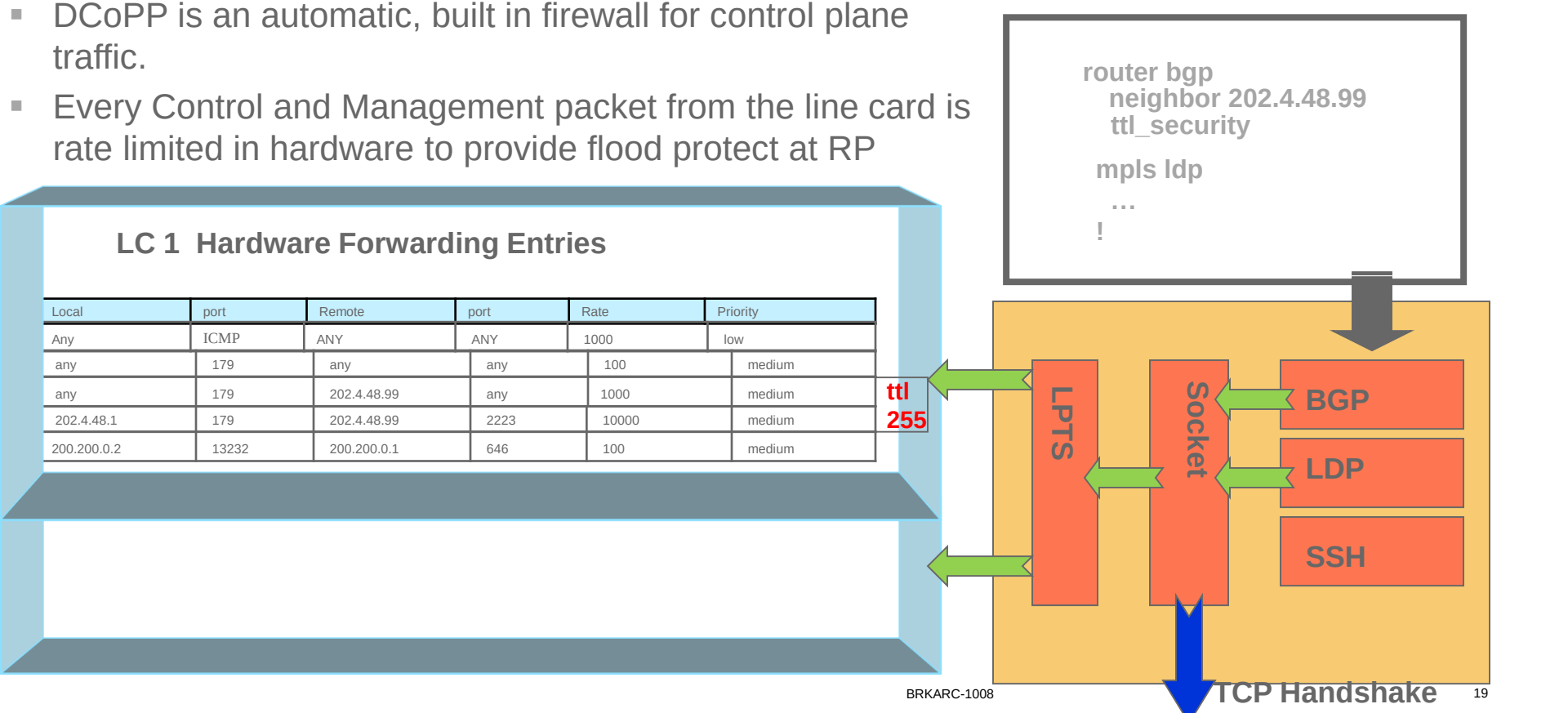
Local Packet Transport Services (LPTS)



- LPTS is transparent and automatic
- LPTS acts as an dynamic internal firewall to protects router resources
  - Rules are dynamically built based upon control plane flows
- Packet forwarding executed in HW - no impact on Line Card CPU
  - Traffic can be rate limited by hardware

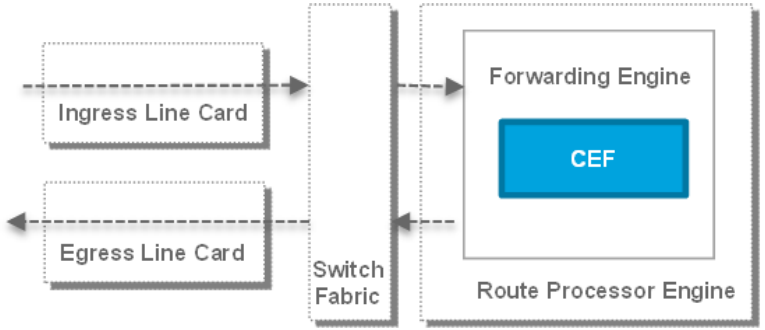
LPTS: Dynamic Control Plane Protection

- 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 protect at RP



IOS XR Architecture : Scalability

Router Forwarding Architectures: Centralized



Not Very Efficient or Scalable!

- Forwarding Engine (FE) resides on the route processor (RP)
- RP is responsible for running the routing protocols and learning the routes. RP programs the FE with next-hop IP address.
- FE identifies the MAC address & outbound interface for the next-hop IP. FE performs the MAC address rewrite
- Packets received on a line card are forwarded to the forwarding engine regardless of outbound line card (LC)

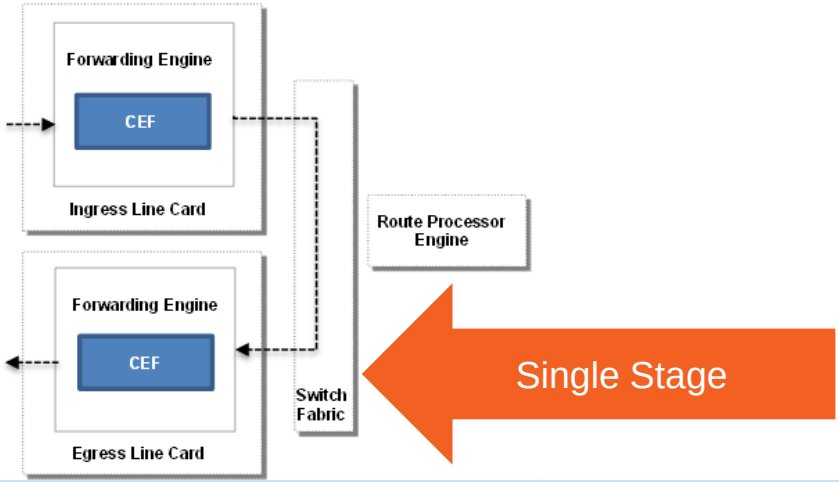
Router Forwarding Architectures: Single-Stage Distributed

LC1 Forwarding Engine Table

Destination Network	Out Int	Next-Hop IP	Next-Hop MAC	Outbound MAC
10.10.10.0/24	Gi 2/0	10.1.1.1	00:10:00:C1:5C:01	00:10:00:00:11:01
10.20.20.0/24	Gi 1/0	10.2.2.2	00:10:00:C1:5C:02	00:10:00:00:11:02

LC2 Forwarding Engine Table

Destination Network	Out Int	Next-Hop IP	Next-Hop MAC	Outbound MAC
10.10.10.0/24	Gi 2/0	10.1.1.1	00:10:00:C1:5C:01	00:10:00:00:11:01
10.20.20.0/24	Gi 1/0	10.2.2.2	00:10:00:C1:5C:02	00:10:00:00:11:02



- Route processor programs forwarding engine based upon routing protocols
- Forwarding Engine (FE) resides on the line cards. Contains all the routes, next-hops, and outbound IP addresses *'on the router'*
- Ingress line card performs destination lookup.
- Packets are sent out on the fabric only if the egress port is on a different LC. Packets are locally forwarded if the ingress and egress ports are on the same LC.

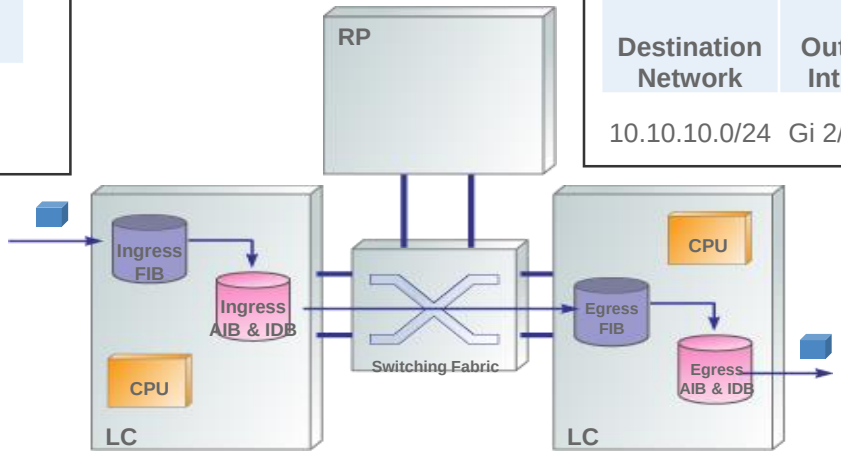
Router Forwarding Architectures: Two-Stage Distributed

LC1 Ingress FE Table

Destination Network	Out LC
10.10.10.0/24	2
10.20.20.0/24	1

LC2 Egress FE Table

Destination Network	Out Int	Next-Hop IP Address	Next-Hop MAC	Outbound MAC
10.10.10.0/24	Gi 2/0	10.1.1.1	00:10:00:C1:5C:01	00:10:00:00:11:01

