Mastering BGP: A Deep Dive into Basics and Design Best Practices for BGP and L3VPN

cisco Live!

Mankamana Mishra
Technical Leader Engineering

Serge Krier
Technical Leader Engineering

BRKMPL-2103

Cisco Webex App

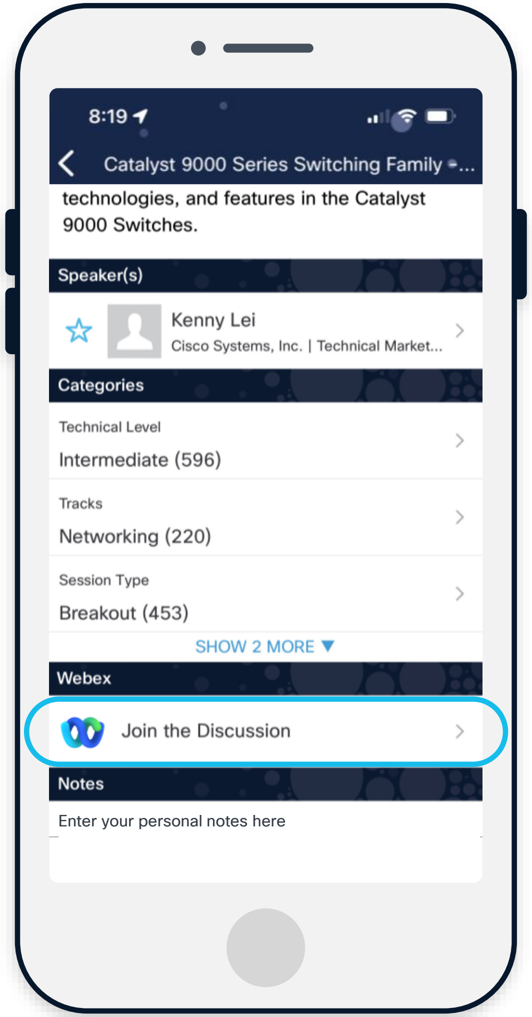
Questions?

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

How

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

Webex spaces will be moderated by the speaker until June 13, 2025.



8:19

Catalyst 9000 Series Switching Family -...

technologies, and features in the Catalyst 9000 Switches.

Speaker(s)

Kenny Lei
Cisco Systems, Inc. | Technical Market...

Categories

Technical Level
Intermediate (596)

Tracks
Networking (220)

Session Type
Breakout (453)

SHOW 2 MORE ▼

Webex

Join the Discussion

Notes

Enter your personal notes here

Before we Start

- This is an **introductory session** – covering basic **BGP** and **L3VPN** concepts
- If you're already a BGP expert, treat this as a **quick revision**
- **Duration**: 1 hour – focused on giving an **overview of BGP fundamentals and some best practices**
- **Q&A at the end** to keep the flow uninterrupted
- For deeper technical discussions, feel free to:
  - **Set up a Meet-the-Engineer session**
  - **Reach out on Webex anytime**

Agenda

01 Basic Terminology

02 BGP Deployment scenarios

03 Scaling BGP

04 BGP Optimal Route reflection

05 BGP Route policy

06 BGP Soft Reconfig

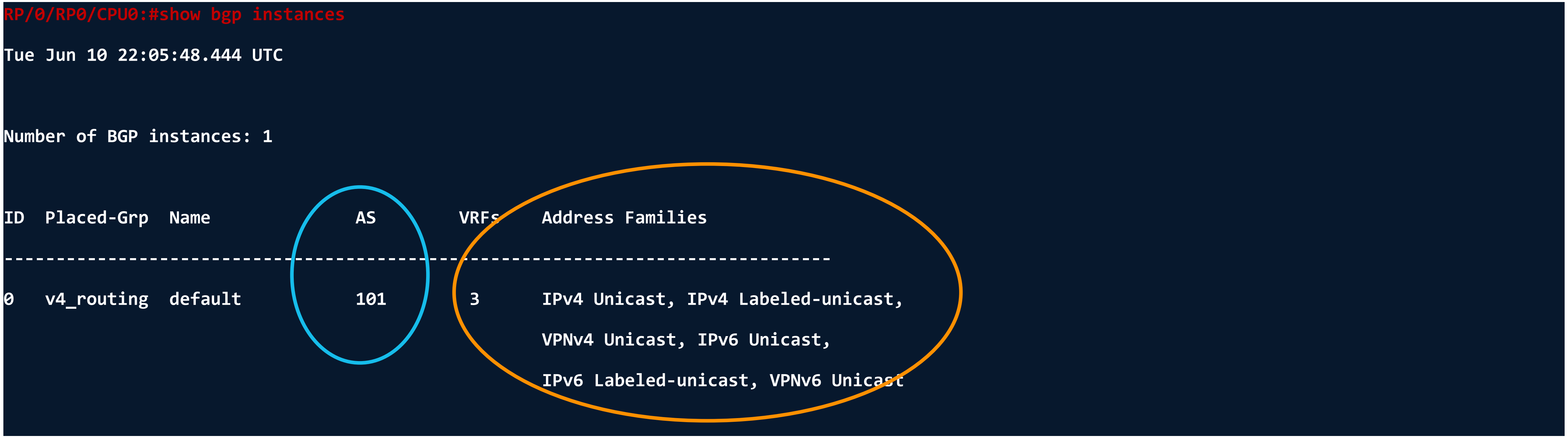
07 BGP Security

08 Summary

Basic Terminology

Terminology

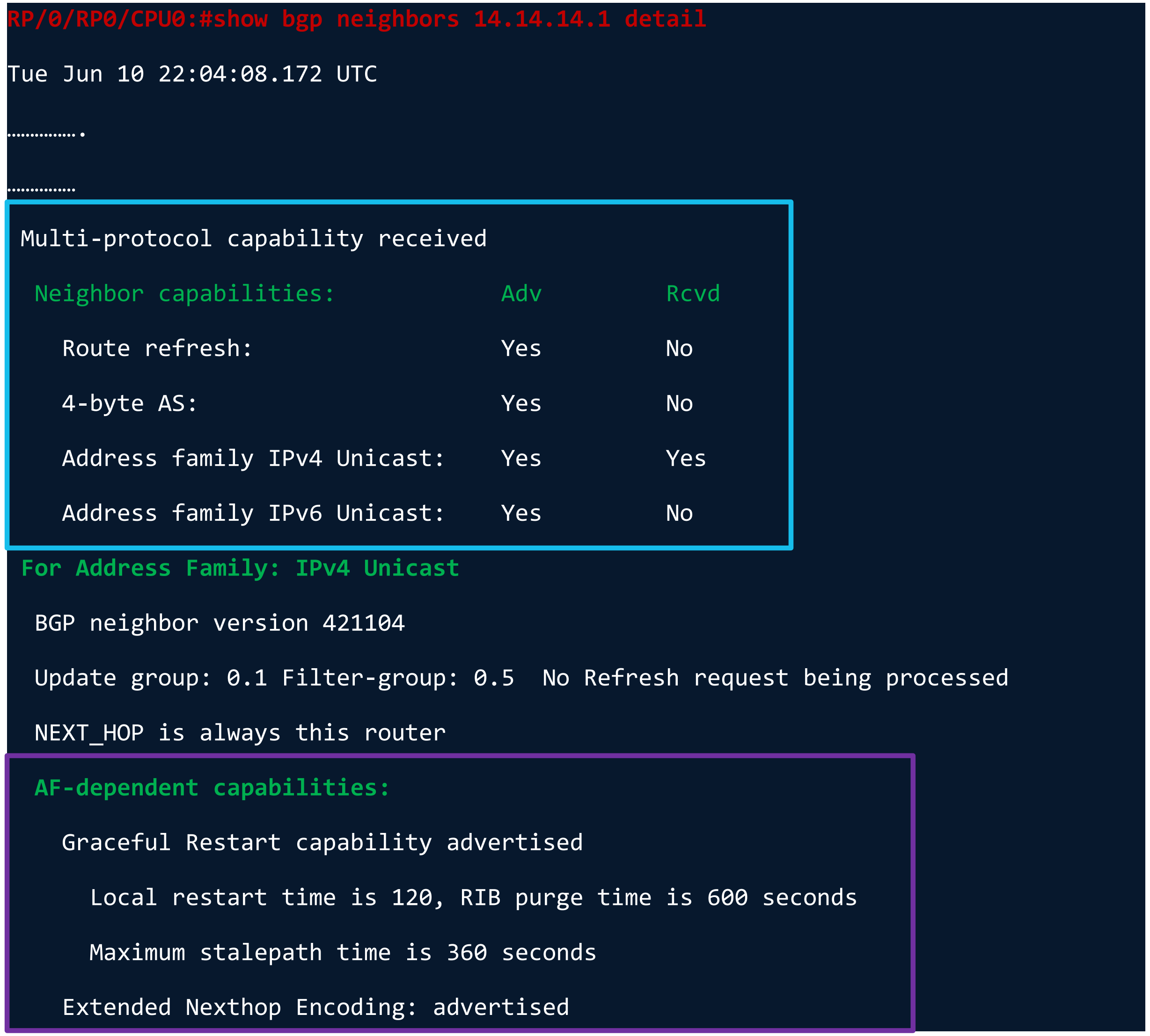
- **AS** : **Autonomous System** : Foundation concept: BGP is inter-AS protocol; each network domain is identified by an AS number. (2 or 4 bytes)
- **AFI** : Address Family Identifier (ex: 1 for IPv4)
- **SAFI** : Subsequent Adress Family Identifier (ex: 1 for Unicast)



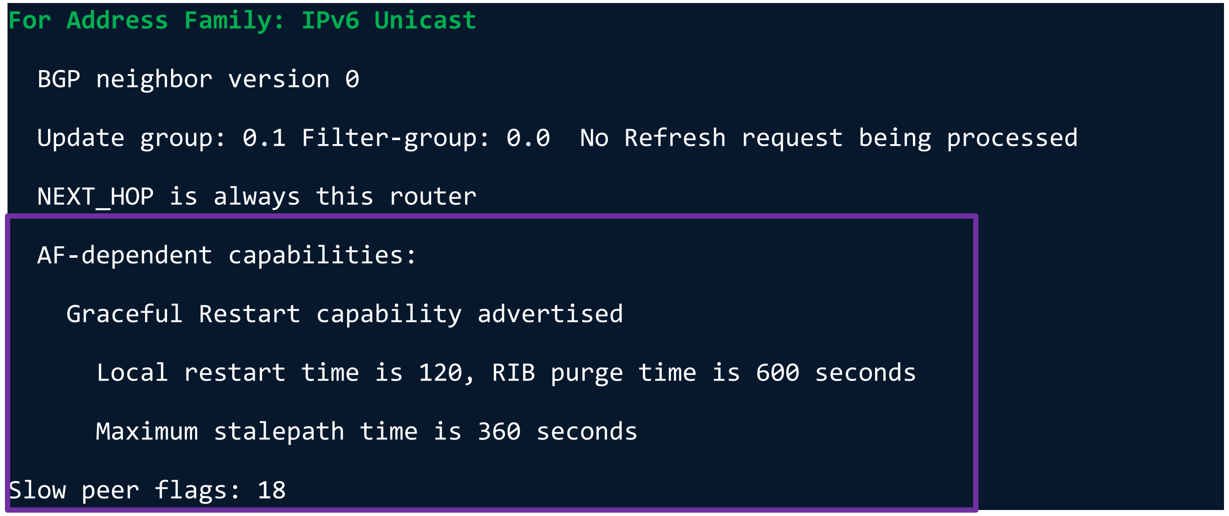
```
RP/0/RP0/CPU0:#show bgp instances

Tue Jun 10 22:05:48.444 UTC


Number of BGP instances: 1


ID  Placed-Grp  Name            AS        VRFs    Address Families

--------------------------------------------------------------------------------

0   v4_routing  default         101        3      IPv4 Unicast, IPv4 Labeled-unicast,

                                                  VPNv4 Unicast, IPv6 Unicast,

                                                  IPv6 Labeled-unicast, VPNv6 Unicast
```

Terminology

- **Capability**: carried in BGP OPEN Message, indicating supporting features in BGP



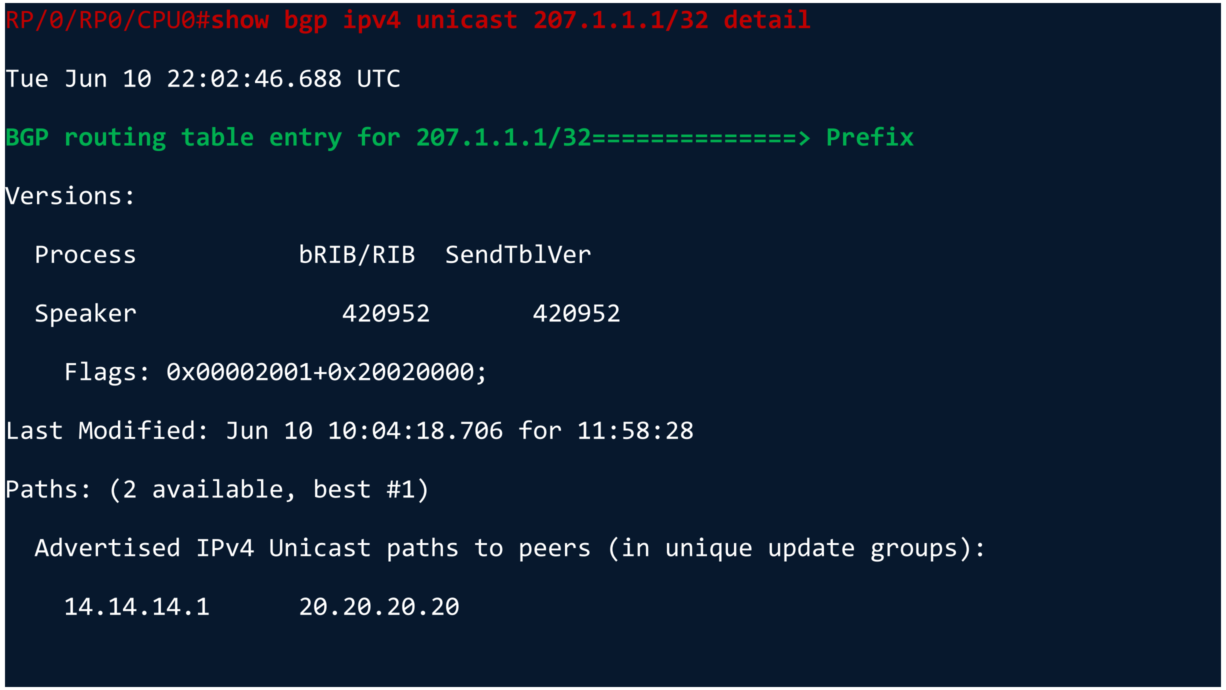
```
RP/0/RP0/CPU0:#show bgp neighbors 14.14.14.1 detail
Tue Jun 10 22:04:08.172 UTC
…………….
……………
 Multi-protocol capability received
  Neighbor capabilities:            Adv         Rcvd
    Route refresh:                  Yes         No
    4-byte AS:                      Yes         No
    Address family IPv4 Unicast:    Yes         Yes
    Address family IPv6 Unicast:    Yes         No
 For Address Family: IPv4 Unicast
  BGP neighbor version 421104
  Update group: 0.1 Filter-group: 0.5  No Refresh request being processed
  NEXT_HOP is always this router
  AF-dependent capabilities:
    Graceful Restart capability advertised
      Local restart time is 120, RIB purge time is 600 seconds
      Maximum stalepath time is 360 seconds
    Extended Nexthop Encoding: advertised
```



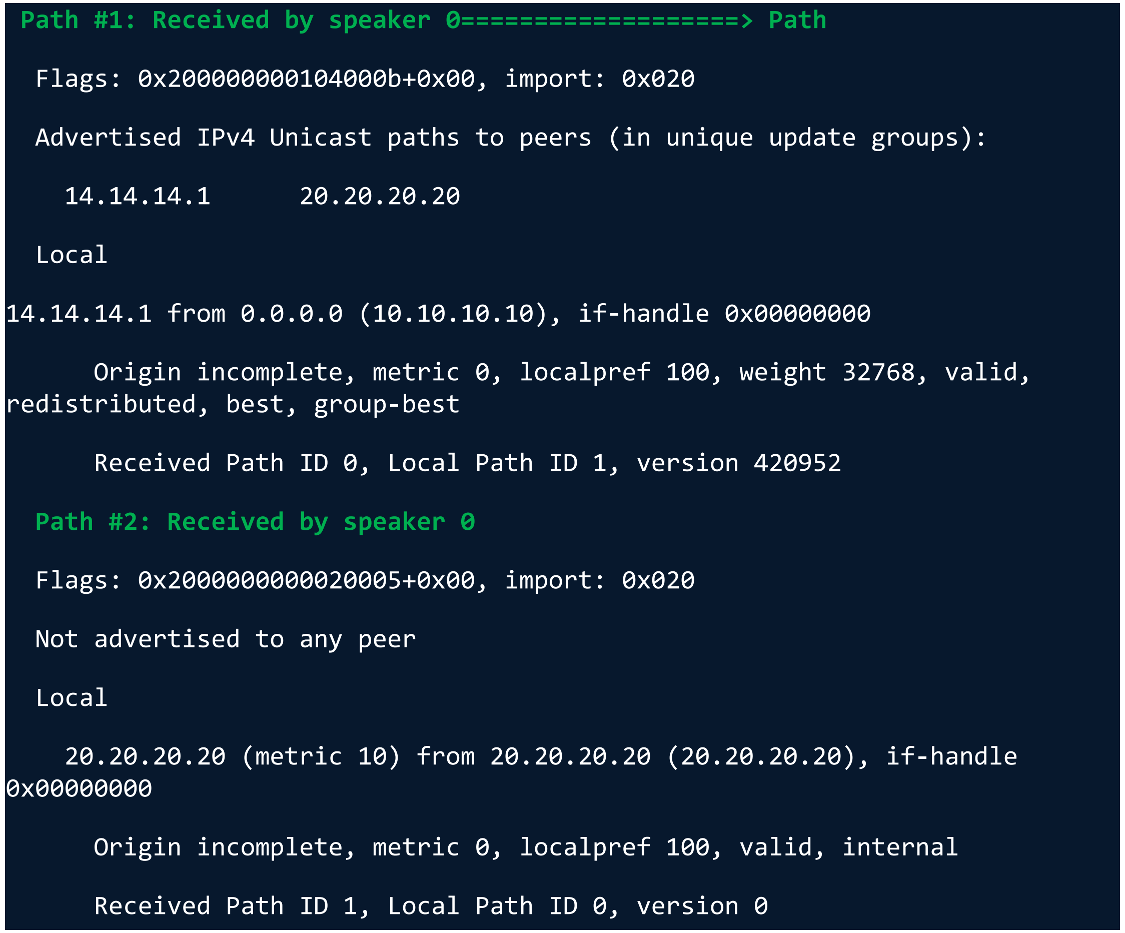
```
For Address Family: IPv6 Unicast
  BGP neighbor version 0
  Update group: 0.1 Filter-group: 0.0  No Refresh request being processed
  NEXT_HOP is always this router
  AF-dependent capabilities:
    Graceful Restart capability advertised
      Local restart time is 120, RIB purge time is 600 seconds
      Maximum stalepath time is 360 seconds
Slow peer flags: 18
```

Terminology

- **Prefix:** The basic destination – a block of IP addresses being advertised. (aka NET in IOS XR)
- **Path**: A complete set of information received from a BGP peer: **prefix (NLRI) + attributes**.



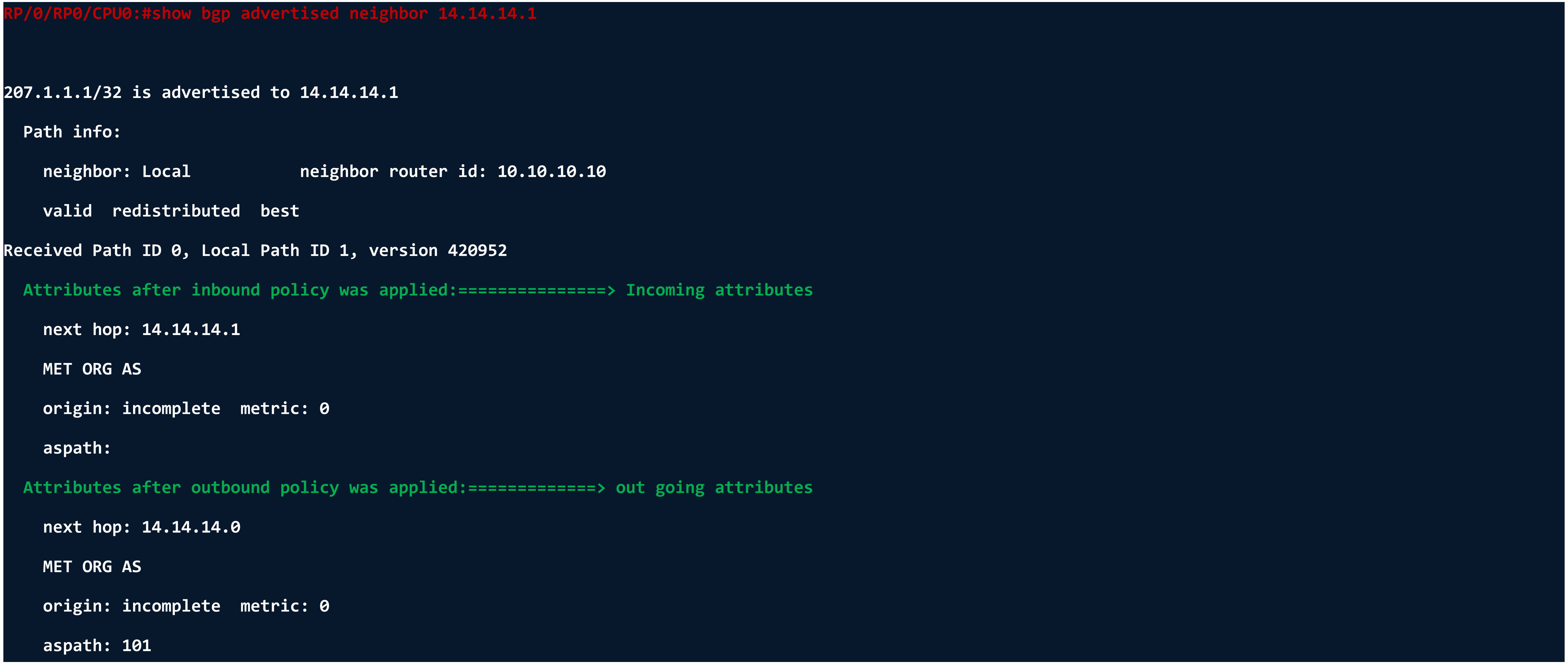
```
RP/0/RP0/CPU0#show bgp ipv4 unicast 207.1.1.1/32 detail
Tue Jun 10 22:02:46.688 UTC
BGP routing table entry for 207.1.1.1/32==============> Prefix
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker              420952       420952
    Flags: 0x00002001+0x20020000;
Last Modified: Jun 10 10:04:18.706 for 11:58:28
Paths: (2 available, best #1)
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    14.14.14.1      20.20.20.20
```



```
 Path #1: Received by speaker 0===================> Path
  Flags: 0x200000000104000b+0x00, import: 0x020
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    14.14.14.1      20.20.20.20
  Local
14.14.14.1 from 0.0.0.0 (10.10.10.10), if-handle 0x00000000
      Origin incomplete, metric 0, localpref 100, weight 32768, valid,
redistributed, best, group-best
      Received Path ID 0, Local Path ID 1, version 420952
  Path #2: Received by speaker 0
  Flags: 0x2000000000020005+0x00, import: 0x020
  Not advertised to any peer
  Local
    20.20.20.20 (metric 10) from 20.20.20.20 (20.20.20.20), if-handle
0x00000000
      Origin incomplete, metric 0, localpref 100, valid, internal
      Received Path ID 1, Local Path ID 0, version 0
```

Terminology

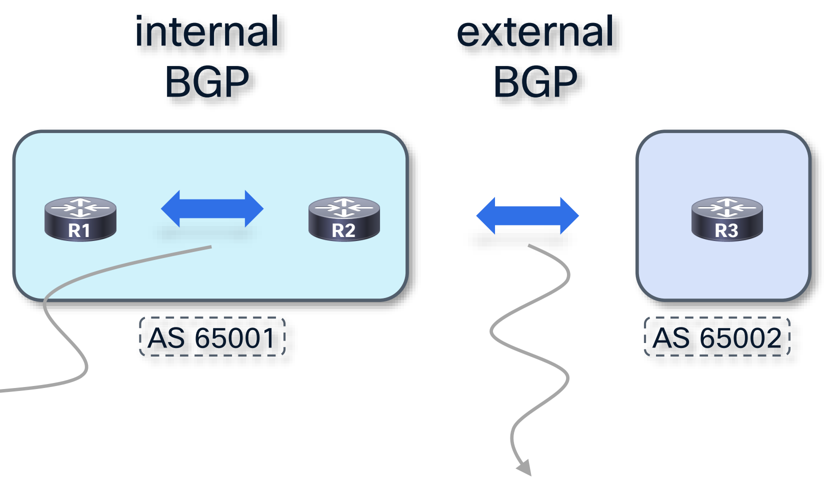
- **Attributes**: carried in BGP Update message indicating additional characteristics of the prefix



```
RP/0/RP0/CPU0:#show bgp advertised neighbor 14.14.14.1


207.1.1.1/32 is advertised to 14.14.14.1
  Path info:
    neighbor: Local           neighbor router id: 10.10.10.10
    valid  redistributed  best
Received Path ID 0, Local Path ID 1, version 420952
  Attributes after inbound policy was applied:===============> Incoming attributes
    next hop: 14.14.14.1
    MET ORG AS
    origin: incomplete  metric: 0
    aspath:
  Attributes after outbound policy was applied:=============> out going attributes
    next hop: 14.14.14.0
    MET ORG AS
    origin: incomplete  metric: 0
    aspath: 101
```

BGP Peering

- Once TCP session is established
- OPEN message
  - Capabilities
- Configured Autonomous System Number (ASN) must match
- eBGP if local AS <> remote AS
- iBGP if local AS = remote AS
- Authentication (if any) must match
- Minimum 1 address family needed



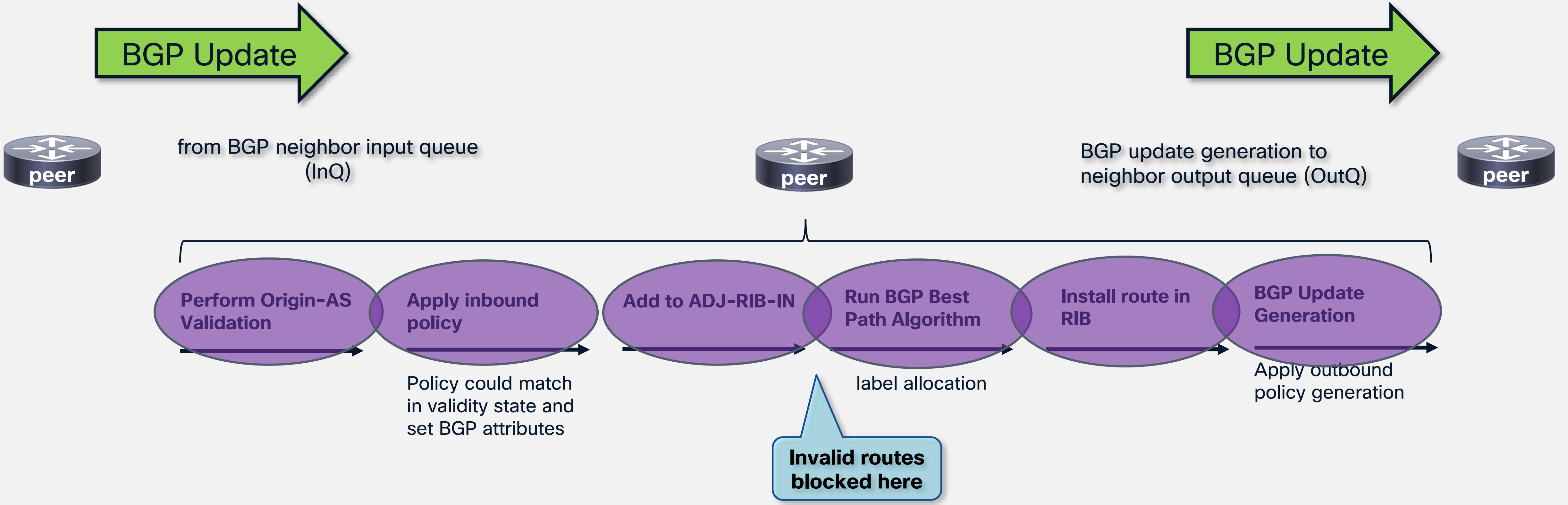
- iBGP TTL 255
- Peering between loopbacks

```
router bgp 65001
 address-family ipv4 unicast
 !
 neighbor 10.0.0.2
  remote-as 65001
  update-source Loopback0
  address-family ipv4 unicast
```

- eBGP TTL 1
- Peering between interface addresses

```
router bgp 65001
 address-family ipv4 unicast
 !
 neighbor 10.5.6.6
  remote-as 65002
  address-family ipv4 unicast
   route-policy PASS in
   route-policy PASS out
```

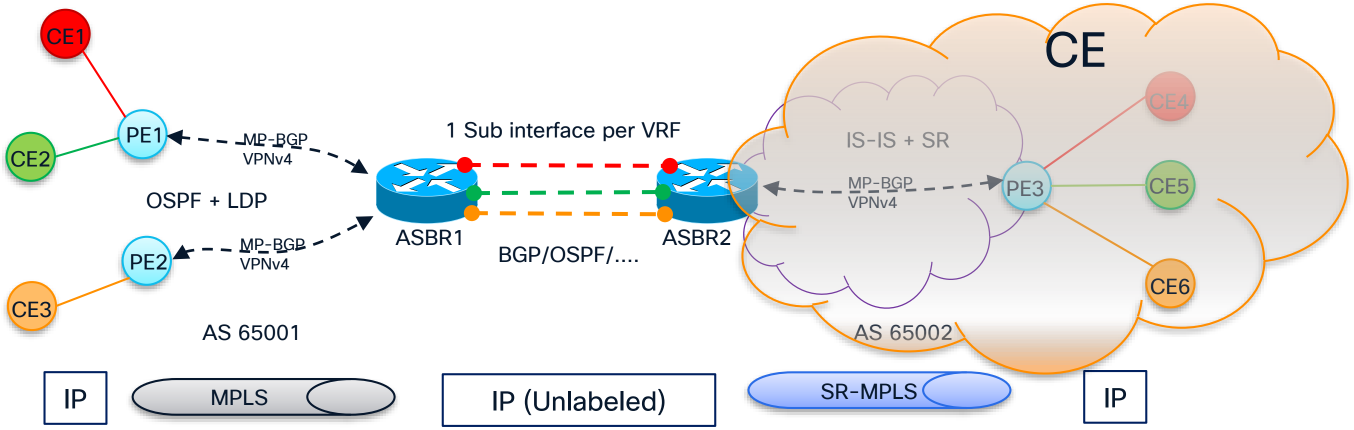
BGP Pipeline


BGP Update
BGP Update
peer
from BGP neighbor input queue (InQ)
peer
BGP update generation to neighbor output queue (OutQ)
peer
Perform Origin-AS Validation
Apply inbound policy
Add to ADJ-RIB-IN
Run BGP Best Path Algorithm
Install route in RIB
BGP Update Generation
Policy could match in validity state and set BGP attributes
label allocation
Apply outbound policy generation
Invalid routes blocked here

BGP Deployment scenarios L3VPN

Inter-AS Option A

- Option A is the simplest of the interconnection options.
- The AS Border Router (ASBR) of each AS defines an interface or sub-interface per VRF. Once defined, the ASBR will instantiate the VRF assigning the sub-interface to the VPN. This needs to be done per VPN requiring Inter-AS service.
- The sub-interfaces facing the other AS doesn't transport labeled traffic, only regular IP traffic. In order to exchange routing information with the remote ASBR, any routing protocol can be used.
- From the ASBR1 point of view, the remote AS is seen like any other regular CE device.