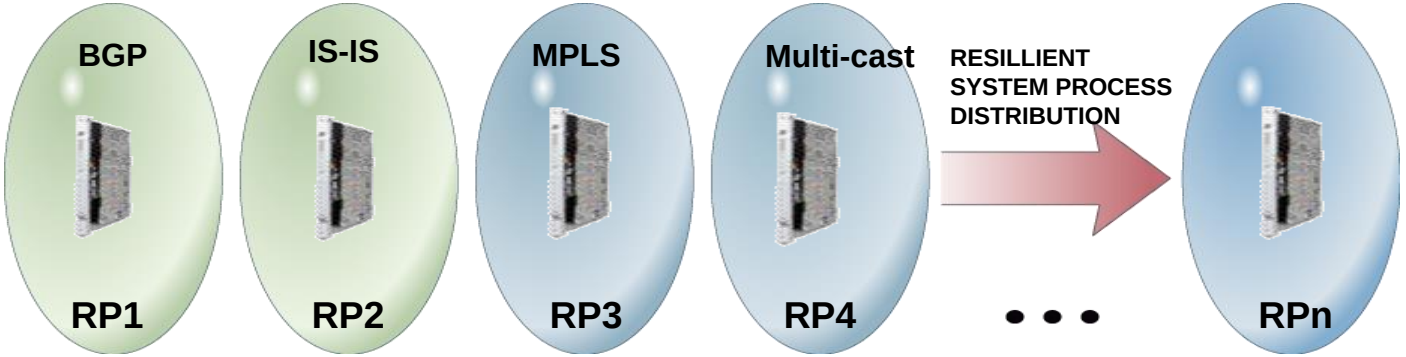


Required for IOS XR

Two stage forwarding(Distributed)

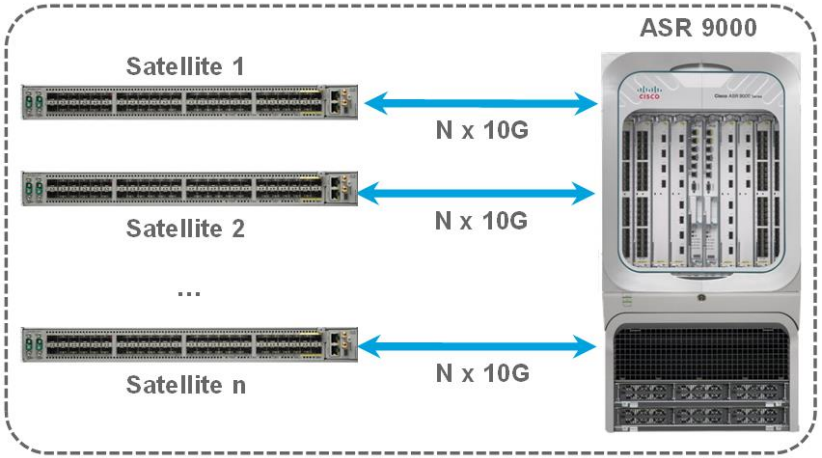
- Each line card has two forwarding engines: Ingress and Egress
- Ingress forwarding engine contains destination networks and destination LC
- Egress forwarding engine maintains AIB & IDB for networks attached to it. Egress FE Identifies next-hop, re-writes MAC address, etc

Distributed Control Plane



- Routing protocols and signaling protocols can run on one or more Route Processors (RPs)
- Redundancy comes with a standby RP

nV Satellite



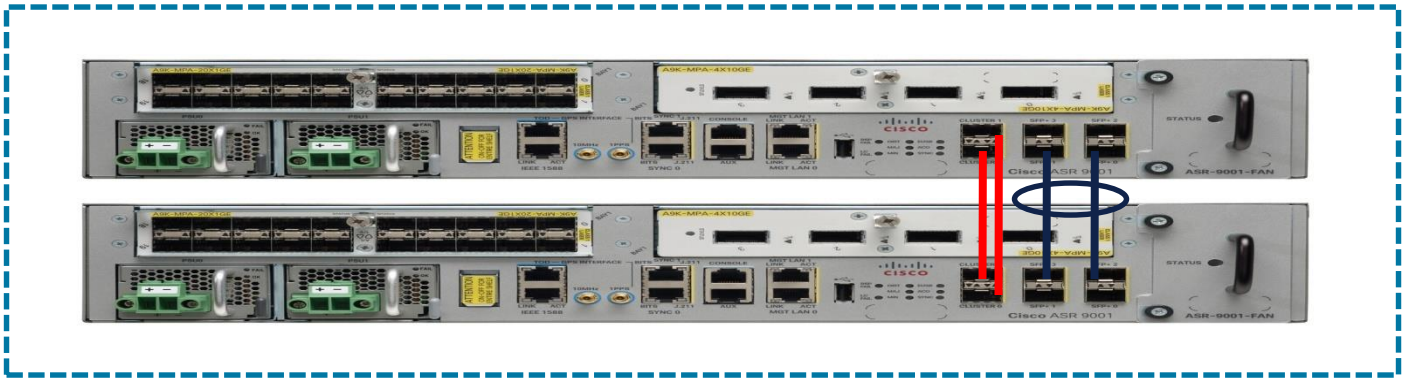
One Virtual System

Satellite chassis act like remote line cards. Distance can be measured in KM, not M.

One centralized management for a large geographical area

Service providers use this technology to reduce truck rolls

nV Cluster



nV Cluster (1 Logical Router)

Two physical chassis

Provides redundancy in control plane

One logical management; reduces management overhead

Similar to VSS

IOS XR 6.0 vs IOS XR Classic

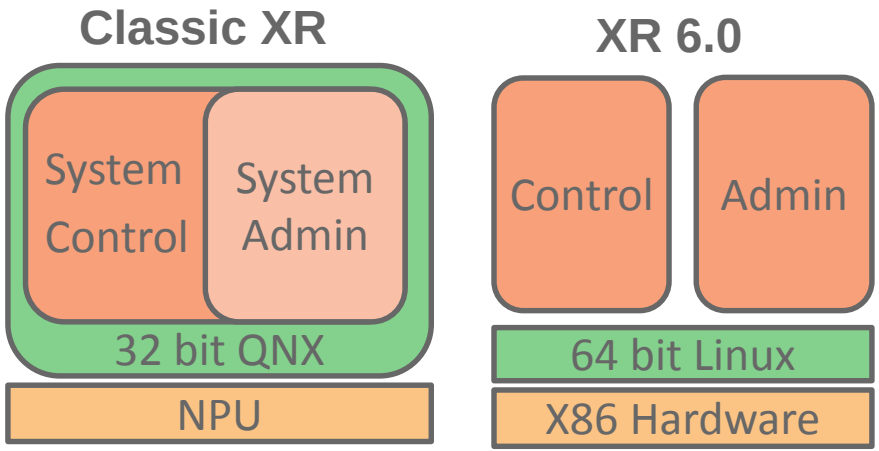
IOS-XR 6.0: A New Software Infrastructure

64-bit OpenEmbedded Linux support.

- Processes containerization.
- Allows future third-Party applications support.

Process containerization provides:

- Isolation
- Allows future third-party applications
- More efficient than running VMs
- Allows more efficient method of patching software versions via ISSU



Installing and Updating Third Party Packages

- Third Party Packages are traditional Linux tools available from the Shell
  - Communication: lighttpd ,openssh, wget, curl, etc.
  - Programing: python, ruby, perl, etc.
  - Utilities: sed, gawk, tar, gzip, vi, etc.
- Additional packages provided by vendors (No Cisco Support)
  - Chef
  - Puppet
- Installed using yum or rpm

```
yum-config-manager --add-repo=http://192.168.0.254/XR/6.0.0
yum install chef –y
```

Classic IOS XR PIE Install

- No dependency management
- Offline process required to copy packages
- Require multiple operations
  - install add
  - install activate
  - install commit
- CSM for package content

IOS XR 6.0 RPM Install

- Dependency management
- Online process over secure transport
- Single operation
  - install update or install upgrade
- On-box / Off-box package inspection using rpm tool
  - Description
  - Dependencies
  - Content

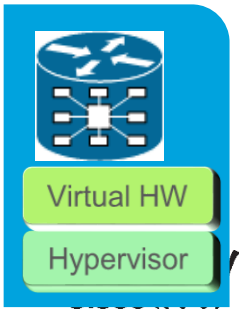
IOS XR Devices

IOS XR Devices

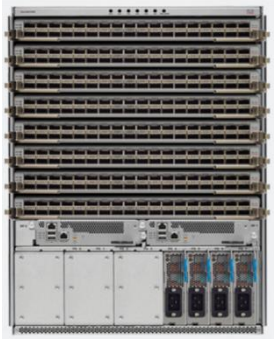
Carrier Routing System (CRS)

Network Convergence System (NCS)

Aggregation Services Router (ASR) 9000



IOS XRv 9000



IOS XR
Command Line Interface

CLI Modes

There is no user mode prompt (Like in IOS)

```
IOS>
```

Executive mode

(Protocol Configuration)

```
RP/0/0/CPU0:IOS-XR#config t
RP/0/0/CPU0:IOS-XR(config)#
```

Admin Mode

(Platform Power and Software Version)
(Package Installation/Upgrade/Removal)

```
RP/0/0/CPU0:IOS-XR#admin
RP/0/0/CPU0:IOS-XR(admin)#
```

CLI Parsing Utilities

- Supports multiple parsing utilities to reduce output

```
RP/0/RP0/CPU0:CRS-D#show run | ?
  begin    Begin with the line that matches
  exclude  Exclude lines that match
  file     Save the configuration
  include  Include lines that match
  utility  A set of common unix utilities
  <cr>     Shows current operating configuration
```

There is not a '*Section*'

More on this later…..