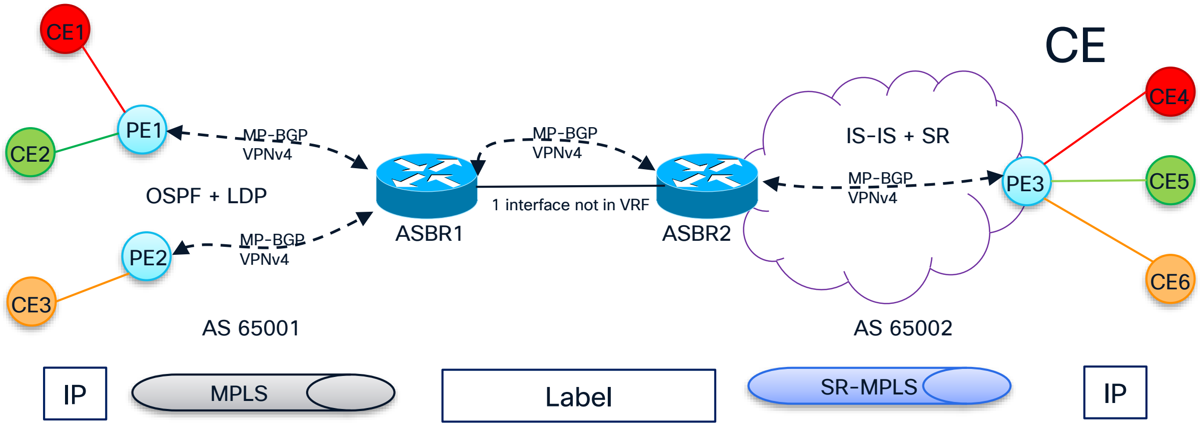
Inter-AS option A

- Benefits
  - **Simplicity** – Easy to understand and implement
  - **Flexibility** – Adapts to diverse network needs
  - **Clear Demarcation** – Separates responsibilities between MPLS L3VPN service providers
  - **Ease of Deployment** – Quick to roll out in various environments
  - **Traffic Control** – Leverages standard IP access-lists for filtering

- Drawback
  - **Poor Scalability** – Not ideal for large-scale or high-growth deployments

Inter-AS Option B

- The Option B is the second option covered in RFC 4364 for interconnecting sites of VPN customers connected to different autonomous systems
- Inter-AS Option B tries to avoid the operational complexity needed to set up a new VPN customer with inter-as connectivity *by moving complexity*. The new procedure partially solve scalability problems but introduces some new ones we didn't have with Option A.
- There is no need to configure one VRF per-VPN customer demanding interconnection. The ASBRs should be directly connected and perform the route exchange using a single interface (physical or logical) not assigned to a VRF.



Inter-AS option B

➢ Benefits

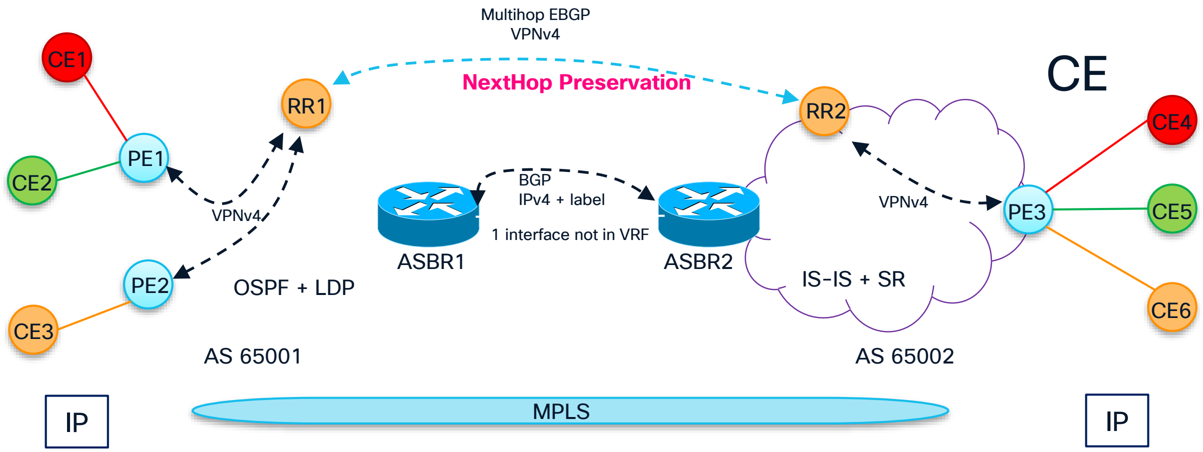
- **Enhances Scalability** – Supports large-scale multi-provider deployments
- **Simplifies Deployment** – Reduced operational complexity and faster provisioning

➢ Drawback

- **Diffuse Demarcation Points** – Interface aggregation blurs provider boundaries
- **Challenging Policy Enforcement** – Difficult to apply IP filtering precisely
- **Stronger Trust Dependencies** – Requires higher trust levels between providers
- **Need for Additional Security** – Extra measures needed to ensure data integrity and isolation

Inter-AS Option C

- Inter-AS Option C is the third option for interconnecting multi-AS backbones covered in RFC 4364. It’s the most scalable option of the three so far and it has its own applicability scenarios that we must be aware of to apply this design properly.
- the ASBRs don’t carry any of the VPN routes. ASBRs only take care of distributing labeled IPv4 routes of the PEs within their own AS.
- To improve scalability, one MP-EBGP VPNv4 session transports all VPN routes (external routes) between PEs or RR. In the case of using RR to exchange the external routes, the next hop of the VPNv4 routes must be preserved.
- The ASBR use EBGP to exchange the internal PE routing information between AS (internal routes). These internal routes correspond to the BGP next-hops of the external routes advertised through the multi-hop MP-EBGP session between PEs or RRs. The internal routes advertised by the ASBRs can be used to establish the MP-EBGP sessions between PEs and allows for LSP setup from the ingress to the egress PE.



Inter AS option C

Scalability

- The ASBRs do not store external routing information
- Resource conservation as the external information is not duplicated on the ASBRs.
- The RRs already store the routes.
- The RRs does not allocate label

Planes isolation

- Multi-hop EBGP VPNv4 for VPN routes
- EBGP labeled IPv4 for internal routes

Security

- Advertising of PE addresses to another
- not always a good option

QoS enforcement per VPN isn't possible at ASBR

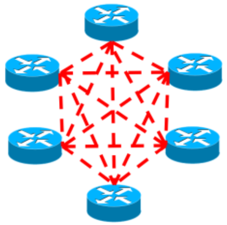
- VPN context doesn't exist at ASBRs
- Not possible to perform policing, filtering or accounting with per VPN granularity at ASBR

Which make this solution not a very good option when Autonomous Systems don't have a **strong trust relationship** between them

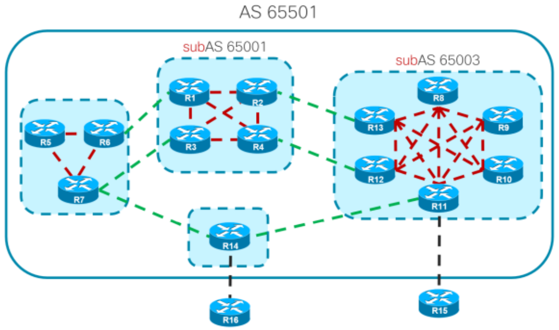
Scaling BGP

iBGP: 3 Models

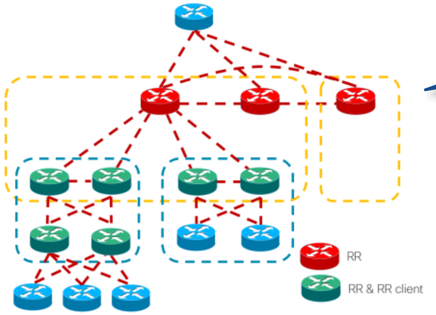
- Full iBGP mesh
  - Suffers from n*(n-1)/2: total number of sessions in networks
  - Still only (n-1) iBGP sessions per BGP edge router
  - Manageability: adding 1 edge BGP peer involves touching all other BGP speakers



- Confederations
  - Scalability by divide and conquer: Sub-Autonomous Systems
    - One or ore IGPs allowed
    - Slightly different BGP Best Path Calculation
    - Key routers are inline
    - Difficult to change design; merge Autonomous Systems, deploy new features



- Route Reflectors
  - Highest scalability; allows for hierarchical RRs
  - Few BGP sessions from the edge BGP speakers
  - RRs have many BGP sessions
  - ✓ Dedicated RRs do not forward packets: only BGP matters!
  - ✓ Easier to deploy new features (only on RRs)
  - ✓ Can be virtual routers

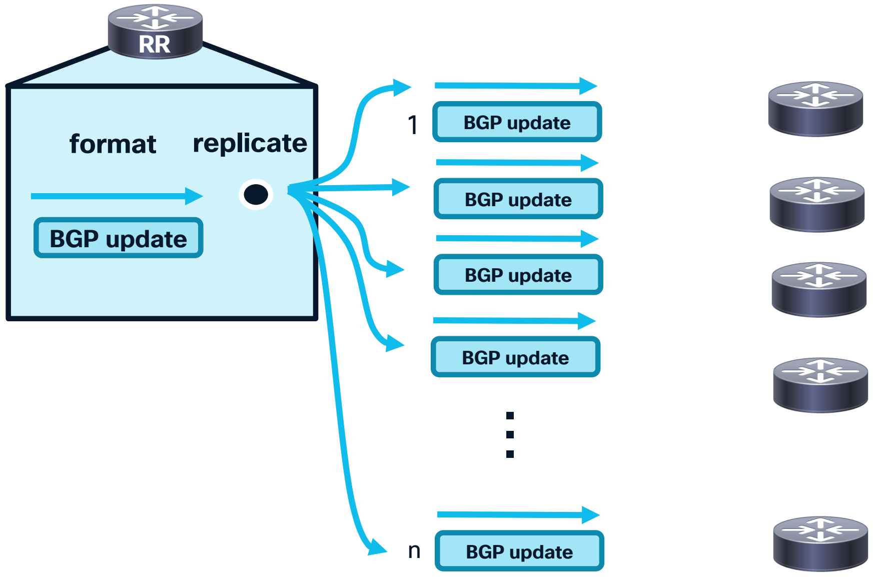


Hierarchical RRs

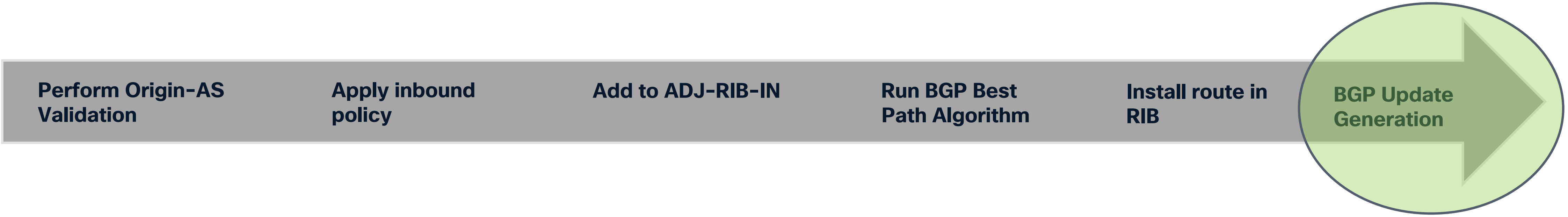
Update-Group Replication

- An update group is a collection of peers with identical outbound policy.
  - Mostly iBGP, mostly RRs!

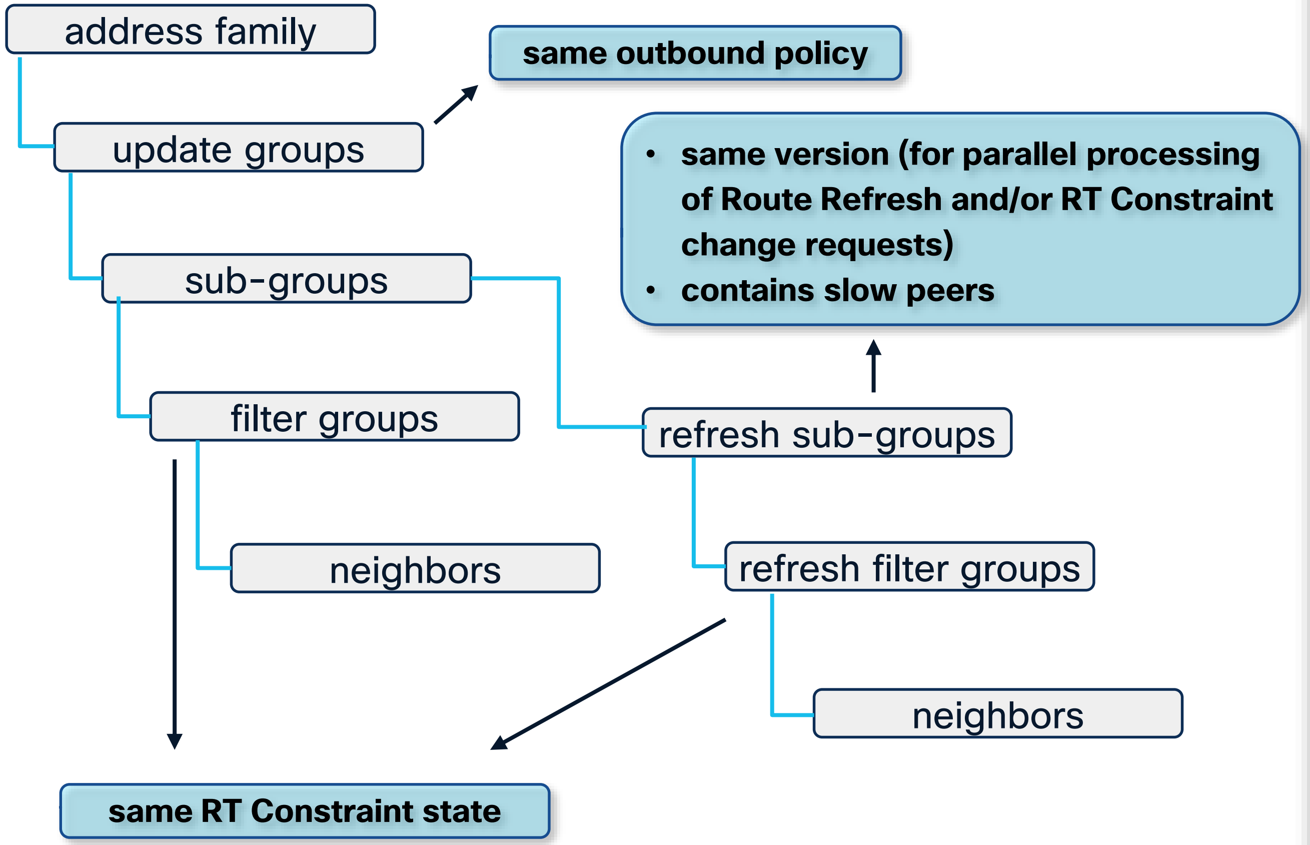
- When generating updates, the group policy is used to format messages that are then replicated and transmitted to the members of the update group.