CLI Parsing Utilities (continued)

- Multiple Unix parsing utilities.

```
RP/0/RP0/CPU0:CRS-D#show run | utility ?
  cut     Cut out selected fields of each line of a file
  egrep   Extended regular expression grep
  fgrep   Fixed string expression grep
  head    Show set of lines/characters from the top of a file
  less    Fixed string pattern matching
  more    Paging Utility More
  script  Launch a script for post processing
  sort    Sort, merge, or sequence-check text files
  tail    Copy the last part of files
  uniq    Report or filter out repeated lines in a file
  wc      Counting lines/words/characters of a file
  xargs   Construct argument list(s) and invoke a program
```

RegEx Queries

- Multiple keywords requires the use of quotes “ and separation of terms by the pipe |

```
RP/0/RP0/CPU0:CRS1#show processes  cpu | exclude  0%
CPU utilization for one minute: 2%; five minutes: 2%; fifteen minutes: 2%

PID    1Min    5Min    15Min Process
131105   1%      1%       1% ce_switch
131106   1%      1%       1% eth_server

RP/0/RP0/CPU0:CRS1#show processes cpu | exclude " 0%|ce_"
CPU utilization for one minute: 2%; five minutes: 2%; fifteen minutes: 2%

PID    1Min    5Min    15Min Process
131106   1%      1%       1% eth_server
```

Multi pipe support (continued)

- Support multiple pipes on the command line so that the output can be processed by multiple parsers with the output of any show command.
- Up to 8 pipes are supported.
- Support for Include, Exclude and Regex all at the same time ☺

```
RP/0/RP0/CPU0:CRS1#show log start Jan 3 07:00:00 | in LDP | in " UP|DOWN" | ex "10.2[0-5]"
RP/6/RP0/CPU0:Jan 3 17:10:18: mpls_ldp[1038]: %ROUTING-LDP-5-NBR_CHANGE : Neighbor 10.1.1.1:0, DOWN
       (Interface state down)
RP/6/RP0/CPU0:Jan 3 17:10:58: mpls_ldp[1038]: %ROUTING-LDP-5-NBR_CHANGE : Neighbor 10.1.1.1:0, UP
```

IOS-XR Interface Format

- New CLI reflects the HW position in the system
  - Introduces the Hierarchical location scheme
  - Interfaces have the Rack/Shelf/(Slot/Bay)/Interface scheme
- Protocol referenced by address family type – v4/v6
- Command-set identical or similar to IOS

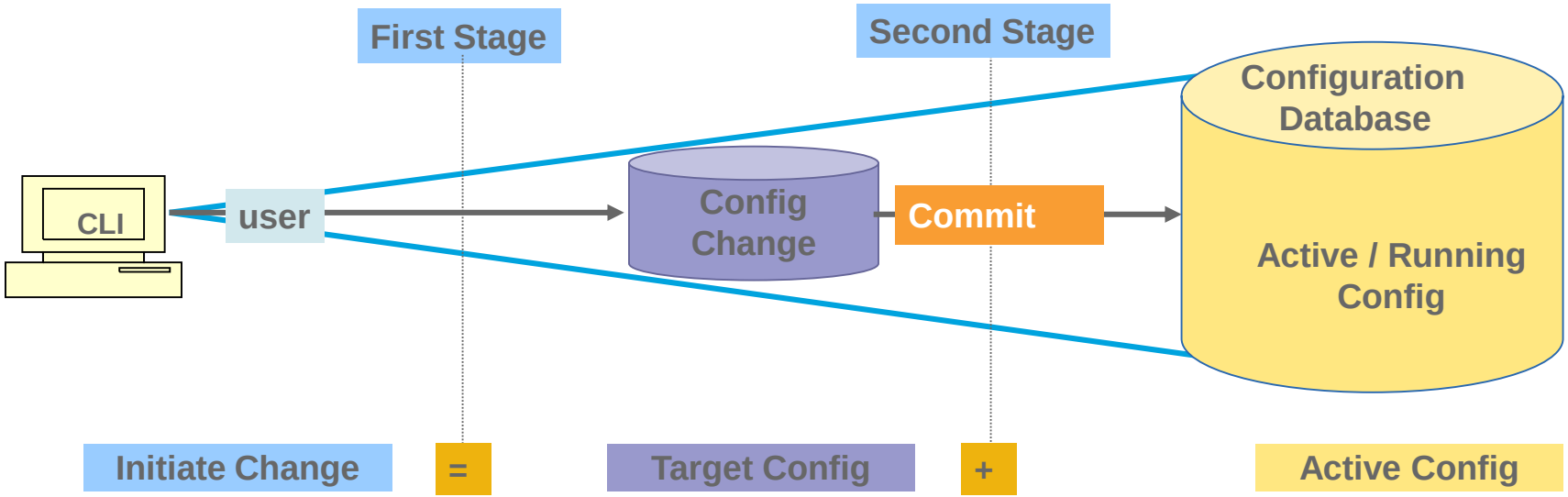
```
RP/0/0/CPU0:CRS1#show ipv4 interface brief

Interface                      IP-Address      Status          Protocol
MgmtEth0/0/CPU0/0              10.23.1.69      Up              Up
MgmtEth0/0/CPU0/1              unassigned      Shutdown        Down
GigabitEthernet0/2/0/0         100.12.1.1      Up              Up
```

IOS XR Configuration Management

Differences in IOS XR's Configuration Management

- IOS-XR's configuration is held in a System DB (SysDB)
  There is not a startup-config
- Router configuration is based on two stage configuration model.
  The "***running***" or "***active***" configuration can not be modified directly.
- User makes changes to a ***Target Configuration*** for the staging of all the changes.
- Order of Operations is not important. 1st stage to prevent router lock-outs.
- Supports pre-configuration of hardware
- The Target Configuration must be explicitly **Commit**ed to the active configuration (second stage) which applies the changes to the running-config.

IOS-XR CLI: Two Stage Configuration Model



User establishes config session

Adds/deletes/modifies configuration; these changes:

- Are entered in the staging area
- Are validated for syntax and authorized
- Can be reviewed and modified

Promotes the changes to active configuration; these changes:

- Are verified for semantic correctness
- Are check-pointed on the router

Target Configuration Changes

- Syntax is parsed for CLI syntax only
- The output for the command **show configuration** is dependent upon the CLI mode it is entered

```
RP/0/0/CPU0:ios#conf terminal
RP/0/0/CPU0:ios(config)#hostname XR1
RP/0/0/CPU0:ios(config)#cdp
RP/0/0/CPU0:ios(config)#int gigabitEthernet 0/0/0/0
RP/0/0/CPU0:ios(config-if)#cdp
RP/0/0/CPU0:ios(config-if)#int gigabitEthernet 0/0/0/1
RP/0/0/CPU0:ios(config-if)#cdp

RP/0/0/CPU0:ios(config-if)#show configuration
Building configuration...
hostname XR1
cdp
interface GigabitEthernet0/0/0/0
 cdp
!
interface GigabitEthernet0/0/0/1
 cdp
!
End
```

Nothing has been processed to the running-configuration

This is the target configuration

Commit'ng the Change

- The target configuration is checked for validity. It is applied all at once to the running-configuration.
- If an invalid configuration is found, the commit will fail.
- All successful commits are given a Commit-ID and is stored in the SysDB.

```
RP/0/0/CPU0:ios(config-if)#commit
RP/0/0/CPU0:Sep 21 00:26:25.360 : config[66391]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
committed by user 'JCHAMBR'.  Use 'show configuration commit changes 1000000638' to view
the changes.
RP/0/0/CPU0:XR1(config-if)#
```

IOS-XR CLI: Config error handling

- **Parser/Syntax error**
  - Identified by the parser when the <*return*> key is entered
- **Commit error**
  - None of the configuration is applied to the running-configuration
  - Syntactically correct but 'invalid' from configuration commit standpoint
  - Error details viewed through "**show configuration failed**" command

```
RP/0/0/CPU0:CRS1#configuration term
RP/0/0/CPU0:CRS1(config)#policy p1
RP/0/0/CPU0:CRS1(config-pmap)#class c0
RP/0/0/CPU0:CRS1(config-pmap-c)#set precedence 0
RP/0/0/CPU0:CRS1(config-pmap-c)#
RP/0/0/CPU0:CRS1(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)#
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
```

Viewing of List of Commit-IDs in SysDB

```
RP/0/0/CPU0:XR1#show configuration commit list
No. Label/ID    User      Line                Client   Time Stamp
~~~~ ~~~~~~~~    ~~~~      ~~~~                 ~~~~~~  ~~~~~~~~~~
1    1000000038  CROBBIN vty3:node0_0_CPU0   CLI      Fri May 13 11:06:35 2015
2    1000000037  KJOHNS   vty3:node0_0_CPU0   CLI      Fri May 13 11:05:33 2015
3    1000000036  BEDGEW   vty3:node0_0_CPU0   CLI      Fri May 13 11:00:41 2015
4    1000000035  MOALI    vty3:node0_0_CPU0   CLI      Fri May 13 10:59:39 2015
5    1000000034  CROBBIN  vty3:node0_0_CPU0   CLI      Tue Apr 27 15:08:04 2015
6    1000000033  KJOHNS   vty1:node0_0_CPU0   CLI      Tue Mar 16 15:32:27 2015
7    1000000032  MOALI    vty3:node0_0_CPU0   CLI      Mon Mar 15 16:22:54 2015
8    1000000031  BEDGEW   vty3:node0_0_CPU0   CLI      Mon Mar 15 16:21:14 2015
```

Viewing List of Configuration Changes in SysDB

- Allows you to view the changes occurred over a period of time or a specific commit-id

```
RP/0/0/CPU0:XR1#show configuration commit changes 1000000025
Building configuration...
!
no route-policy RPL-L3-IPv4-IN-BETA
end
```

```
RP/0/0/CPU0:XR1#show configuration commit changes last 3
Building configuration...
no cdp
!
no interface Loopback0
!
no router ospf 1
end
```

Configuration Rollback

- Allows the configuration to be rolled back a # of changes or restores the configuration to a specific point of time.
- Configuration rollback is considered a change.
- It is possible to view the list of changes being made before issuing a rollback

```
RP/0/0/CPU0:XR1-COMMITREPLACE#rollback configuration last 3
Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing..
10 items committed in 2 sec (4)items/sec
Updating.
Updated Commit database in 1 sec
Configuration successfully rolled back 3 commits.
RP/0/0/CPU0:XR1#
RP/0/0/CPU0:XR1#show configuration commit list
SNo. Label/ID    User      Line              Client      Time Stamp
~~~~ ~~~~~~~~    ~~~~      ~~~~              ~~~~~~      ~~~~~~~~~~
1    1000000021  JCHAMBR   con0_0_CPU0       Rollback    Fri May 20 16:37:10 2015
2    1000000020  JCHAMBR   con0_0_CPU0       CLI         Fri May 20 16:08:57 2015
```

Commit Confirmed

Packet Loss for 'Commit Confirmed' is a lot less than a 'Reload In'