The reason update groups were introduced



Update Group Hierarchy



- same version (for parallel processing of Route Refresh and/or RT Constraint change requests)
- contains slow peers

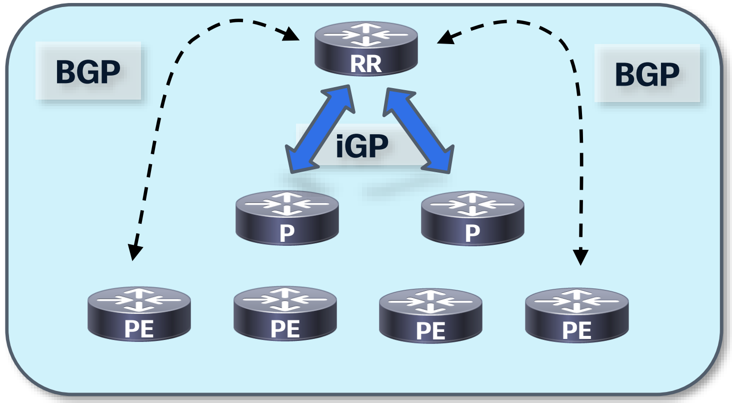
```
RP/0/6/CPU0:router#show bgp vpnv4 unicast update-group 0.2 performance-statistics

Update group for VPNv4 Unicast, index 0.2:
  Attributes:
    Internal
    Common admin
    First neighbor AS: 1
    Send communities
    Send extended communities
    Route Reflector Client
    4-byte AS capable
    Send AIGP
    Minimum advertisement interval: 0 secs
  Update group desynchronized: 0
  Sub-groups merged: 5
  Number of refresh subgroups: 0
  Messages formatted: 36, replicated: 68
  All neighbors are assigned to sub-group(s)
    Neighbors in sub-group: 0.2, Filter-Groups num:3
     Neighbors in filter-group: 0.3(RT num: 3)
      10.1.100.1
     Neighbors in filter-group: 0.1(RT num: 3)
      10.1.100.2
     Neighbors in filter-group: 0.2(RT num: 3)
      10.1.100.8

  Updates generated for 0 prefixes in 26 calls(best-external:0) (time spent: 0.002
secs)
  Update timer last started: Apr  3 08:44:21.425
  Update timer last stopped: not set
  Update timer last processed: Apr  3 08:44:21.435
```

Dedicated vs Inline RR

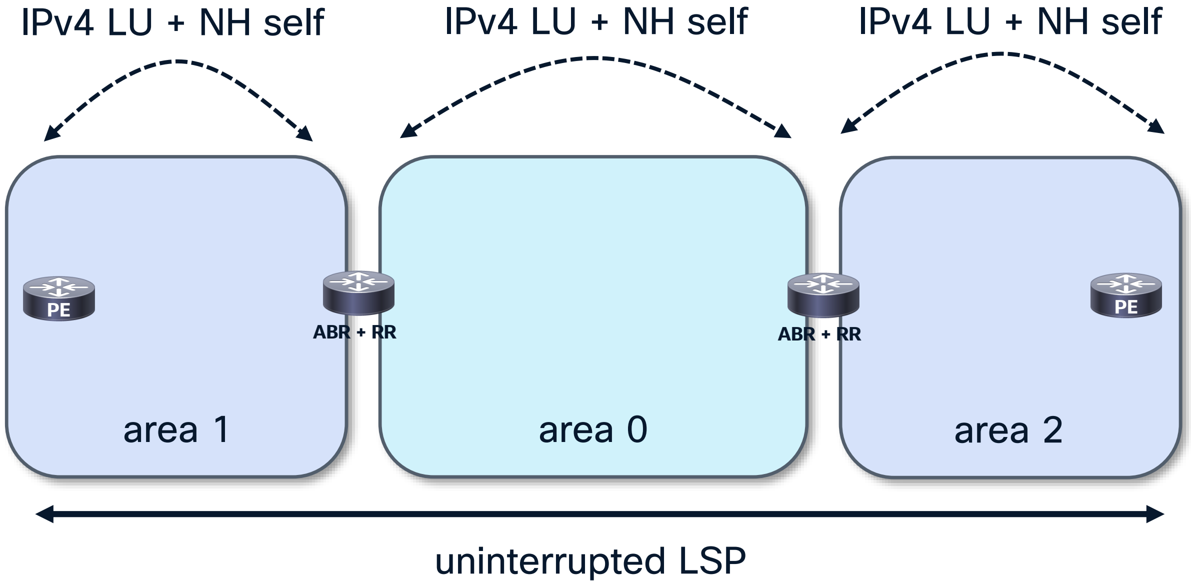
- Dedicated RR = BGP (and IGP) only!
  - No forwarding through the RR
- Inline RR: RR + A(S)BR role
- Any router can be RR: e.g. sometimes PE is also RR

IPv4 Labelled Unicast = RFC 3107

dedicated RR



inline RR

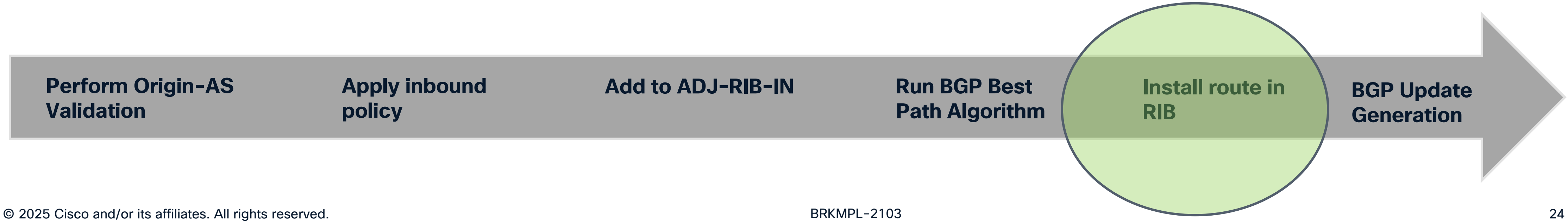


No RIB Download for Dedicated RR

- Selective RIB Download
  - Block all/most BGP routes from installment in the RIB on RR
  - Via BGP table-policy
  - Implemented as filter extension to table-map command
  - For AFs IPv4/6
  - Not needed for AFs VPNv4/6

```
route-policy block-into-rib
 if destination in (...) then
   drop
 else
   pass
 end-if
```

```
router bgp 1


 address-family ipv4 unicast
  table-policy block-into-rib
```



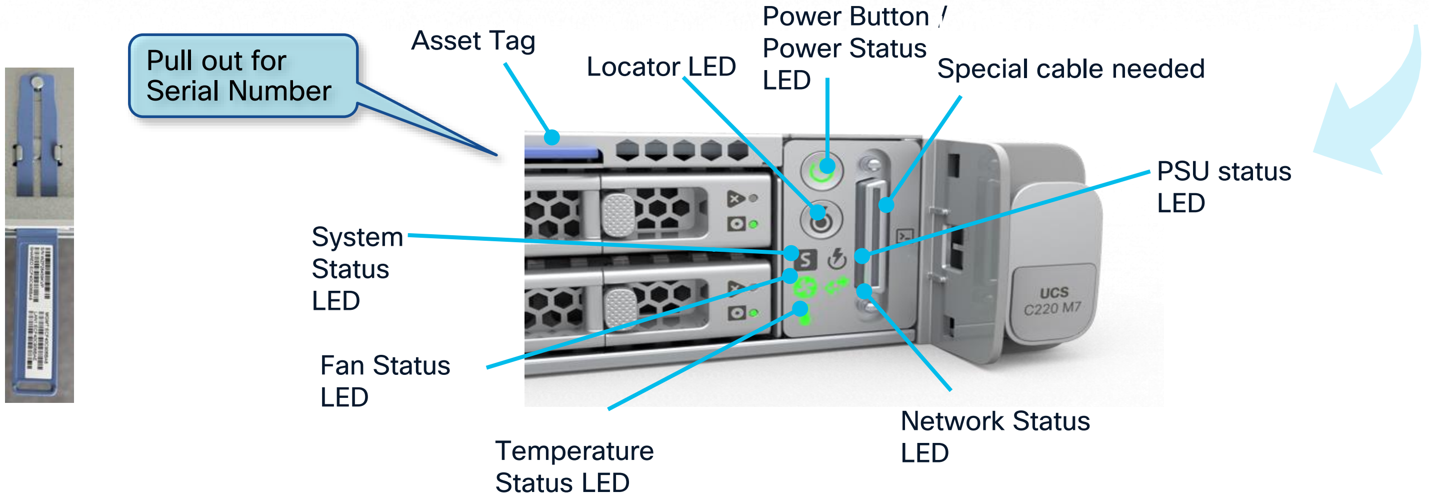
Real vs Virtual RR

- Real router RR: any platform
- Virtual RR:
  - XRv9k (XRd)
    - On KVM
    - Easily manageable for memory, CPU, maintenance window
  - Appliance
    - Dedicated RR
    - On UCS (baremetal)
      - Managed like UCS: CIMC

		BAREMETAL	HYPERVISOR MODE
1	PERFORMANCE	• Faster Disk I/O • Utilizes full CPU & Memory of System	• Lesser performance compared to Appliance based vRR
2	SUPPORT	• Single vendor support for NFVi infrastructure and mounted VNF, easier to reimage	• Support requirements from NFVI vendor and VNF vendor
3	MULTI-APP	• Can support only single application	• Can support Multi-VMs and Applications Cannot benefit from extra HW (TPM, smart NIC)
4	COSTS	• Built for Performance; Slightly Expensive	• Built for flexibility; less expensive
5	FLEXIBILITY	• Fixed Scale, Fixed Resource Mapping for Application	• Variable resource allocation; flexible

Latest Appliance: XRv9k UCS M7

- Fully integrated IOS XRv 9000 router running over Cisco UCS hardware server, out of factory
  - Bare Metal: no need to operate, maintain & optimize NIC drivers/virtualization layer/firmwares
  - Behaves/managed like a regular IOS-XR router
  - But on steroids for BGP RR function: augmented CPU & RAM for optimal scale & convergence
- Two versions available:
  - XRV-M7-APLN-25G: 4x10G/25G ports
  - XRV-M7-APLN-100G: 4x100G

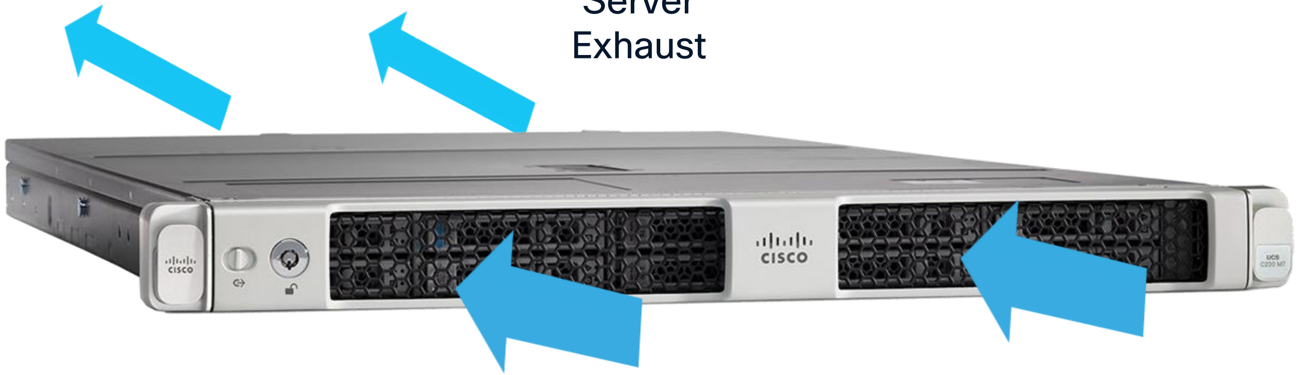


Latest Appliance: XRv9k UCS M7

- 1 x UCS-CPU-I5420+, Intel(R) Xeon(R) Gold 2S 5420+
  - Base frequency: 2GHz
  - Max frequency: 4.10Ghz
  - 28 Cores, 56 threads
  - 52.5MB Cache
  - DDR5 4400MT/s
  - TDP: 205W

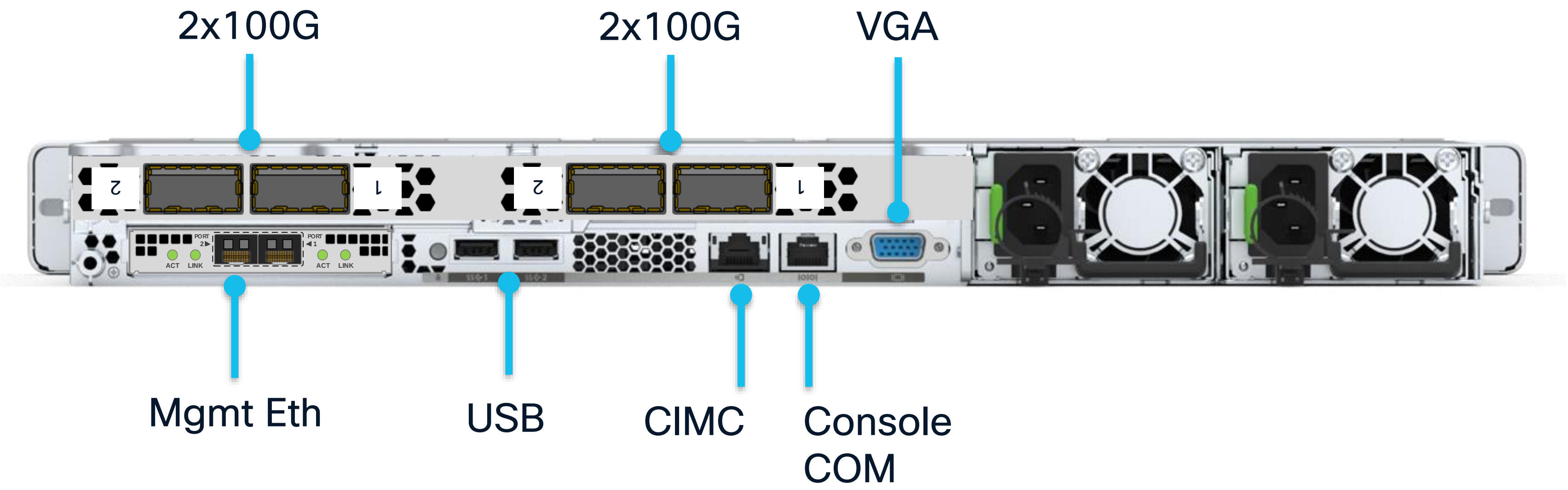
- 128GB DDR5

- Airflow (8 FANs)



- Power supply
  - 2 x 1200W AC or 2 x 1050W DC

Appliance Access



2x100G

2x100G

VGA

Mgmt Eth

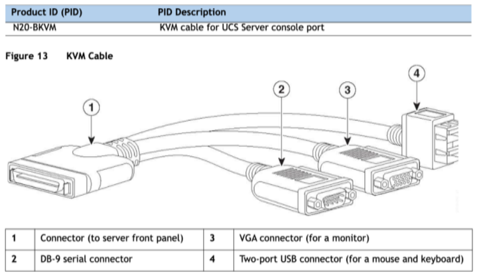
USB

CIMC

Console
COM

KVM options

- Option 1: Use CIMC virtual KVM
- Option 2: Use Cisco KVM cable on front



Product ID (PID)	PID Description
N20-BKVM	KVM cable for UCS Server console port

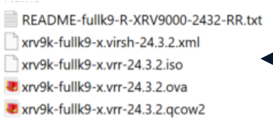
Figure 13 KVM Cable

1	Connector (to server front panel)	3	VGA connector (for a monitor)
2	DB-9 serial connector	4	Two-port USB connector (for a mouse and keyboard)

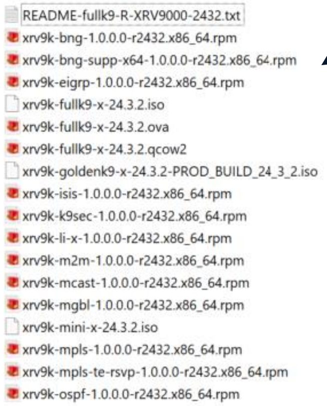
- Option 3: USB keyboard + VGA port on back

Cisco IOS XRv 9000 Profiles

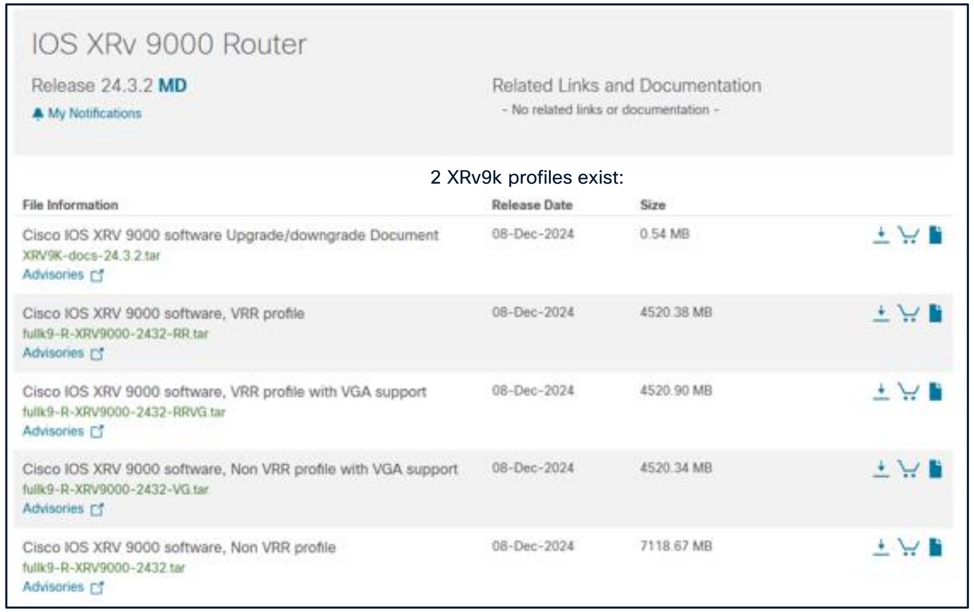
- 2 XRv9k profiles exist
  - VRR
    - Focus on Control Plane
  - VPE
    - Focus on Forwarding Plane



```
README-fullk9-R-XRV9000-2432-RR.txt
xrv9k-fullk9-x.virsh-24.3.2.xml
xrv9k-fullk9-x.vrr-24.3.2.iso
xrv9k-fullk9-x.vrr-24.3.2.ova
xrv9k-fullk9-x.vrr-24.3.2.qcow2
```



```
README-fullk9-R-XRV9000-2432.txt
xrv9k-bng-1.0.0.0-r2432.x86_64.rpm
xrv9k-bng-supp-x64-1.0.0.0-r2432.x86_64.rpm
xrv9k-eigrp-1.0.0.0-r2432.x86_64.rpm
xrv9k-fullk9-x-24.3.2.iso
xrv9k-fullk9-x-24.3.2.ova
xrv9k-fullk9-x-24.3.2.qcow2
xrv9k-goldenk9-x-24.3.2-PROD_BUILD_24_3_2.iso
xrv9k-isis-1.0.0.0-r2432.x86_64.rpm
xrv9k-k9sec-1.0.0.0-r2432.x86_64.rpm
xrv9k-li-x-1.0.0.0-r2432.x86_64.rpm
xrv9k-m2m-1.0.0.0-r2432.x86_64.rpm
xrv9k-mcast-1.0.0.0-r2432.x86_64.rpm
xrv9k-mgbl-1.0.0.0-r2432.x86_64.rpm
xrv9k-mini-x-24.3.2.iso
xrv9k-mpls-1.0.0.0-r2432.x86_64.rpm
xrv9k-mpls-te-rsvp-1.0.0.0-r2432.x86_64.rpm
xrv9k-ospf-1.0.0.0-r2432.x86_64.rpm
```



IOS XRv 9000 Router

Release 24.3.2 **MD**

My Notifications

Related Links and Documentation

- No related links or documentation -

2 XRv9k profiles exist:

File Information	Release Date	Size
Cisco IOS XRV 9000 software Upgrade/downgrade Document XRV9K-docs-24.3.2.tar Advisories	08-Dec-2024	0.54 MB
Cisco IOS XRV 9000 software, VRR profile fullk9-R-XRV9000-2432-RR.tar Advisories	08-Dec-2024	4520.38 MB
Cisco IOS XRV 9000 software, VRR profile with VGA support fullk9-R-XRV9000-2432-RRVG.tar Advisories	08-Dec-2024	4520.90 MB
Cisco IOS XRV 9000 software, Non VRR profile with VGA support fullk9-R-XRV9000-2432-VG.tar Advisories	08-Dec-2024	4520.34 MB
Cisco IOS XRV 9000 software, Non VRR profile fullk9-R-XRV9000-2432.tar Advisories	08-Dec-2024	7118.67 MB

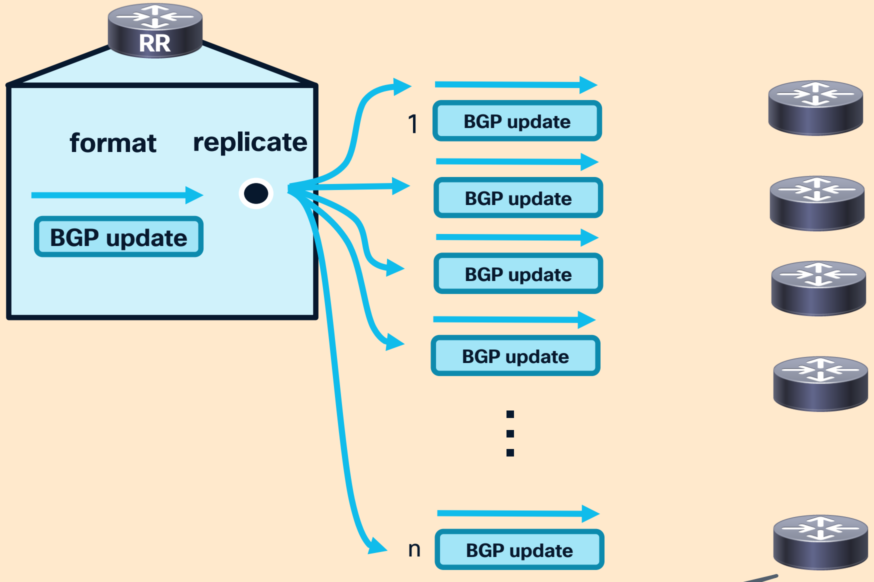
The appliance is always VRR

BGP Slow Peer

- BGP update generation uses the concept of update groups to optimize performance.

- What If one of the peer is not able to process the update fast enough?

- A slow peer is a peer that cannot keep up with the rate at which the router is generating update messages over a prolonged period of time.
  - The slow peer slows down the BGP processing of all peers in that update group.
  - A slow peer has a large Output Queue
  - Slow-peer detection is enabled by default; handling is not
  - Slow peers are moved to a separate refresh sub-group (peer must be slow for 5 min)



Peer **'n'** cannot **process BGP updates** at the **rate** it receives them.

Slow Peer Mitigation Handling

- Configuration

```
router bgp 65001
 slow-peer dynamic threshold 120
```

Enable dynamic slow peer handling and a threshold time of 120 sec (default is 300 sec)

```
router bgp 65001
 neighbor 10.7.15.15
  address-family ipv4 unicast
   slow-peer static
```

You can configure a static slow peer

- Syslog messages

```
bgp[1079]: %ROUTING-BGP-5-AF_SLOW_PEER : BGP neighbor 10.0.0.2 of vrf default afi 0 is detected as slow-peer
```

```
bgp[1079]: %ROUTING-BGP-5-AF_SLOW_PEER_RECOVERED : Slow BGP peer 10.0.0.2 of vrf default afi 0 has recovered
```

Slow Peer Troubleshooting

```
RP/0/RP0/CPU0:RR7#show bgp update out neighbor

VRF "default", Address-family "IPv4 Unicast"
  Main routing table version: 1500037
  RIB version: 1500037

Legend: (S) - Slow peer static configured
        (D) - Slow peer dynamic detected

  Neighbor        FG     SG      SG-R          UG     Status OutQ      OutQ-R     Version      Ack/Ack-R

  10.0.0.1        0.1    0.1     ---           0.4    Normal 0         0          1500037      1461343
  10.0.0.2        0.1    0.1     ---           0.4    Normal 0         0          1500037      1461343
  10.0.0.4        ---    ---     ---           0.4    Normal 0         0          0            0
  10.0.0.5        0.1    0.1     ---           0.4    Normal 0         0          1500037      1461343
  10.7.14.14      ---    ---     ---           0.3    Normal 0         0          0            1
  10.7.15.15      0.1    0.1     0.1:1         0.4    Normal 1832100   891300     1500037      37/0 (D)
```

```
RP/0/RP0/CPU0:RR7#show bgp update-group 0.1
Update group for IPv4 Unicast, index 0.1:
  Attributes:
    Neighbor sessions are IPv4
    …
    Contains Slow peers
    Minimum advertisement interval: 0 secs
  Update group desynchronized: 0
  Sub-groups merged: 0
  Number of refresh subgroups: 0
  Messages formatted: 5760, replicated: 5760
  All neighbors are assigned to sub-group(s)
    Neighbors in sub-group: 0.3, Filter-Groups num:1
     Neighbors in filter-group: 0.6(RT num: 0)
      10.0.0.2
```

```
RP/0/RP0/CPU0:RR7#show bgp all all update out neighbor slow-peers brief
Address Family: IPv4 Unicast
----------------------------
VRF "default", Address-family "IPv4 Unicast"
  Main routing table version: 2100037
  RIB version: 2100037

Legend: (S) - Slow peer static configured
        (D) - Slow peer dynamic detected

  Neighbor        FG     SG      SG-R          UG     Status OutQ      OutQ-R     Version      Ack/Ack-R

  10.7.15.15      0.1    0.1     0.1:1         0.4    Normal 1939100   876400     2100037      37/0 (D)
```

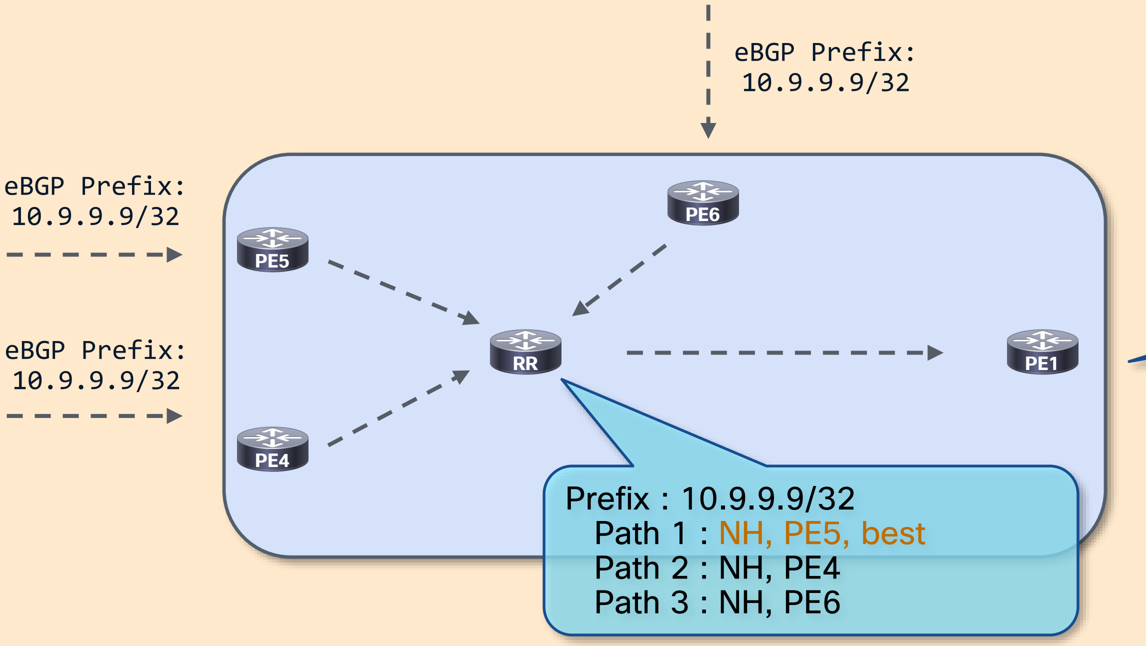
Refresh sub-group under existing update-group

non-zero Output Queue for a long time = indication of a slow peer

BGP Optimal Route Reflection

Optimal Route Reflection (ORR)

- RR has an “IGP” location in the network
- RR sends its best route, which might not be the best route from ingress to egress BGP peer

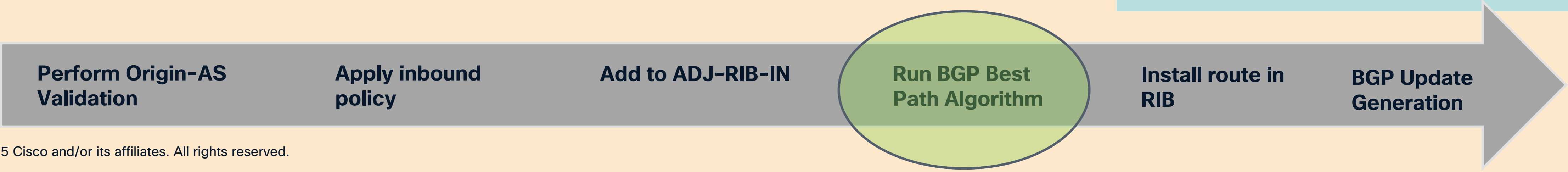


Prefix : 10.9.9.9/32
Path 1 : NH, PE5, best
Path 2 : NH, PE4
Path 3 : NH, PE6

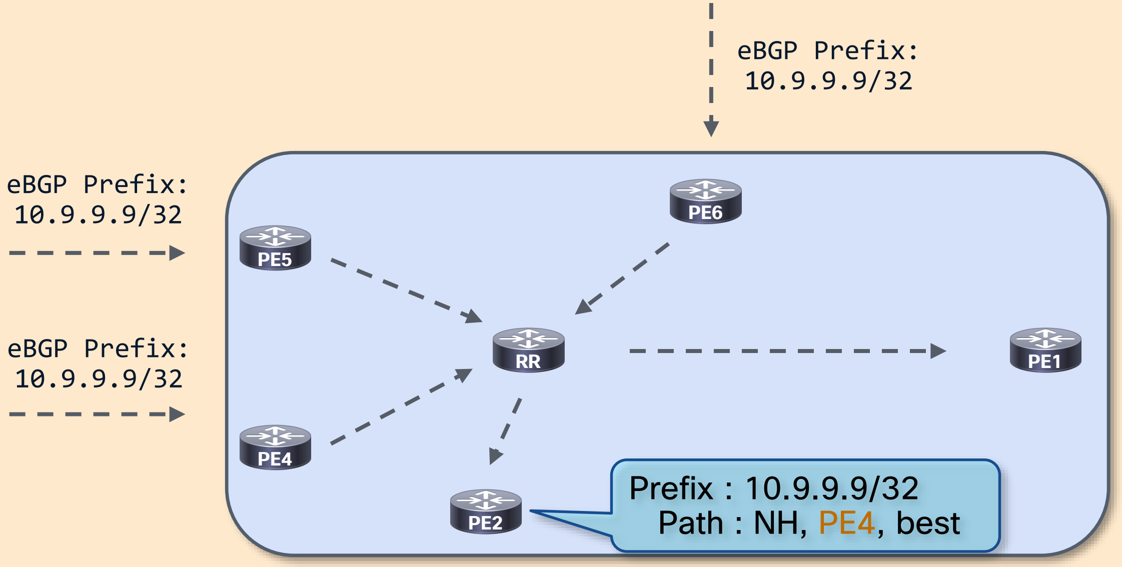
But PE6 is closer (IGP-wise) than PE5 as exit router

Prefix : 10.9.9.9/32
Path : NH, PE5, best

- Solution: Have **the RR calculate different best paths from the viewpoint of the RR client** and advertise those paths to each RR client
- Requirements:
  1. Link State Routing protocol (do perform reverse SPF with **root** the RR client)
  2. Only configure ORR on the RR (not the clients)
  3. Redistribute link-state into IGP on RR



ORR



Prefix : 10.9.9.9/32
Path : NH, PE4, best

Prefix : 10.9.9.9/32
Path : NH, PE6, best

Cost from the root (10.0.0.1) to IGP prefixes in this AS.
Only the BGP next hop address are important for ORR.

```
router bgp 65001
 optimal-route-reflection ipv4 ipv4-orr-group-1 10.0.0.1
 address-family ipv4 unicast
  optimal-route-reflection apply ipv4-orr-group-1
 !
 neighbor 10.0.0.4
  remote-as 65001
  update-source Loopback0
  address-family ipv4 unicast
   optimal-route-reflection ipv4-orr-group-1

router isis 65001
 distribute link-state level 2
```

```
RP/0/RP0/CPU0:RR#show orrspf database ipv4-orr-group-1

ORR policy: ipv4-orr-group-1, IPv4, RIB tableid: 0xe0000001
Configured root: primary: 10.0.0.1, secondary: NULL, tertiary: NULL
Actual Root: 10.0.0.1, Root node: 0000.0000.0001.0000

Prefix                                        Cost
10.0.0.1/32                                   10
10.0.0.2/32                                   50
10.0.0.3/32                                   30
10.0.0.4/32                                   50
10.0.0.5/32                                   40
10.0.0.6/32                                   20
10.1.3.0/24                                   20
10.1.6.0/24                                   10
…
Number of mapping entries: 14
```