- 1st Commit is a Trial.
  Initiates an automatic rollback if the trial commit is not committed a 2nd time.
- Ensures that a change does as it is supposed to.
- **Helps prevent lockouts!!!**

```
RP/0/0/CPU0:XR1(config)#hostname XR1-COMMIT-CONFIRM
RP/0/0/CPU0:XR1(config)#commit confirmed 30
RP/0/0/CPU0:Sep 16 13:46:53.374 : config[66625]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
  committed by user 'BEDGEW'. Use 'show configuration commit changes 1000000042' to view
  the changes.
RP/0/0/CPU0:XR1-COMMIT-CONFIRM(config)#
RP/0/0/CPU0:Sep 16 13:47:24.075 : cfgmgr_trial_confirm[66653]: %MGBL-CONFIG-6-DB_COMMIT :
  Configuration committed by user 'BEDGEW'. Use 'show configuration commit changes
  1000000043' to view the changes.
RP/0/0/CPU0:XR1(config)#

RP/0/0/CPU0:XR1#show configuration commit list
Mon May 16 13:59:44.908 EDT
SNo. Label/ID    User      Line                Client      Time Stamp
~~~~ ~~~~~~~~    ~~~~      ~~~~                ~~~~~~      ~~~~~~~~~~
1    1000000043  BEDGEW    vty3:node0_0_CPU0   Rollback    Mon May 16 13:47:23 2015
2    1000000042  BEDGEW    vty3:node0_0_CPU0   CLI         Mon May 16 13:46:53 2015
```

Commit Confirmed

- 2nd Commit does not register as a change

```
RP/0/0/CPU0:XR1(config)#hostname XR1-COMMIT-CONFIRM
RP/0/0/CPU0:XR1(config)#commit confirmed 30
RP/0/0/CPU0:Sep 16 13:51:47.414 : config[66850]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
  committed by user 'BEDGEW'. Use 'show configuration commit changes 1000000044' to view
  the changes.
RP/0/0/CPU0:XR1-COMMIT-CONFIRM(config)#
RP/0/0/CPU0:XR1-COMMIT-CONFIRM(config)#commit
% Confirming commit for trial session.
RP/0/0/CPU0:XR1-COMMIT-CONFIRM(config)#exit
RP/0/0/CPU0:XR1-COMMIT-CONFIRM#
RP/0/0/CPU0:XR1-COMMIT-CONFIRM#show configuration commit list
SNo. Label/ID    User      Line                Client      Time Stamp
~~~~ ~~~~~~~~    ~~~~      ~~~~                ~~~~~~      ~~~~~~~~~~
1    1000000044  BEDGEW    vty3:node0_0_CPU0   CLI       Mon May 16 13:51:47 2015
2    1000000043  BEDGEW    vty3:node0_0_CPU0   Rollback  Mon May 16 13:47:23 2015
3    1000000042  BEDGEW    vty3:node0_0_CPU0   CLI       Mon May 16 13:46:53 2015
```

Time for a Break!
Questions about Configurations?

IOS XR Operation and Monitoring Tools

Trace functionality

- Trace functionality is a form of '*always-on*' debug without performance hit
- Circular logging
- Built-In to almost every component of IOS XR

```
RP/0/RP1/CPU0:CRS1#show ospf trace
 OSPF Trace Summary (2, RP/1/RP0/CPU0:CRS1, 0M)

    Trace Name      Size      Count  Description
    ------------ ------- ----------  -------------------------
 1. adj            65536       6291  adjacency
 2. adj_cycle      65536     893383  dbd/flood events/pkts
 3. config          2048        486  config events
 4. errors          8192     868816  errors
 5. events          4096        255  mda/rtrid/bfd/vrf
 6. ha              8192        485  startup/HA/NSF
 7. hello           2048    3982447  hello events/pkts
 8. idb             8192        973  interface
 9. pkt             2048    1927767  I/O packets
10. rib            65536      52190  rib batching
11. spf            65536      93138  spf/topology
12. spf_cycle      65536     352143  spf/topology detail
13. te              4096       3893  mpls-te
14. test            1024      20052  testing info
15. mq             65536          5  message queue info
```

Trace functionality (continued)

```
RP/0/RP0/CPU0:CRS1#show ospf trace hello
Traces for OSPF 2 (Wed Jan 22 08:55:38)
Traces returned/requested/available:  2048/2048/2048
Trace buffer: hello

1    Jan 22 08:49:45.305* ospf_send_hello: area 0.0.0.80 intf MADJ: BE1008 from 0.0.0.0
2    Jan 22 08:49:45.546  ospf_rcv_hello: intf BE1009 area 0.0.0.74 from 10.1.0.9 10.1.9.2
3    Jan 22 08:49:45.546  ospf_check_hello_events: intf MADJ: BE1009 area 0.0.0.74 from 0.0.0.0
4    Jan 22 08:49:45.573* ospf_send_hello: area 0.0.0.74 intf MADJ: BE1008 from 0.0.0.0
5    Jan 22 08:49:45.845* ospf_rcv_hello: intf BE1009 area 0.0.0.80 from 10.1.0.9 10.1.9.2
6    Jan 22 08:49:45.845* ospf_check_hello_events: intf MADJ: BE1009 area 0.0.0.80 from 0.0.0.0
7    Jan 22 08:49:45.917* ospf_send_hello: area 0.0.0.80 intf Te0/5/0/7 from 10.1.80.1
8    Jan 22 08:49:46.232  ospf_rcv_hello: intf BE1008 area 0.0.0.74 from 10.1.0.8 10.1.8.2
```

‘Monitor interface’ command

Monitoring Time is shown.
Great for differentiating output if reviewing router session logs later

```
RP/0/RP1/CPU0:CRS1#monitor interface

CRS1-CRS                Monitor Time: 00:00:37          SysUptime: 245:59:24

Protocol:General
Interface               In(bps)       Out(bps)      InBytes/Delta  OutBytes/Delta
Mg1/RP0/CPU0/0           1000/  0%         0/  0%    173.8M/954       79086/0
Te1/6/0/0                   0/  0%         0/  0%         0/0             0/0
Te1/6/0/1                7000/  0%      2000/  0%    748.4M/479      276.0M/478
Te1/6/0/2                   0/  0%         0/  0%         0/0             0/0
--- snip
Gi1/15/3/5                  0/  0%         0/  0%         0/0             0/0
Gi1/15/3/6                  0/  0%         0/  0%         0/0             0/0

Quit='q',     Clear='c',    Freeze='f', Thaw='t',
Next set='n', Prev set='p', Bytes='y',  Packets='k'
(General='g', IPv4 Uni='4u', IPv4 Multi='4m', IPv6 Uni='6u', IPv6 Multi='6m')
```

‘Monitor interface *int-type int-number*’ command

```
RP/0/RP1/CPU0:CRS1#monitor interface Bundle-ether 1008

CRS1                Monitor Time: 00:00:18          SysUptime: 246:02:20

Bundle-Ether1008 is up, line protocol is up
Encapsulation ARPA

Traffic Stats:(2 second rates)                                 Delta
  Input  Packets:                    6489005                      14
  Input  pps:                              8
  Input  Bytes:                   1507217455                    1274
  Input  Kbps (rate):                      5                  (  0%)
  Output Packets:                    7079943                      15
  Output pps:                              9
  Output Bytes:                   1490126647                    2024
  Output Kbps (rate):                      8                  (  0%)

Errors Stats:
  Input  Total:                            0                       0
  Input  CRC:                              0                       0
  Input  Frame:                            0                       0
  Input  Overrun:                          0                       0
  Output Total:                            0                       0
  Output Underrun:                         0                       0
```

Process Management

- Process
  - An executable portion of code run within its own memory space
  - 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
  - Processes can be viewed or restarted by name or JID on a system level or for a specific LC
- Threads
  - A process may contain one or more threads or a‘sub-process’. e.g. OSPF process has a thread which handles ‘hellos’.
  - Each thread is assigned a PID#. The PID changes if the process is stopped and restarted
- Processes can even be followed or completely dumped (memory capture) for troubleshooting by TAC

Process Restartability

```
RP/0/RP1/CPU0:CRS1#process shutdown snmpd

RP/0/RP1/CPU0:CRS1#show processes snmpd
                  Job Id: 288`
                     PID: 143532
          Executable path: /disk0/hfr-base-4.2.1/bin/snmpd
               Instance #: 1
                  Respawn: ON
            Respawn count: 1
             Last started: Mon May  9 15:32:22 2015
            Process state: Killed (last exit status: 15)
                    Ready: 11.636s
```

Process state reported as 'killed'

```
RP/0/RP1/CPU0:CRS1#process restart snmpd

RP/0/RP1/CPU0:CRS1#show processes snmpd
                  Job Id: 288
                     PID: 8528114
          Executable path: /disk0/hfr-base-4.2.1/bin/snmpd
               Instance #: 1
                  Respawn: ON
            Respawn count: 2
             Last started: Thu May 12 11:46:38 2015
            Process state: Run (last exit status : 15)
                    Ready: 6.657s
```

JID# remains constant, PID# changed on restart

Respawn counter incremented with process restart

‘Monitor process’ command

- Command provide Unix ‘top’ like information
- Displays details on number of running processes, CPU and memory utilization
- 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
- To change the parameters displayed by monitor processes, enter one of the interactive commands eg. ? to get help, n for the number of entries, t – sorted on cpu time, q to quit

```
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

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

t - Sort on CPU time

m - Sort on memory usage

‘Show memory compare’ command

Process how to use the command:

- **Takes the initial snapshot of heap usage**

```
RP/0/RP1/CPU0:CRS1#show memory compare start
Successfully stored memory snapshot /harddisk:/malloc_dump/memcmp_start.out
```

- **Takes the second snapshot of heap usage**

```
RP/0/RP1/CPU0:CRS1#show memory compare end
Successfully stored memory snapshot /harddisk:/malloc_dump/memcmp_end.out
```

- **Display the heap memory comparison report**

```
RP/0/RP1/CPU0:CRS1#show memory compare report

JID    name                 mem before    mem after     difference mallocs restart
---    ----                 ----------    ---------     ---------- ------- -------
57     i2c_server           11756         11916         160        1
121    bgp                  2522256       2522208       -48        -1
234    lpts_pa              408536        407632        -904       -14
224    isis                 3089108       3087900       -1208      0
314    tcp                  247196        245740        -1456      -9
241    netio                808136        806464        -1672      -46
```

‘Show tech’ command