ORR: BGP Output

```
RP/0/RP0/CPU0:RR#show bgp ipv4 unicast 10.9.9.9/32
BGP routing table entry for 10.9.9.9/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 10          10
Last Modified: Apr 10 14:31:21.786 for 00:11:07
Paths: (3 available, best #2)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Path #1: Received by speaker 0
  ORR bestpath for update-groups (with more than one peer):
    0.5
  Local, (Received from a RR-client)
    10.0.0.4 (metric 30) from 10.0.0.4 (10.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal, add-path
      Received Path ID 0, Local Path ID 4, version 10
  Path #2: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Local, (Received from a RR-client)
    10.0.0.5 (metric 20) from 10.0.0.5 (10.0.0.5)
      Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 1, version 8
  Path #3: Received by speaker 0
  ORR bestpath for update-groups (with more than one peer):
    0.2
  Local, (Received from a RR-client)
    10.0.0.6 (metric 30) from 10.0.0.6 (10.0.0.6)
      Origin IGP, metric 0, localpref 100, valid, internal, add-path
      Received Path ID 0, Local Path ID 5, version 10
```

ADD PATH is not enabled.
Path IDs are used by ORR

ORR bestpath: path with next-hop 10.0.0.4 advertised to 1 RR client (configured as root for ORR)

ORR bestpath: path with next-hop 10.0.0.6 advertised to 1 RR client (configured as root for ORR)

ORR and Traffic Engineering

- To make ORR work, minimal Traffic Engineering is needed on the root(s)
  - MPLS TE is enabled in the specific ISIS level
  - The MPLS TE router-ID is configured matching the configured root address on the RR
  - MPLS TE is configured on at least one interface
  - There is no need for other MPLS TE configuration or RSVP anywhere!

root configuration

```
router isis 65001
 address-family ipv4 unicast
  mpls traffic-eng level-2-only
  mpls traffic-eng router-id Loopback0
```

```
mpls traffic-eng
```

```
IS-IS 1 (Level-2) Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime/Rcvd  ATT/P/OL
PE1.00-00             0x00000019   0xcf63        1194 /1200         0/0/0
  Area Address:   49.0001
  Metric: 20         IS-Extended RR3.00
    Physical BW: 1000000 kbits/sec
  Metric: 10         IS-Extended PE6.00
    Physical BW: 1000000 kbits/sec
  NLPID:          0xcc
  IP Address:     10.0.0.1
  Router ID:      10.0.0.1
```

ISIS Router ID TLV is advertised

BGP Route Policy

Why RPL (Route Policy Language)

Scaling

- Using route-maps could lead to 100k – 1M lines of configuration (e.g. 1.000s of BGP peers)

Modularity

- Exploit modularity to reuse common portions of configuration

Parameterization

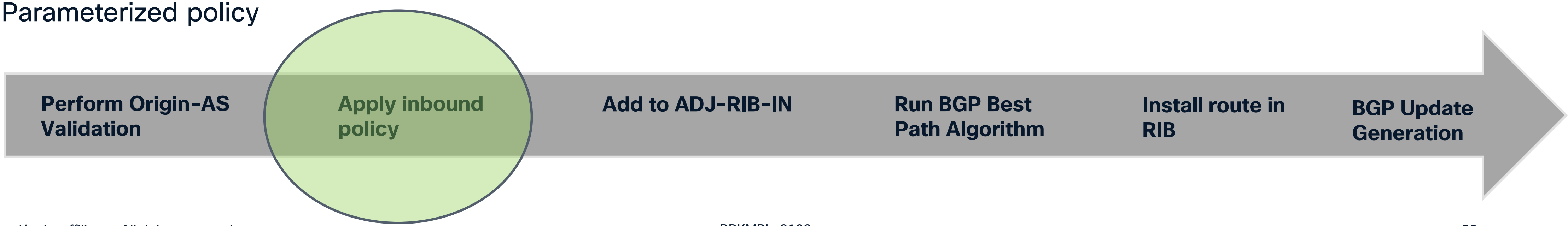
- For elements which are not exact copies of each other we can add parameterization (think variables) to get further re-use

Improved Clarity

- No silently skipped statements
- What you see is what you get!

Human-readable

- Hierarchical policy
- Parameterized policy



RPL BGP Attach Points

- Policy Attach Points are the points where an association is formed between a specific protocol entity, in this case a BGP neighbor, and a specific named policy.
- Implicit drop if no match/set

attachment point

```
router bgp 65500
 neighbor 10.2.3.4
  address-family ipv4 unicast
   route-policy foo in
   route-policy bar out
```

- Neighbor inbound
- Neighbor outbound
- Neighbor ORF
- Aggregation
- Default originate
- Dampening
- Redistribution
- Import
- Export
- Retain RT
- Allocate-label
- Table policy
- Network command
- Some/debug BGP commands

RPL Examples

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

if, then, elseif, else

```
route-policy one
    set med 100
end-policy

route-policy two
    apply one
    set community (10:100)
end-policy
```

hierarchical

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

nested if

```
route-policy one ($med)
    set med $med
end-policy

route-policy two
    apply one (10)
end-policy
```

parameterized

no looping or recursion allowed

RPL Global Variable

Global variable to be used across route policies

```
policy-global
   PRIMARY '1’
end-global
```

A change of the value here is propagated to multiple route policies using this global variable

```
route-policy XXX_IN
  var globalVar1 $PRIMARY
  if globalVar1 is 1 then
    set local-preference 150
  elseif globalVar1 is 2 then
    set local-preference 110
  endif
end-policy
```

Using RPL

Hot troubleshooting tip!

Use RPL to match prefixes: show or debug commands

Match certain RT in show command

```
extcommunity-set rt ext1
  4:3
end-set

route-policy ext_rp1
  if extcommunity rt matches-any ext1 then
    pass
  else
    drop
  endif
end-policy
```

```
RP/0/0/CPU0:R1#show bgp vpnv4 unicast policy route-policy ext_rp1
Route Distinguisher: 1:3
50.1.1.0/24 is advertised to 10.0.101.1
  Path info:
    neighbor: 10.3.101.1      neighbor router id: 10.3.101.1
    valid  external  best  multipath  import-candidate
Received Path ID 0, Local Path ID 1, version 6
  Attributes after inbound policy was applied:
    next hop: 10.3.101.1
    ORG AS EXTCOMM
    origin: IGP  neighbor as: 1001
    aspath: 1001
    extended community: RT:4:3
  Attributes after outbound policy was applied:
    next hop: 0.0.0.0
    ORG AS COMM EXTCOMM
    origin: IGP  neighbor as: 1001
    aspath: 1 1001
    community: graceful-shutdown
    extended community: RT:4:3
```

Limit debug output for updates with prefixes in the range 199.1.1.0/25 eq 32/

```
RP/0/0/CPU0:R1#show rpl route-policy ldg
route-policy ldg
  if destination in (199.1.1.0/25 eq 32) then
    pass
  endif
end-policy

RP/0/0/CPU0:R1#debug bgp update ipv4 unicast route-policy ldg in

RP/0/0/CPU0:R1#show debug
####  debug flags set from tty 'vty0'  ####
ip-bgp update flag is ON with value '#ipv4#unicast#in###ldg####'
```

The RPL does not need an attachment point

Check RPL Before Applying It

```
RP/0/RP0/CPU0:R1#show bgp neighbors 10.1.4.4 dryrun-policy new-rpl

Policy Statistics
-----------------

     AFI:                          IPv4 Unicast
     Direction:                    Inbound
     In-use Policy:                neighbor_10_1_4_4_in
     Dry-run Policy:               new-rpl
     Remote-as:                    65003
     Total Networks walked:        350
     Total Paths walked:           382
     Dry-run elapsed time(ms):     3
     Dry-run request complete:     True
---------------------------------------------------------------------------------------
                                        Dry-run-Policy      In-use-Policy      Delta
---------------------------------------------------------------------------------------
 Neighbor: 10.1.4.4
     Accepted Unmodified:               250                 154                 96
     Accepted Modified:                 0                   32                  -32
       Pre-inbound policy copy:         0                   32                  -32
     Denied:                            0                   64                  -64
     Estimated Total Paths Memory:      25.39KB             28.64KB             -3.25KB
---------------------------------------------------------------------------------------
```

```
RP/0/RP0/CPU0:R1#show bgp scale detail

VRF: default
 Neighbors Configured: 3      Established: 2

 Address-Family    Prefixes Paths    PathElem    Prefix      Path        PathElem
                                                 Memory      Memory      Memory
  IPv4 Unicast     350      382      350         64.26KB     38.80KB     43.41KB
   SoftReconfig Changed     32                               3.25KB
  ------------------------------------------------------------------------------
  Total            350      382      350         64.26KB     38.80KB     43.41KB

Total VRFs Configured: 0
```

check the difference with the existing route-policy and the new route-policy before you apply it

Check Performance of RPL

- PCL = Policy Clientlib Information
- Policy profiling tool for route policies which can be used without impact on performance in order to measure the time spent in each statement of a route policy at a specific attach point.
- You can check the run time of the route policy at this specific attach point.
- Works for policy IN or OUT
- By default, the profiling is enabled only for aggregate route policy stats.

```
RP/0/RP0/CPU0:R1#show pcl protocol bgp speaker-0 ?
  debug-policy                 Attachpoint name
  permnet                      Attachpoint name
  import                       Attachpoint name
  export                       Attachpoint name
  interafi-import              Attachpoint name
  source-rt                    Attachpoint name
  interafi-export              Attachpoint name
  retain-rt                    Attachpoint name
  addpath                      Attachpoint name
  neighbor-in-dflt             Attachpoint name
  neighbor-in-vrf              Attachpoint name
  neighbor-out-dflt            Attachpoint name
  neighbor-out-vrf             Attachpoint name
  orf-dflt                     Attachpoint name
  orf-vrf                      Attachpoint name
  dampening-dflt               Attachpoint name
  dampening-vrf                Attachpoint name
  default-originate-dflt       Attachpoint name
…
```

clearing the stats

```
RP/0/RP0/CPU0:R1#clear pcl protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6
policy profile
```

Check Performance of RPL

enabling debug pcl profile to get more detailed stats

```
RP/0/RP0/CPU0:R1#debug pcl profile detail
```

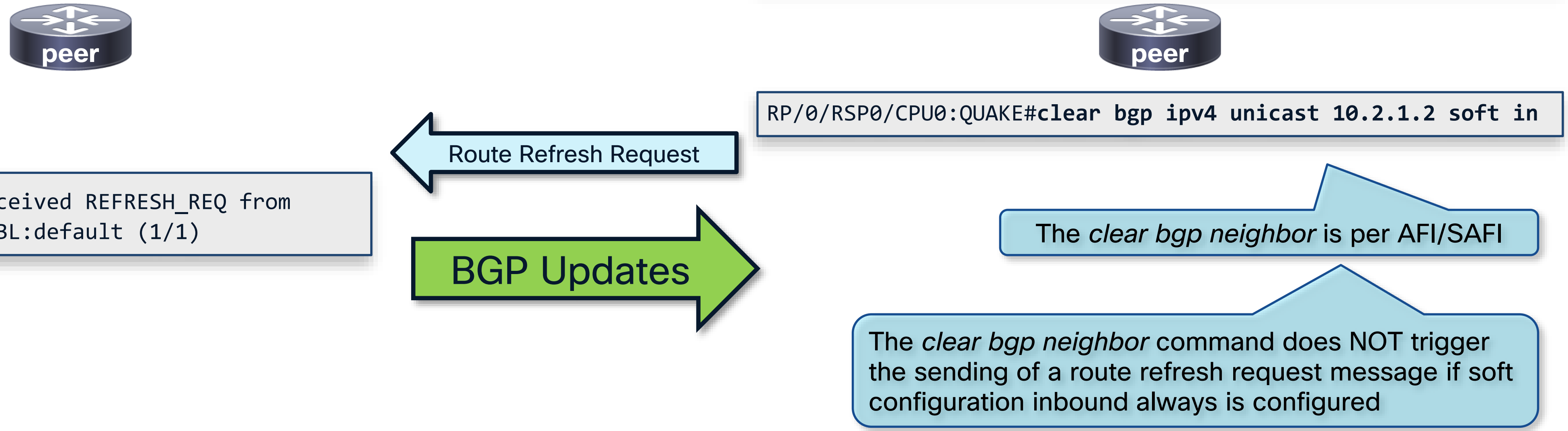
```
RP/0/RP0/CPU0:R1#show pcl protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile
Policy profiling data
Policy : INGRESS-ROUTE-POLICY
Pass : 720100
Drop : 0
# of executions : 720100
Total execution time : 222788msec !!!! about 3.7 minutes to process ingress updates

Node Id   Num visited      Exec time  Policy engine operation
--------------------------------------------------------------------------------
PXL_0_1        720100     221796msec  if as-path aspath-match ... then
                                       <truePath>
PXL_0_3          3525          3msec    set local-preference 150
                 3525          0msec    <end-policy/>
                                       </truePath>
                                       <falsePath>
PXL_0_2        716575        225msec    set local-preference 50
               716575         82msec    <end-policy/>
                                       </falsePath>
```

BGP Soft Reconfig

Route-Refresh

- A hard reset is to clear the neighbor that would lead to the router re-learning the routes from its neighbor
- Route Refresh capability
  - The original routes are not stored because they can be retrieved from the neighbor through a route refresh request
  - No hard reset required
- A route refresh can be triggered:
  - Automatically by the router
  - Manually with ***clear bgp ...***

```
RP/0/RP0/CPU0:R1#show bgp neighbor 10.1.4.4
 For Address Family: IPv4 Unicast
  Route refresh request: received 0, sent 0
  Policy for incoming advertisements is neighbor_10_1_4_4_in
  186 accepted prefixes, 186 are bestpaths
  Exact no. of prefixes denied: 64
  Cumulative no. of prefixes denied: 64
    No policy: 0, Failed RT match: 0
    By ORF policy: 0, By policy: 64
```



```
RP/0/RSP0/CPU0:QUAKE#clear bgp ipv4 unicast 10.2.1.2 soft in
```

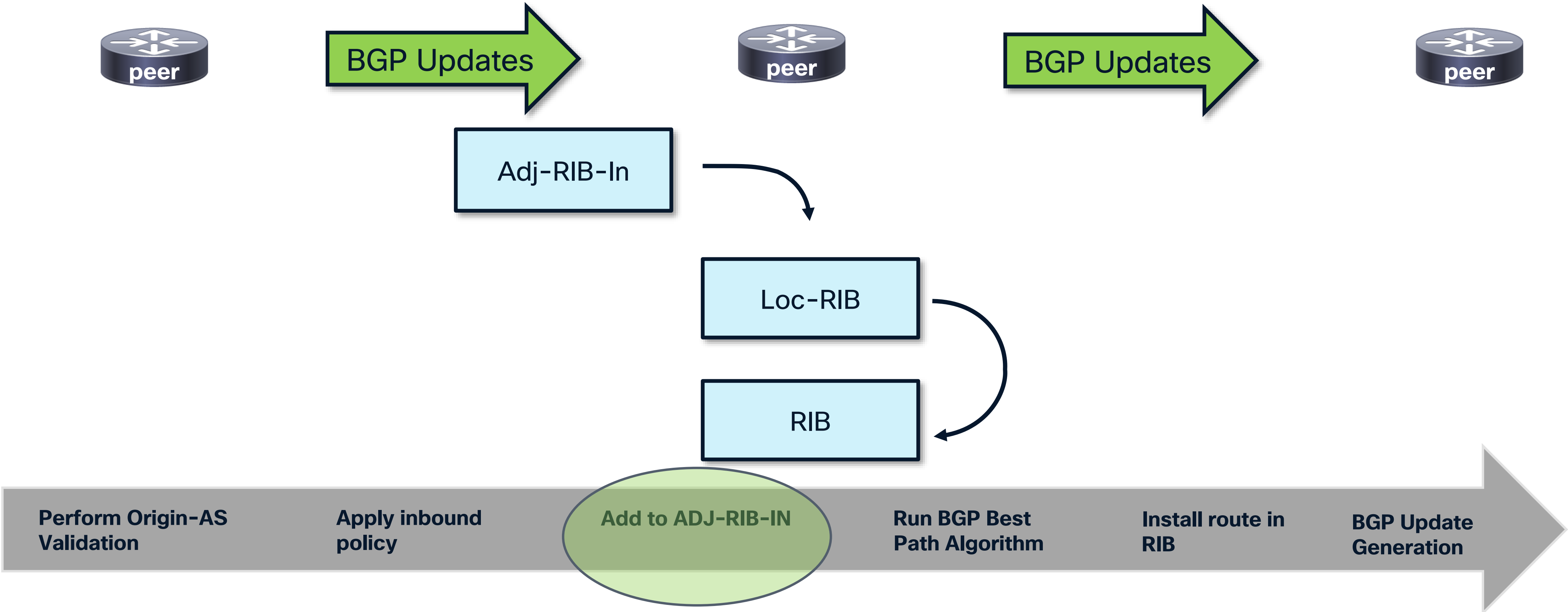
```
bgp[1019]: [default-iord]: Received REFRESH_REQ from
10.2.1.1 for address family TBL:default (1/1)
```