```
RP/0/RP0/CPU0:CRS1# show tech-support snmp file harddisk:sh_tech_snmp
++ Show tech start time: 2015-Jan-22.090643.UTC ++
Wed Jan 22 09:06:44 UTC 2015 Waiting for gathering to complete
.....................................................................
Wed Jan 22 09:10:24 UTC 2015 Compressing show tech output
Show tech output available at 0/RP0/CPU0 : harddisk:/demo_sh_tech.tgz
++ Show tech end time: 2015-Jan-22.091025.UTC ++

RP/0/RP0/CPU0:CRS1# dir harddisk: | in sh_tech_snmp
Wed Jan 22 09:10:46.951 UTC
58948       -rw-  709261      Wed Jan 22 09:10:25 2015  sh_tech_snmp.tgz
RP/0/RP0/CPU0:CRS1#


RP/0/RP1/CPU0:CRS1#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
```

Routing Protocol Configuration and Verification

IOS-XR Protocol Configuration

IOS XR

- Protocol configuration exists:
  - Only within the protocol
  - Hierarchical

  (multiple levels deep is common)

Avoids having to scroll back and forth in the configuration

Other Cisco Operating Systems

- Protocol configuration exists at:
  - Global Process Level
  - Routing Process
  - Interface Configuration

  (normally one level deep)

Configuring Static Routes

- Initialize the routing protocol
- Define the address family
- Define the route

```
router static
 address-family ipv4 unicast
  192.168.1.0/24 Serial0/0/0/0
  192.168.2.0/24 1.2.3.4
 !
 address-family ipv6 unicast
  fec0:1234::3/64 fec0::88
```

What about IPv6?

Add a description for each route

Configuring EIGRP

- Initialize the routing protocol
- Define the address-family
- Identify the interfaces

```
router eigrp 100
 address-family ipv4
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
 !
!
```

EIGRP Verification Commands

```
RP/0/0/CPU0:XR1#show eigrp interfaces

IPv4-EIGRP interfaces for AS(100)

                    Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface     Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi0/0/0/0         0        0/0       0     640/640          0           0
Gi0/0/0/2         2        0/0     106       0/10         532           0
```

```
RP/0/0/CPU0:XR1#show eigrp neighbors

IPv4-EIGRP neighbors for AS(100) vrf default

H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.123.1.2              Gi0/0/0/2         12 00:00:26 1275  5000  0  6
0   10.123.1.3              Gi0/0/0/2         12 00:00:26    2   200  0  6
```

Backwards support for older commands

Does not provide context sensitive help. (i.e. ?)

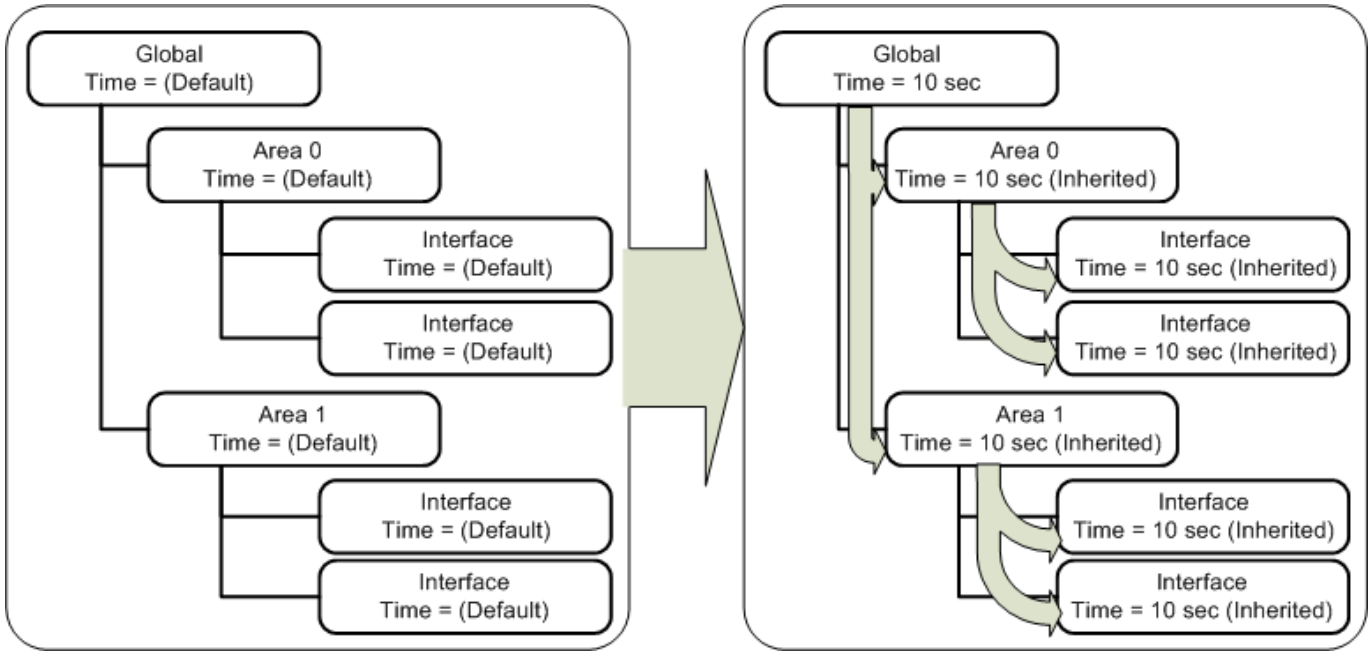
Configuring OSPF

- Initialize the routing protocol
- Define the OSPF areas
- Identify the interfaces

```
router ospf 1
 area 0
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
 !
 area 2
  interface GigabitEthernet0/0/0/2
  !
  interface GigabitEthernet0/0/0/3
  !
 !
!
```

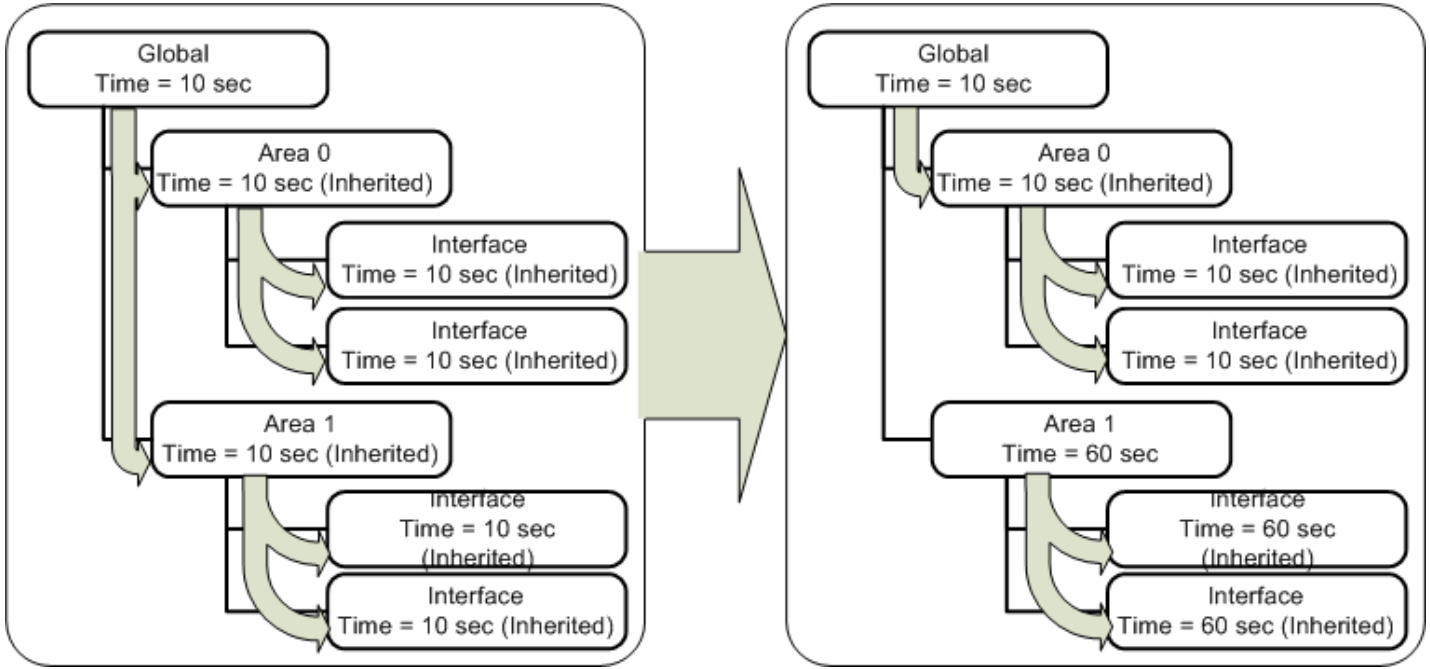
OSPF Hierarchical Configuration

- Settings at higher levels are inherited at lower levels



Protocol Pre-emption

- Configuration at lower level pre-empts global configuration



Configuring OSPF Features

Everything related to OSPF configuration resides in the process:

- Timers, network types, authentication

```
router ospf 1
 authentication message-digest
 message-digest-key 1 md5 CISCO
 network point-to-point
 area 0
  cost 100
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
 !
 area 2
  cost 2000
  interface GigabitEthernet0/0/0/2
  !
  interface GigabitEthernet0/0/0/3
   cost 9999
```

OSPF Side-By-Side Comparison to IOS with Inheritance

IOS XR

```
router ospf 1
 authentication message-digest
 message-digest-key 1 md5 CISCO
 network point-to-point
 area 0
  cost 100
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
 !
 area 2
  cost 2000
  interface GigabitEthernet0/0/0/2
  !
  interface GigabitEthernet0/0/0/3
   cost 9999
```

IOS

```
router ospf 1
  area 0 authentication message-digest
  area 2 authentication message-digest
  network 10.100.1.0 0.0.0.7 area 0
  network 10.200.1.0 0.0.0.15 area 2

interface gi0/0
 ip ospf network point-to-point
 ip ospf message-digest-key 1 md5 CISCO
 ip ospf cost 100
!
interface gi0/1
 ip ospf network point-to-point
 ip ospf message-digest-key 1 md5 CISCO
 ip ospf cost 100
!
interface gi0/2
 ip ospf network point-to-point
 ip ospf message-digest-key 1 md5 CISCO
 ip ospf cost 2000
!
interface gi0/3
 ip ospf network point-to-point
 ip ospf message-digest-key 1 md5 CISCO
 ip ospf cost 999
```

OSPF Verification Commands

```
RP/0/0/CPU0:XR1#show ospf interface brief

Interfaces for OSPF 1

Interface          PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Lo0                1     0               192.168.1.1/32     1     LOOP  0/0
Gi0/0/0/0          1     0               10.1.1.1/24        1     DR    0/0
Gi0/0/0/2          1     0               10.123.4.1/24      1     DROTH 2/3
```

```
RRP/0/0/CPU0:XR1#show ospf neighbor
Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.2.2     1     2WAY/DROTHER    00:00:32    10.123.4.2      GigabitEthernet0/0/0/2
    Neighbor is up for 00:46:45
192.168.3.3     1     FULL/BDR        00:00:35    10.123.4.3      GigabitEthernet0/0/0/2
    Neighbor is up for 00:49:55

Total neighbor count: 2
```

Configuring IS-IS

- Create the IS-IS routing process
- Identify the IS-IS NET
- Identify the interface(s)
- Activate the address-family
- Log Adjencies (optional)

```
router isis ISIS
 net 49.1234.0000.0000.0001.00
 log adjacency changes
 interface GigabitEthernet0/0/0/0
  address-family ipv4 unicast
  !
 !
 interface GigabitEthernet0/0/0/1
  address-family ipv4 unicast
  !
 !
!
```

Configuring IS-IS Features

Everything related to IS-IS resides in the process:

- Timers, circuit/router IS-IS levels, metric style, etc.

```
router isis ISIS
 net 49.1234.0000.0000.0001.00
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
 !
 interface GigabitEthernet0/0/0/0
  circuit-type level-1
  address-family ipv4 unicast
   !
 !
 interface GigabitEthernet0/0/0/1
  address-family ipv4 unicast
   !
 !
!
```

IS-IS Neighbor Verification

```
RP/0/0/CPU0:XR1#show isis neighbors

IS-IS ISIS neighbors:
System Id      Interface     SNPA            State Holdtime Type IETF-NSF
R2             Gi0/0/0/2     aabb.cc00.6500  Up    25       L1L2 Capable
R3             Gi0/0/0/2     aabb.cc00.6600  Up    7        L1L2 Capable

Total neighbor count: 2
```

Configuring BGP

- Create the BGP routing process
- Identify Router-ID
  (Required if no loopback interfaces)
- Initialize the Address-Family
- Advertise networks (optional)
- Identify the neighbors IP address
- Configure Session Details
- Associate the Address-Family to that neighbor's BGP session

```
router bgp 100
 bgp router-id 192.168.1.1
 address-family ipv4 unicast
  network 192.168.0.0/16
 !
 neighbor 10.0.0.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
 !
!
```

Routing policies are optional for IBGP sessions

BGP Route Policies

- Route-policies are optional for IBGP peers
- Route-policies are mandatory for EBGP peers

Considered a safety mechanism

```
router bgp 100
 address-family ipv4 unicast
!
 neighbor 10.0.0.1
  remote-as 200
  address-family ipv4 unicast
```

```
RP/0/0/CPU0: 16:28:06.171 : bgp[1047]: %ROUTING-BGP-6-NBR_NOPOLICY : No inbound
IPv4 Unicast policy is configured for eBGP neighbor 10.0.0.1. No IPv4 Unicast
prefixes will be accepted from the neighbor until inbound policy is configured.

RP/0/0/CPU0:16:28:06.171 : bgp[1047]: %ROUTING-BGP-6-NBR_NOPOLICY : No outbound
IPv4 Unicast policy is configured for eBGP neighbor 10.0.0.1. No IPv4 Unicast
prefixes will be sent to the neighbor until outbound policy is configured.
```

BGP Route Policies

- Route-policies are optional for IBGP peers
- Route-policies are mandatory for EBGP peers

```
router bgp 100
 address-family ipv4 unicast
!
 neighbor 10.0.0.1
  remote-as 200
  update-source Loopback0
  address-family ipv4 unicast
   route-policy PASS in
   route-policy PASS out
```

Routing policies are optional for IBGP sessions

BGP Verification 

```
RP/0/0/CPU0:XR1#show bgp ipv4 unicast summary
! Output omitted for brevity
BGP router identifier 192.168.1.1, local AS number 100
BGP main routing table version 4

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               4          4          4          4           4           4

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.12.1.2         0   100       8       7        4    0    0 00:05:23           0
```

```
RP/0/0/CPU0:XR1#show bgp ipv4 unicast
! Output omitted for brevity
BGP main routing table version 6

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 10.12.1.0/24       0.0.0.0                  0         32768 i
* i                   10.12.1.2                0    100      0 i
*> 192.168.1.1/32     0.0.0.0                  0         32768 i
*>i192.168.2.2/32     10.12.1.2                0    100      0 i
Processed 2 prefixes, 2 paths
```

Show run ……

As explicit as you want!!!

```
RP/0/0/CPU0:XR1#show run router ospf 1
! Output omitted for brevity
router ospf 1
 area 0
  interface TenGigE9/0/0/0
  !
 !
 area 1
  interface TenGigE9/0/0/1
    cost 10
  interface TenGigE9/0/0/2
  !
 !
!
RP/0/0/CPU0:XR1#show run router ospf 1 area 1 interface TenGigE9/0/0/1
! Output omitted for brevity
router ospf 1
 area 1
  interface TenGigE9/0/0/1
    cost 10
  !
 !
!
```

Supports Autocomplete

Works with all protocols and almost any process!!

IOS XR: Multicast

Configuring Multicast Routing

- Enable multicast routing process
- Identify the address-family (IPv4 / IPv6)
- List Interfaces and **enable**
  (Or enable all of them)
- Specify PIM parameters (opt.) (hello intervals, RP, etc.)
- Specify IGMP parameters (opt.)

```
multicast-routing
 address family ipv4
   interface GigabitEthernet0/0/0/0
     enable
   interface GigabitEthernet0/0/0/1
      enable

OR

multicast-routing
 address family ipv4
   interface all enable
```

Configuring PIM (Optional) 

- PIM is enabled by default for all interfaces with multicast routing enabled
- Used for setting RPs, hello intervals or other PIM related settings

```
router pim
 address family ipv4
   hello-interval 20
   rp-address 10.10.10.1
```

Configuring IGMP (Optional)

- IGMP is enabled by default for all interfaces with multicast routing enabled
- Used for tweaking various IGMP settings

```
router igmp
  interface GigabitEthernet0/0/0/0
    version 3
    query interval 40
```

Route Policy Language

What is RPL

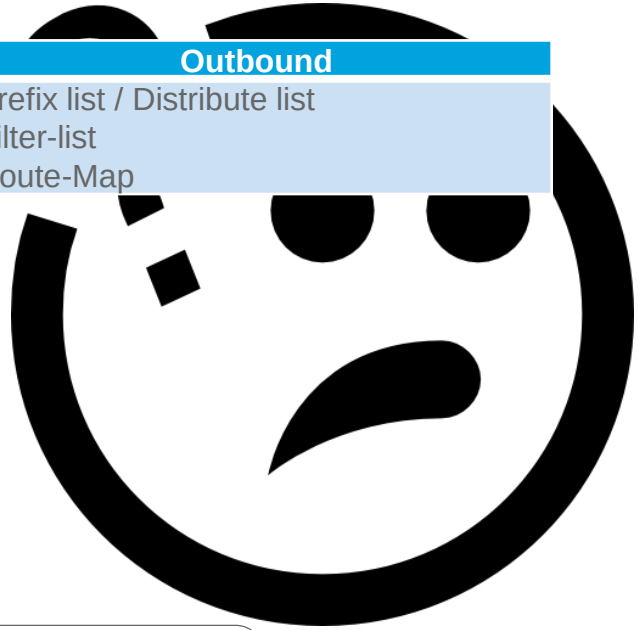
- Route Policy Language
- Used to filter routing information
  - Remove routes
  - Change attributes
- Test a specific policy before its applied

How many of you get confused by route-maps?

RPL brings clarity to Route-Maps

```
router bgp 100
  bgp log-neighbor-changes
  neighbor 100.64.1.1remote-as 65000
  neighbor 100.64.1.1password bgn2dcx
  neighbor 100.64.1.1remove-private-as
  neighbor 100.64.1.1soft-reconfiguration inbound
  neighbor 100.64.1.1prefix-list INTERNET-IN in
  neighbor 100.64.1.1prefix-list INTERNET-OUT out
  neighbor 100.64.1.1route-map INTERNET-IN in
  neighbor 100.64.1.1route-map INTERNET-OUT out
  neighbor 100.64.1.1filter-list 3 in
  neighbor 100.64.1.1filter-list 7 out
```

Inbound	Outbound
Route-Map	Prefix list / Distribute list
Filter-list	Filter-list
Prefix list / Distribute list	Route-Map



What is the order of processing?

Actions in a RPL

There is an implicit drop at the end of RPL processing.
A route must be given a '**ticket**' to ensure that it has been inspected by the RPL

- Pass – prefix allowed if not later dropped
  - **`pass`** grants a ticket to defeat default drop
  - Execution continues after pass
- Set – value changed, prefix allowed if not later dropped
  - Any **`set`** at any level grants a ticket
  - Execution continues after **`set`**
  - Values can be set more than once
- Drop – prefix is discarded
  - Explicit drop stops policy execution
  - Implicit drop (if policy runs to end without getting a ticket)
- **Done –** accepts prefix and stops processing

Basic RPL Examples

- Basic Pass Policy

Example Configuration

```
route-policy PASS-ALL
  pass
end-policy
```

- Basic Drop Policy

Example Configuration