The *clear bgp neighbor* is per AFI/SAFI

The *clear bgp neighbor* command does NOT trigger the sending of a route refresh request message if soft configuration inbound always is configured

Adj-RIB-Out vs Adj-RIB-In

- Adj-RIB-In
  - Unmodified routing information received from the BGP neighbors
  - The inbound RPL will apply the changes to this table and store the result in the BGP table (Loc-RIB)
- Adj-RIB-Out
  - A table holding the routing information to be sent to one neighbor



Adj-RIB-Out vs Adj-RIB-In: All 1 Show BGP Command

```
RP/0/RSP0/CPU0:R1#show bgp ipv4 unicast 198.168.12.1/32

Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    10.0.0.1
  65001, (received & used)
```

passed prefix

rpl action: pass

```
RP/0/RSP0/CPU0:R1#show bgp ipv4 unicast 198.168.12.2/32

Paths: (1 available, no best path)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  65001, (received-only)
```

blocked prefix

rpl action: drop

```
RP/0/RSP0/CPU0:R1#show bgp ipv4 unicast 198.168.12.3/32

Paths: (2 available, best #1)
Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    10.0.0.1
  65001
     Origin IGP, metric 0, localpref 200, valid, external, best, group-best
Path #2: Received by speaker 0
  Not advertised to any peer
  65001, (received-only)
     Origin IGP, metric 0, localpref 100, valid, external
```

rpl action: modify

new modified path

original received path

Soft Reconfig Inbound Overview

```
RP/0/RP0/CPU0:R1#show bgp summary soft-reconfig-stats

Process    RcvTblVer     bRIB/RIB     LabelVer    ImportVer    SendTblVer   StandbyVer
Speaker          138           138           138           138           138             0

Neighbor        Spk          AS    MsgRcvd    MsgSent       TblVer        InQ       OutQ    Up/Down  St/PfxRcd    SoftChgd    Denied
10.0.0.2          1       65001        219        237          582          0          0   03:26:45        100            0         0
10.0.0.4          1       65001          0          0            0          0          0   00:00:00  Idle
10.1.4.4          1       65003        241        230          582          0          0   00:03:28        186           32        64

  Total                                                                                                    286           32        64
Legend:
  Total PfxRcd    : Sum of accepted unmodified and modifed paths
  Total SoftChgd  : Sum of accepted modified paths
  Total Denied    : Sum of denied paths
```

Optimizations

Label Allocation Mode

- Per prefix
  - The default allocation mode
  - 1 unique label per prefix
  - Good for load balancing MPLS traffic (unique hash per flow)
  - Least scalable (risk of running out of MPLS labels)

```
RP/0/RP0/CPU0:R1#show bgp vrf one labels
Network            Next Hop        Rcvd Label      Local Label
*> 10.1.4.0/24        0.0.0.0         nolabel         24000
*> 10.100.0.0/24      10.1.4.4        nolabel         24001
*> 10.100.1.0/24      10.1.4.4        nolabel         24002
*> 10.100.2.0/24      10.1.4.4        nolabel         24003
*> 10.100.3.0/24      10.1.4.4        nolabel         24004
*> 10.100.4.0/24      10.1.4.4        nolabel         24005
```

- Per CE (Customer Edge)

```
R1(config-bgp-vrf)#label-allocation-mode per-ce
```

  - Unique label per (CE) net hop
  - Very scalable
  - Still MPLS lookup only

different attached CE

```
RP/0/RP0/CPU0:R1#show bgp vrf one labels
Network            Next Hop        Rcvd Label      Local Label
*> 10.1.4.0/24        0.0.0.0         nolabel         24000
*> 10.100.0.0/24      10.1.4.4        nolabel         24001
*> 10.100.1.0/24      10.1.5.5        nolabel         24002
*> 10.100.2.0/24      10.1.4.4        nolabel         24001
*> 10.100.3.0/24      10.1.4.4        nolabel         24001
*> 10.100.4.0/24      10.1.4.4        nolabel         24001
```

- Per VRF

```
R1(config-bgp-vrf)#label-allocation-mode per-vrf
```

  - Same label for all prefixes in the VRF
  - Very scalable
  - IP lookup is forced (hence no PIC)
  - Not always good for load balancing MPLS traffic

```
RP/0/RP0/CPU0:R1#show bgp vrf one labels
Network            Next Hop        Rcvd Label      Local Label
*> 10.1.4.0/24        0.0.0.0         nolabel         24000
*> 10.100.0.0/24      10.1.4.4        nolabel         24000
*> 10.100.1.0/24      10.1.5.5        nolabel         24000
*> 10.100.2.0/24      10.1.4.4        nolabel         24000
*> 10.100.3.0/24      10.1.4.4        nolabel         24000
*> 10.100.4.0/24      10.1.4.4        nolabel         24000
```

* Connected and BGP aggregate prefixes always have the same label (“per-vrf aggregate” prefixes)

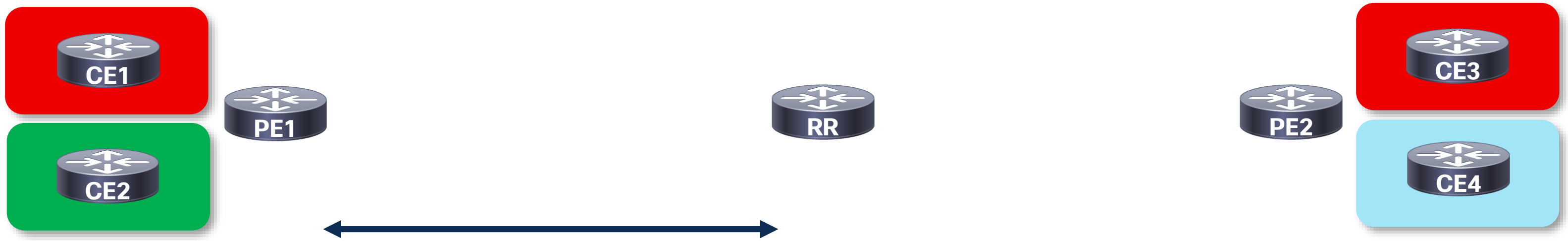
BGP RR Optimization: RTC

- RTC = RT Constraint
  - Constrain the vpnv4/6 routes to the PE routers which need them
  - “need them” = PE routers that have a VRF importing the routes

- Trade-off between
  - Sending all to every RR Client (less processing)
  - Processing and sending (filtering) only to interested RR Clients

amount of BGP updates vs BGP processing time

RT-Constraint - RFC4364



BGP capability exchange OPEN message

capability 1/132 (RTFilter)
for vpnv4 & vpnv6

AF RTFilter exchange

MP_REACH_NLRI
AF RTFilter

origin AS — Route Target 1:2

origin AS — Route Target 1:1

origin AS — Route Target 0:0

PE1 installs
Default RT filter for
RR

RR installs RT filter RT:1:1 &
RT 1:2 for PE1 (implicitly
denying all else)

AF vpnv4/6 prefixes exchange

PE sends all its
vpnv4/6 prefixes
to RR

RR sends only red and green
(not blue) vpnv4/6 prefixes to
PE1

Security

RFC 7454 “BGP Operations and Security” - Overview

- Best Current Practice as of 2015 (RFC 7454 published in 2015)
- Max prefixes on a BGP peering
- Protect BGP sessions
  - Control plane policing
  - MD5/TCP-AO
  - GTSM (Generalized TTL Security Mechanism)
- Dampening
- Prefix filtering
  - Also filtering prefixes that are too specific
- Communities scrubbing
- BGP RPKI

Max Prefixes Limit

- Max number of prefixes allowed from neighbor (post inbound policy)
- Bring down session above limit, each add-path is counted
- Need manual clear to bring up if no restart configured

```
RP/0/RP0/CPU0:R1#show bgp neighbors 10.1.4.4
 For Address Family: IPv4 Unicast
  Maximum prefixes allowed 10000 (discard-extra-paths)
```

```
router bgp 65001
 neighbor 10.1.4.4
  remote-as 65002
  address-family ipv4 unicast
   maximum-prefix 10000 80 restart 20
```

max 10.000 prefixes
syslog warning at 80%
bring down if exceed
restart time interval is 20 min

```
router bgp 65001
 neighbor 10.1.4.4
  remote-as 65002
  address-family ipv4 unicast
   maximum-prefix 10000 90 discard-extra-paths
```

max 10.000 prefixes
syslog warning at 90%
discard extra paths when limit is exceeded

- Max prefixes in RIB (routing protocol independent), any VRF

```
address-family ipv4 unicast
 maximum prefix 1500000
!
address-family ipv6 unicast
 maximum prefix 500000
```

RPKI (Resource Public Key Infrastructure)

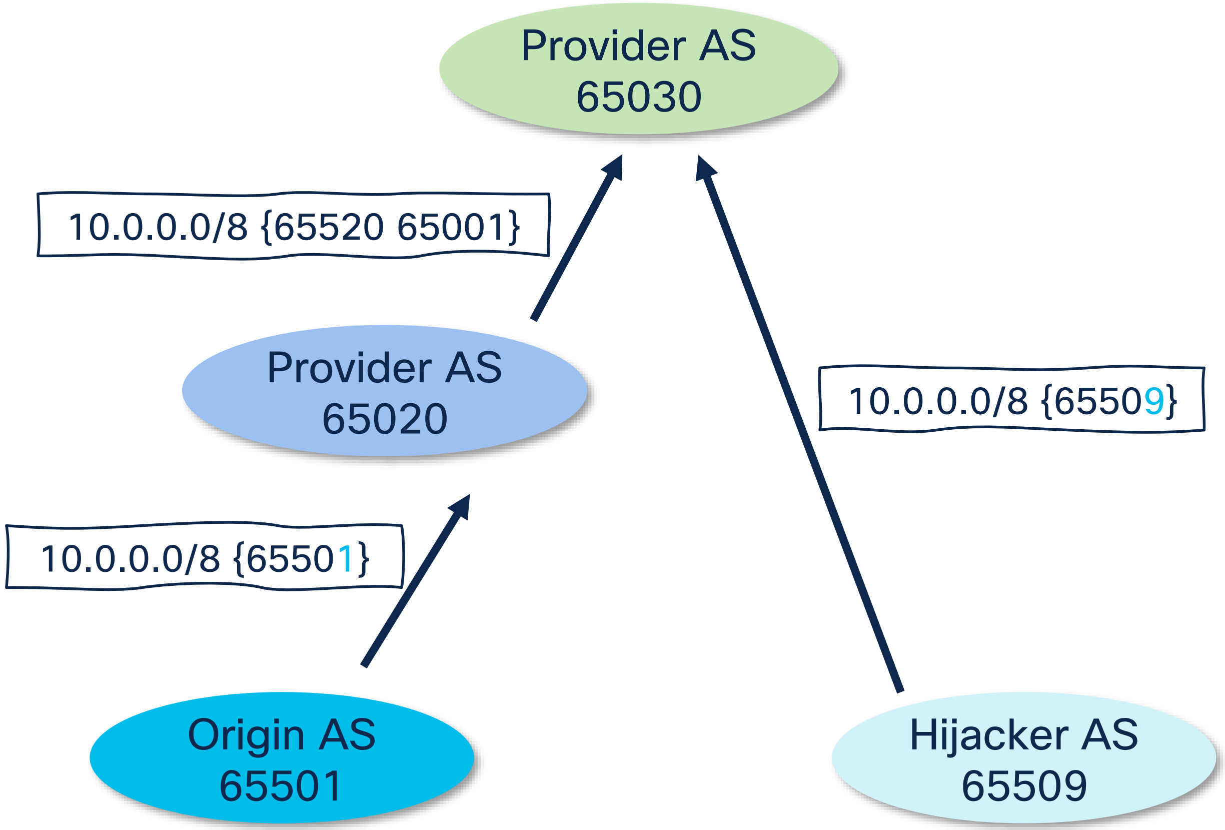
Prefix Hijacking

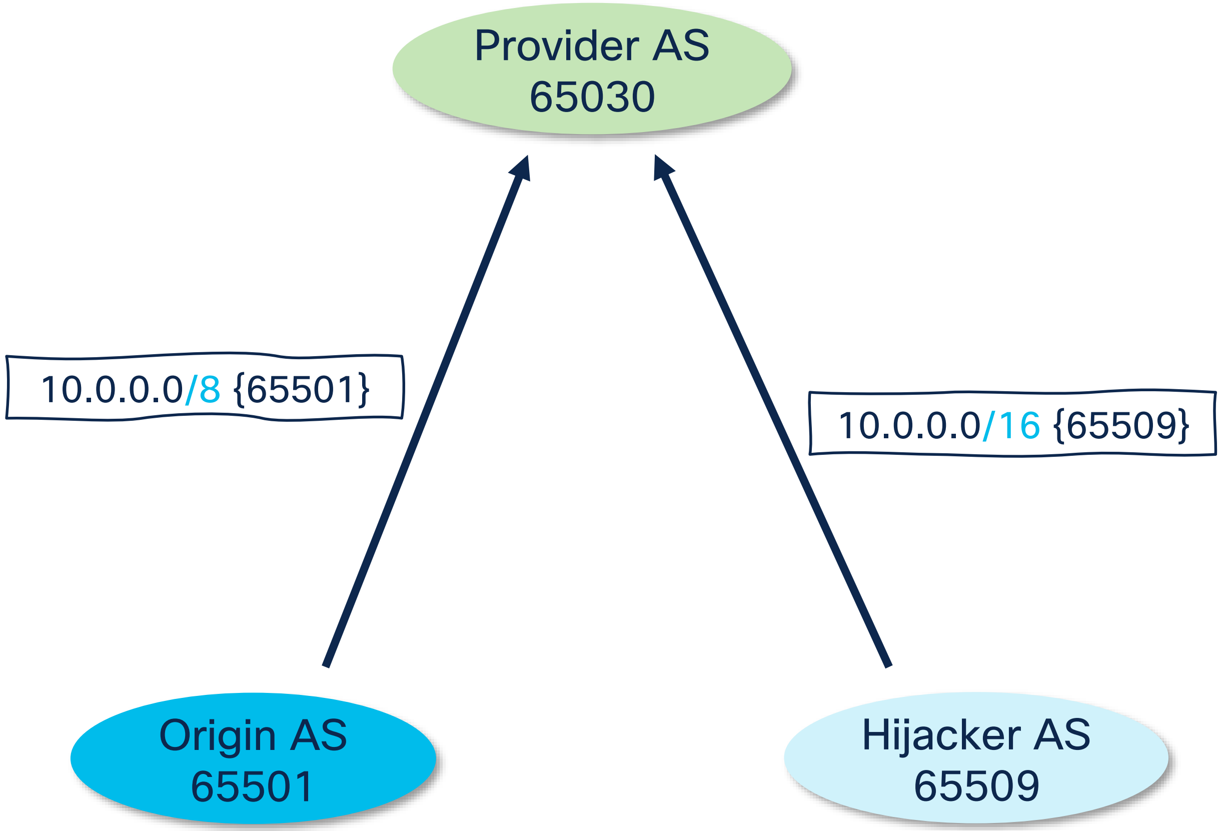
- Same prefix, but shorter AS_PATH length: wins