```
route-policy DROP-ALL
  drop
end-policy
```

Somewhat redundant due to implicit drop

RPL Examples

- Basic conditional statement

Logic	Example Configuration

if **Match-Condition-One** then
  Action-One
end-if
	`if med eq 150 then`
	`  pass`
	`endif`

Conditional Match

Action

- Branching options

Logic	Example Configuration

if **Match-Condition-One** then
  Action-One
else
  Action-Two
end-if
	if destination in (10.0.0.0/8 ge 8) then
	  pass
	else
	  drop
	endif

Comparison operator

Notice we are matching networks directly in the RPL.

Supports Prefix Matching or WildCard

RPL Examples
(continued)

- Multiple Branching options

Logic	Example Configuration

if **Match-Condition-One** then
Action-One
elseif **Match-Condtion-Two** then
Action-Two
else
Action-Three
end-if
	if destination in (10.0.0.0/8 ge 8) then
	set tag 1
	elseif destination in (172.16.0.0/12 ge12) then
	set tag 2
	else
	drop
	endif

Nested Conditions

- ***If*** statements within other ***if/elseif/else*** statements
  - Method or placing multiple conditions
- Nesting can be any depth

Logic	Example Configuration

if **MATCHING-CONDITION-ONE** then
  if **MATCHING-CONDITION-TWO** then
  ACTION-ONE
  end-if
end-if
	if as-path passes-through '100' then
	  if destination in (172.16.0.0/12 ge12) then
	  pass
	  endif
	  endif

Original Value

- A conditional match does not occur on intermediary values during the route policy processing.

Conditional Matches on Original Value

```
route-policy ORGINAL-VALUES
  if med eq 100 then
    set med 200
  endif
  if med eq 200 then
    drop
  endif
end-policy
```

- In the example, only the original routes with a MED of 200 are dropped and the routes with values set to 200 are not dropped.

Simplifying BGP AS-Path Conditions

AS Path Selection Criteria	Route-Map AS-Path ACL Logic *(ip as-path access-list 1)*	RPL Logic
Local Routes	`permit ^$`	`if as-path is-local`
Only Routes From Neighbor AS 200	`permit ^200_`	`if as-path neighbor-is '200'`
Only Routes Originating From AS 200	permit _200$	`if as-path originates-from '200'`
Passes Through AS200	permit _200_	`if as-path passes-through '200'`
Routes From 3 ASes or less away	permit ^[0-9]+ [0-9]+ [0-9]+?	`if as-path length le 3`

RPL Policy Sets

- Prefix-lists, ACLs, AS_Path ACLs can be confusing because of permit/deny actions
- IOS XR uses policy sets to store the same information: Prefix Set, Community Set, Extended Community Set, AS_Path Set
- There is not a deny in a Policy Set
- Processing occurs until the first match is made

Named and Inline Set (Same behavior)

Inline Example Configuration

```
if destination in (10.0.0.0/8 ge 8, 172.16.0.0/12 ge 12, 192.168.0.0/16 ge 16) then
   pass
  else
   drop
  endif
```

Set Example Configuration

```
route-policy RFC1918-PREFIX-SET
  if destination in PREFIX-SET-RFC1918 then
    pass
  endif
end-policy
!
prefix-set PREFIX-SET-RFC1918
  10.0.0.0/8 ge 8,
  172.16.0.0/12 ge 12,
  192.168.0.0/16 ge 16
end-set
```

Viewing Set Based RPLS

Keyword required to see sets in the RPL

Inline Example Configuration

```
RP/0/0/CPU0:XR1#show rpl route-policy RFC1918-PREFIX-SET inline

route-policy RFC1918-PREFIX-SET
  if destination in (10.0.0.0/8 ge 8, 172.16.0.0/12 ge 12, 192.168.0.0/16 ge 16) then
    pass
  endif
end-policy
```

Avoids having to scroll back and forth in the configuration

RPL Existence

- **Question:** Non IOS XR systems have a difference behavior to a route if there is reference to a non-existent ACL, prefix-list, or route-map. How does IOS XR process these occurrences?

- **Answer:** IOS XR RPL does not use access-lists or prefix-lists. It uses policy-sets.
- If a policy-set or RPL is referenced and does not exist, the configuration will fail on the commit!

RPL Examples

Bad RPL Logic

```
route-policy METRIC-MODIFICATION
  if destination in (192.168.0.0/16 ge 16) then
    set med 100
  endif
  set med 200
end-policy
```

Overwrites Setting

Good RPL Logic
Option #1

```
route-policy METRIC-MODIFICATION
  if destination in (10.0.0.0/8 ge 8) then
    set med 100
  else
    set med 200
  endif
end-policy
```

Option #2

```
route-policy METRIC-MODIFICATION
  if destination in (10.0.0.0/8 ge 8) then
    set med 100
    done
  endif
  set med 200
end-policy
```

Stops all processing on matched prefixes

Nesting of RPLs

Example Configuration

```
route-policy PARENT
    apply CHILD-ONE
    apply CHILD-TWO
    pass
end-policy

route-policy CHILD-ONE
     set weight 100
end-policy

route-policy CHILD-TWO
     set community (2:1234) additive
end-policy
```

Can go multiple levels deep

Parameter Passing

Single Parameter

```
route-policy PARAM($MED)
  set med $MED
end-policy

router bgp 300
address-family ipv4 unicast
 !
neighbor 192.1.1.2
  remote-as 400
  address-family ipv4 unicast
    route-policy PARAM (50) in
    route-policy PASS-ALL out
```

List of policy parameters

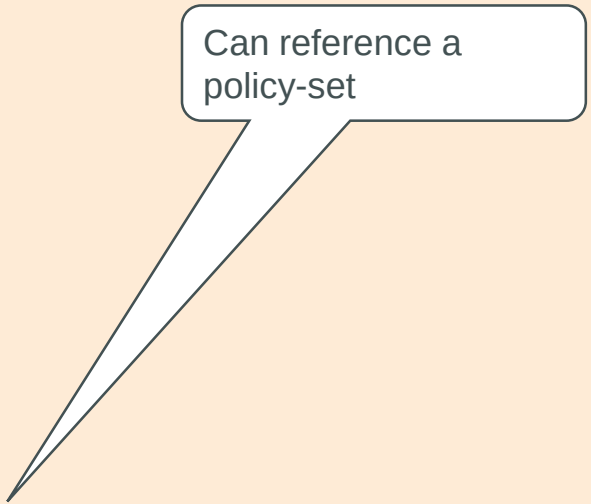
Accessing the passed parameter

Calling policy and passing parameter

Using Multiple Parameters in BGP

Multiple Parameter

```
route-policy SP-PEER($AS, $PREFIX)
  if destination in $PREFIX and as-path originates-from '$AS'  then
    pass
  endif
end-policy

router bgp 300
address-family ipv4 unicast
 !
neighbor 192.1.1.2
  remote-as 400
  address-family ipv4 unicast
    route-policy SP-PEER (50, CUST1-PREFIX-SET) in
    route-policy PASS-ALL out
```



Boolean Operators

Negation

```
if not destination in PREFIX-SET-RFC1918 then
    pass
endif
```

Conjunction

```
if destination in PREFIX-SET-RFC1918 and as-path passes-through '100'  then
    pass
endif
```

Disjunction

```
if destination in PREFIX-SET-RFC1918 or as-path passes-through '100'  then
    pass
endif
```

Boolean (Logical) Operations

- Comparison operators are context sensitive
  - Semantic check not done until RPL policy use is committed
- Supported Operators - Not, And, & Or (in order of precedence)

```
if Not Match-Condition-One and Match-Condition-Two or Match-Condition-Three
```

1. Prefixes not in **Match-Condition-One**
2. Prefixes not in **Match-Condition-One**, but are in **Match-Condition-Two**
3. Prefixes not in **Match-Condition-One**, but are in **Match-Condition-Two**, however any prefix in **Match-Condition-Three** are allowed regardless of **Match-Condition-One** or **Match-Condition-Two**

Boolean (Logical) Operations

```
if (((Not Match-Condition-One) and Match-Condition-Two) or Match-Condition-Three)
```

- Conditional match that requires a route to not pass through AS 100 or AS 200, and must be within the 192.168.0.0/16 network range

Use of parentheses

if not (as-path passes-through '100' or as-path passes-through '200') and destination in (192.168.0.0/16 ge 16)

Prefix List Functionality in an RPL

- Prefix List

```
ipv4 prefix-list PREFIX-LIST deny 192.168.1.1/32
ipv4 prefix-list PREFIX-LIST permit 192.168.0.0/16 ge 16
```

- RPL Logic

```
if (destination in (192.168.0.0/16 ge 16) and (not destination in (192.168.1.1))) then
   pass
endif
```

RPL Show Commands

Only display prefixes matching policy – filter show command

```
RP/0/0/1:CRS1#show bgp route-policy SAMPLE
BGP router identifier 172.20.1.1, local AS number 1820
BGP main routing table version 729
Dampening enabled
BGP scan interval 60 secs
Status codes: s suppressed, d damped, h history, * valid, > best
i - internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network            Next Hop            Metric      LocPrf
Weight    Path
* 10.13.0.0/16      192.168.40.24 0
1878 704 701 200 ?
* 10.16.0.0/16      192.168.40.24 0
1878 704 701 I
```

RPL Show Commands [attachpoint]

RPL Attachment Points

```
RP/0/RP0/CPU0:CRS1#show rpl route-policy PASS attachpoints

BGP Attachpoint: Neighbor

Neighbor/Group   type   afi/safi    in/out    vrf name
-----------------------------------------------------
cavs             nbr    IPv4/uni    in          default
```

RPL Show Commands

RPL Attachment States

```
RP/0/0/CPU0:CRS1#show rpl route-policy states
ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced


The following policies are (ACTIVE)
----------------------------------------
..
The following policies are (INACTIVE)
----------------------------------------
None found with this status.


The following policies are (UNUSED)
----------------------------------------
…
```

Modifying an Existing RPL

```
RP/0/RP0/CPU0:CRS-D#conf t
RP/0/RP0/CPU0:CRS-D(config)#route-policy PASS-ALL-TEST
RP/0/RP0/CPU0:CRS-D(config-rpl)#pass
RP/0/RP0/CPU0:CRS-D(config-rpl)#end-policy
RP/0/RP0/CPU0:CRS-D(config)#commit
Mon Jul 11 20:53:33.817 EDT
RP/0/RP0/CPU0:CRS-D(config)#route-policy PASS-ALL-TEST
Mon Jul 11 20:53:41.436 EDT
% WA
Reco
defi
```

```
RP/0/RP0/CPU0:CRS-D#edit route-policy PASS-ALL-TEST ?
  emacs  to use Emacs editor
  nano   to use nano editor
  vim    to use Vim editor
  <cr>
RP/0/RP0/CPU0:CRS-D#edit route-policy PASS-ALL-TEST
```

exists!

t

IOS XR: MPLS LDP

Configuring MPLS LDP

- Enable LDP process
- List Interfaces
- Specify Interface specific items under each interface (opt.) (discovery address, hello)
- Configure MPLS OAM (opt.)

```
mpls ldp
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
  address-family ipv4
   discovery transport-address 1.1.1.1
  !
 !
mpls oam
```

MPLS Verification Commands

```
RP/0/0/CPU0:CRS1#show mpls forwarding
Local  Outgoing    Prefix           Outgoing     Next Hop        Bytes       T
Label  Label       or ID            Interface                    Switched    O
------ ----------- ---------------- ------------ --------------- ----------- -
17     Pop Label   192.168.1.1/32   PO0/6/0/2    192.168.6.1     0
18     Pop Label   192.168.1.3/32   PO0/6/0/0    192.168.7.3     0
19     Unlabelled  192.168.1.4/32   PO0/6/0/1    192.168.8.5     0
20     Unlabelled  192.168.1.5/32   PO0/6/0/1    192.168.8.5     0
21     29          192.168.1.6/32   PO0/6/0/2    192.168.6.1     0
22     30          192.168.1.7/32   PO0/6/0/0    192.168.7.3     0
       Unlabelled  192.168.1.7/32   PO0/6/0/1    192.168.8.5     0
```

```
RP/0/0/CPU0:CRS1#show mpls interfaces
Interface                  LDP      Tunnel   Enabled
-------------------------- -------- -------- --------
POS0/6/0/0                 Yes      Yes      Yes
POS0/6/0/1                 No       Yes      Yes
POS0/6/0/2                 Yes      Yes      Yes
```

IOS XR L3VPN

VRF Configuration 

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
```

Does not remove the IP address like IOS does.

BGP VPNv4 Configuration

Configuring PE to RR VPNV4 iBGP Neighbors

```
router bgp 100
  bgp router-id 100.100.100.100
  address-family vpnv4 unicast
!
  neighbor 192.168.1.1
  remote-as 100
  update-source loopback 0
  address-family vpnv4 unicast
```

PE-CE Configuration

```
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 static
    vrf vpn1
        address-family ipv4 unicast
            10.1.1.1/32 g0/1/0/2
```

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

Reasons Why IOS XR Rocks!

IOS XR Benefits

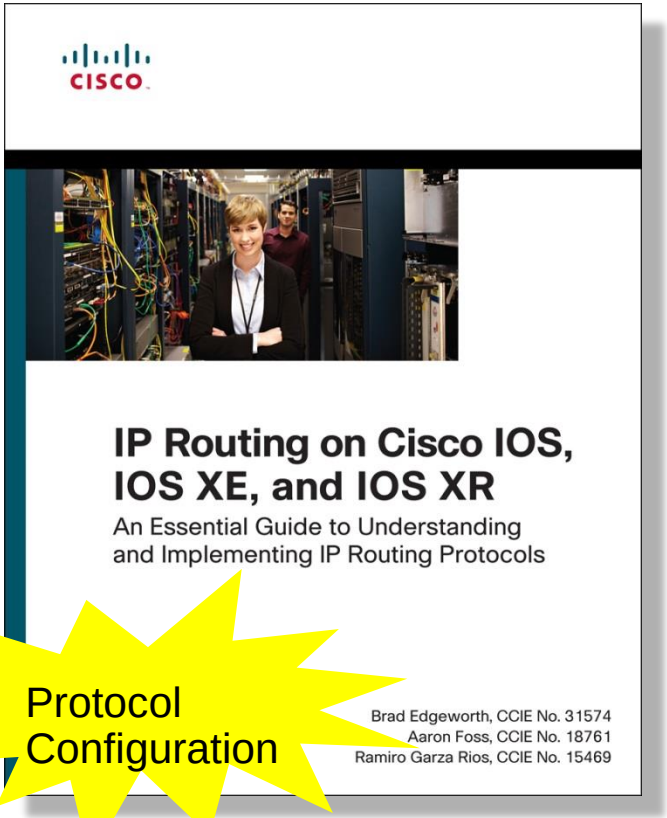
- Modular architecture:
  - Built with Scalability, Stability, and Security in Mind
- Configuration is hierarchical:
  - Everything is in one place – Reduces the Scroll-A-Thon
  - Ability to view the specify the portion of the protocol configuration in the running-configuration that you want to see
- Improved Configuration management:
  - SysDB
    History of Changes (Who, When, What) and rollback ability
  - 2-Stage Commits, and Commit Confirm – Reduces router lockouts

IOS XR Benefits

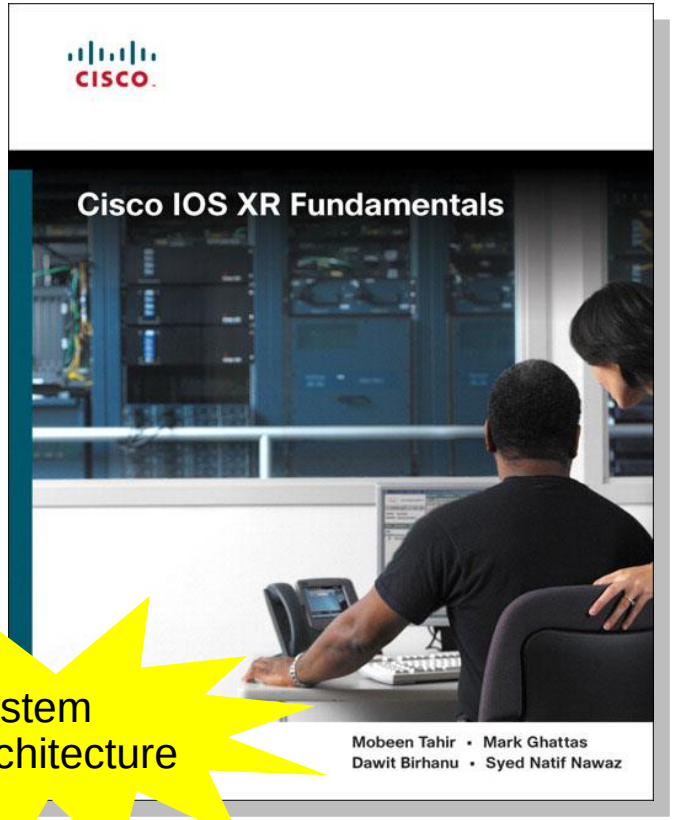
- Operational support
  - Multi-pipe filtering
  - Traces
  - Interface Monitoring
  - Process Monitoring
  - Configuration Rollbacks (Did we mention that already?)
- Route Policy Language
  - Provides Clarity
  - Scalability (Nesting & Parameterization)
  - Preview of impact to routing table before applying

Are You Excited About IOS XR Yet?

Recommended Reading for BRKARC-1008



Protocol Configuration



System Architecture

Suggested Sessions

- **BRKSPG-2724**: Network Function Virtualization with IOS XR
- **BRKSPG-2904**: ASR-9000/IOS XR Understanding forwarding, troubleshooting the system and XR operations
- **BRKSPF-2069**: Introduction to the Next Generation IOS XR Architecture
- **BRKARC-2022**: Introduction to Network Convergence System (NCS6k)
- **BRKARC-2003**: Cisco ASR 9000 Architecture
- **BRKARC-2017**: Packet Journey inside ASR 9000

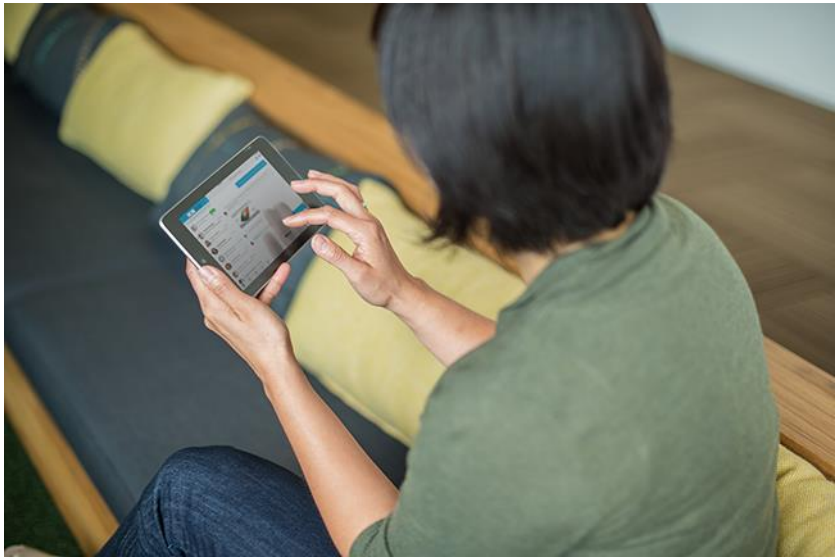
Cisco IOS XRv

Learn IOS XR with On-Hands Practice

- **Virtualized IOS XR Router**
  FREE DEMO VERSION (Requires CCO Account)
  - **Download Location**
    https://upload.cisco.com/cgi-bin/swc/fileexg/main.cgi?CONTYPES=Cisco-IOS-XRv
- **Cisco VIRL**
  Visit them at the DevNet Hub
  http://virl.cisco.com

Complete Your Online Session Evaluation

- Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 Amazon gift card.
- Complete your session surveys through the Cisco Live mobile app or from the Session Catalog on CiscoLive.com/us.



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

Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
  - LABRST-2000 (Intro to IOS XR)
  - LABRST-2001 (Learning BGP and RPL on IOS XR)
- Lunch & Learn
  - Wednesday – IOS XR Architecture at Lunch
- Meet the Engineer 1:1 meetings
- Related sessions

Please join us for the Service Provider Innovation Talk featuring:

Yvette Kanouff | Senior Vice President and General Manager, SP Business

Joe Cozzolino | Senior Vice President, Cisco Services

Thursday, July 14th, 2016

11:30 am - 12:30pm, In the Oceanside A room

What to expect from this innovation talk

- Insights on market trends and forecasts
- Preview of key technologies and capabilities
- Innovative demonstrations of the latest and greatest products
- Better understanding of how Cisco can help you succeed

Register to attend the session live now or watch the broadcast on cisco.com

Thank you


cisco
Cisco live!
July 10-14, 2016 • Las Vegas, NV