Intent

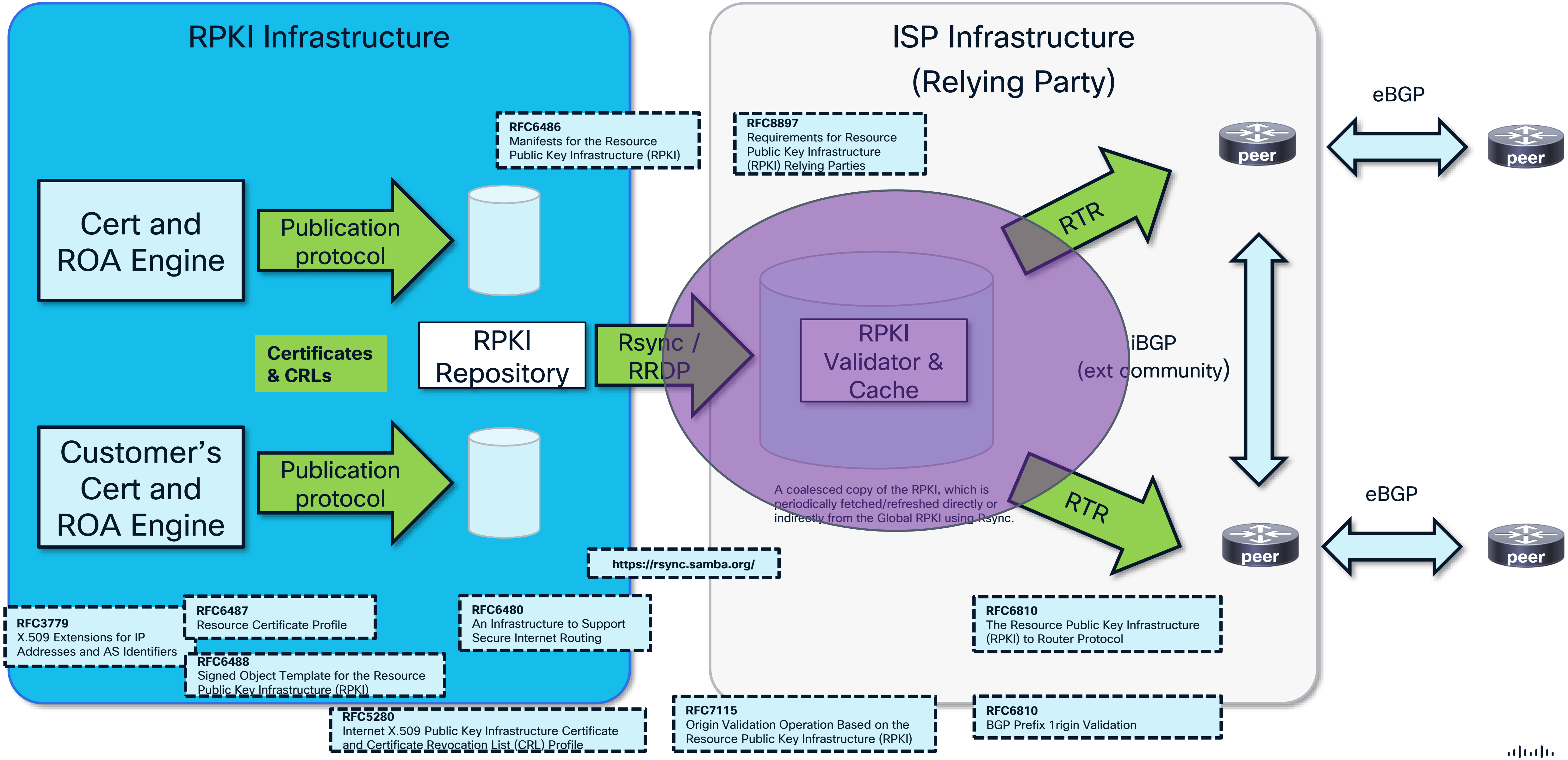
Capture
Inspect
Redirect
Manipulate traffic

- Longer mask: more specific prefix wins





RPKI System



RPKI Infrastructure

Cert and ROA Engine

Publication protocol

Certificates & CRLs

RPKI Repository

Customer’s Cert and ROA Engine

Publication protocol

Rsync / RRDP

ISP Infrastructure (Relying Party)

RPKI Validator & Cache

A coalesced copy of the RPKI, which is periodically fetched/refreshed directly or indirectly from the Global RPKI using Rsync.

RTR

RTR

iBGP (ext community)

eBGP

eBGP

peer

peer

peer

peer

RFC6486
Manifests for the Resource Public Key Infrastructure (RPKI)

RFC8897
Requirements for Resource Public Key Infrastructure (RPKI) Relying Parties

https://rsync.samba.org/

RFC3779
X.509 Extensions for IP Addresses and AS Identifiers

RFC6487
Resource Certificate Profile

RFC6488
Signed Object Template for the Resource Public Key Infrastructure (RPKI)

RFC6480
An Infrastructure to Support Secure Internet Routing

RFC5280
Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

RFC6810
The Resource Public Key Infrastructure (RPKI) to Router Protocol

RFC7115
Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)

RFC6810
BGP Prefix 1rigin Validation

RPKI Prefix States

RFC6811
BGP Prefix Origin Validation

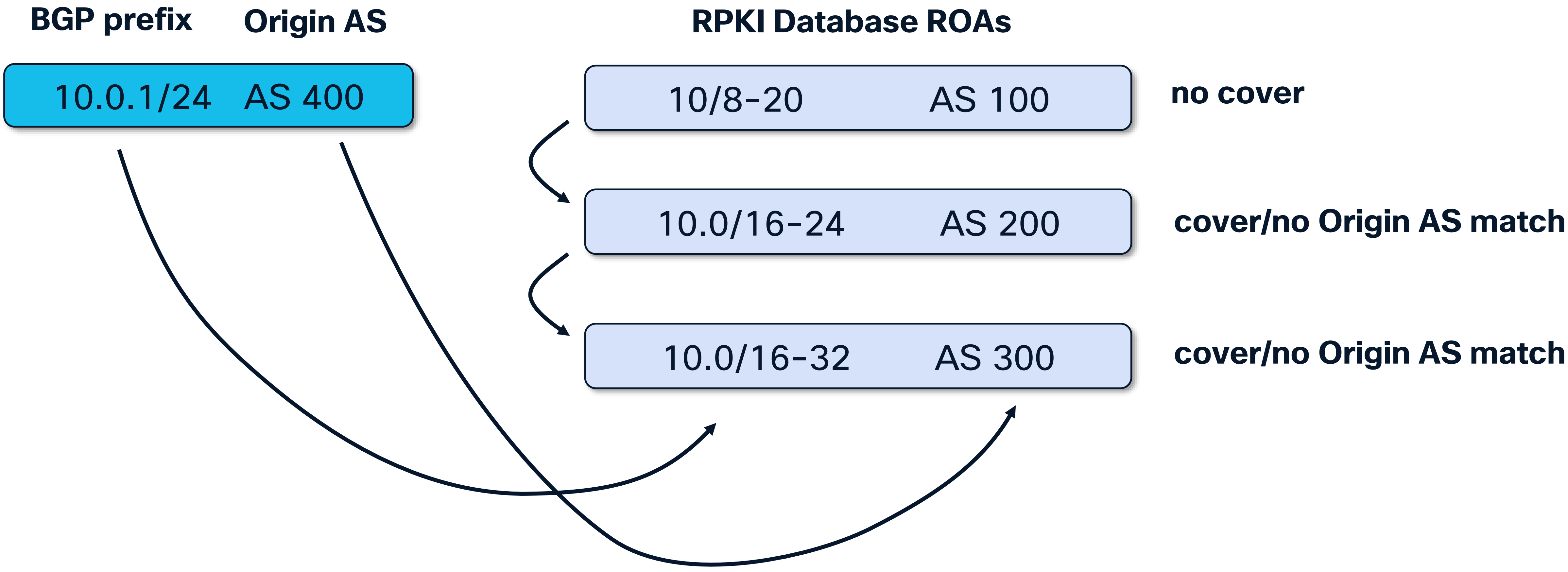
- Origin is:
  - Valid
    - At least one VRP *matches* the Origin AS of the prefix
  - NotFound
    - No VRP *covers* the route prefix
  - Invalid
    - At least one VRP *covers* the prefix, but the Origin AS does not *match* it

No change in prefix behavior

No change in prefix behavior.
As long as BGP RPKI is in adoption mode: prefixes are advertised.

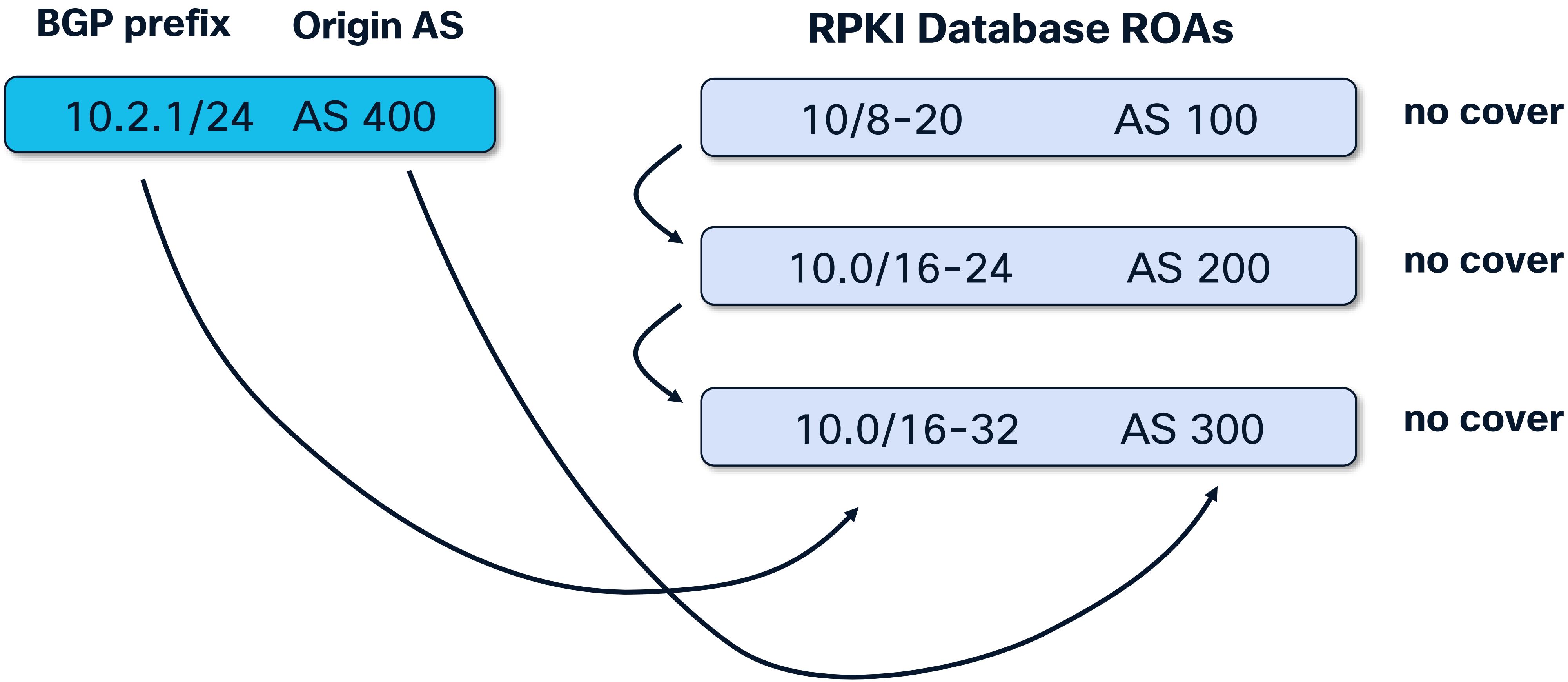
Change in prefix behavior.
Prefix should not be advertised.

RPKI Prefix State Invalid



Result: Invalid

RPKI Prefix State NotFound



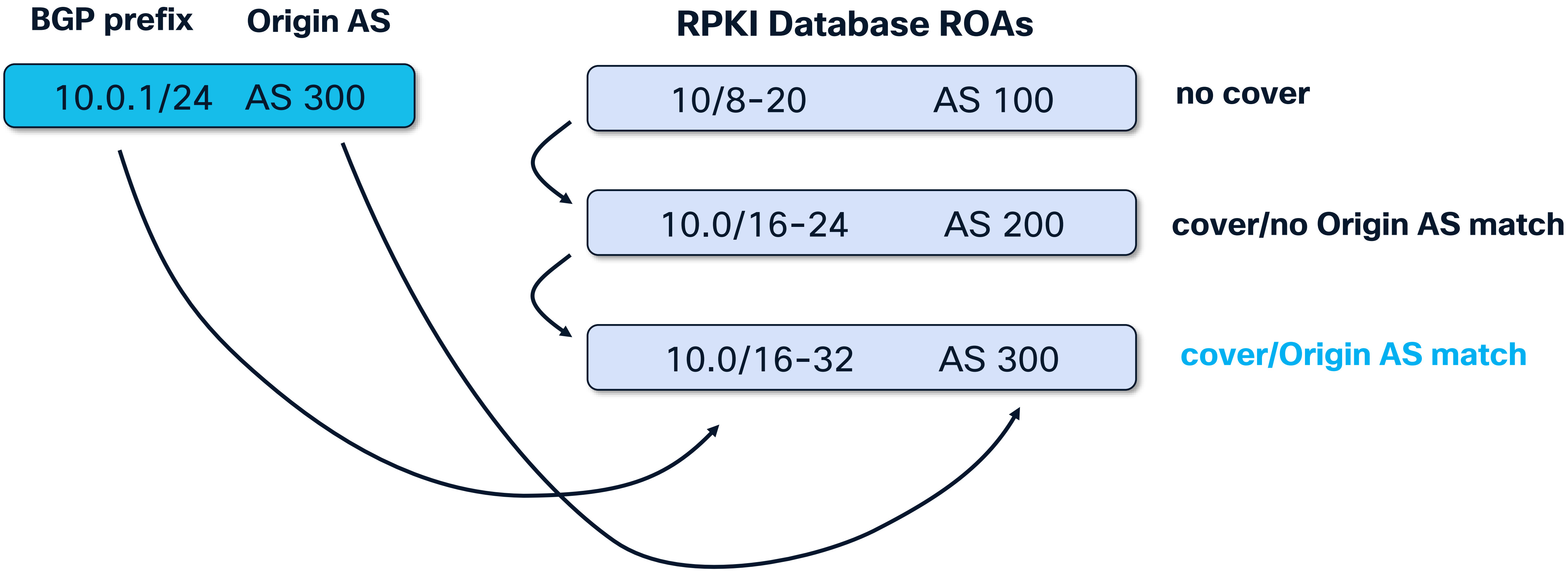
BGP prefix	Origin AS
10.2.1/24	AS 400

RPKI Database ROAs

10/8-20	AS 100	**no cover**
10.0/16-24	AS 200	**no cover**
10.0/16-32	AS 300	**no cover**

Result: NotFound

RPKI Prefix State Valid



BGP prefix | Origin AS

10.0.1/24 | AS 300

RPKI Database ROAs

10/8-20	AS 100	no cover
10.0/16-24	AS 200	cover/no Origin AS match
10.0/16-32	AS 300	cover/Origin AS match

Result: Valid

IOS-XR Configuration – Validator

```
router bgp 65001
 rpki server 10.7.14.14
  transport tcp port 8083
```

```
RP/0/RP0/CPU0:R1#show bgp rpki server summary

Hostname/Address        Transport       State           Time            ROAs (IPv4/IPv6)
10.7.14.14              TCP:8083        ESTAB           00:00:02        39099/6963
```

```
RP/0/RP0/CPU0:R1#show bgp rpki server 10.7.14.14

RPKI Cache-Server 10.48.42.204
  Identifier: 1
  Transport: TCP port 3323
  Bind source: (not configured)
  Connect state: ESTAB
  Conn attempts: 1
  Total byte RX: 15501652
  Total byte TX: 56948
RPKI-RTR protocol information
  Serial number: 2932
  Cache nonce: 0x7543
  Protocol state: DATA_END
  Refresh  time: 600 seconds
  Response time: 30 seconds
  Purge time:     60 seconds
  Protocol exchange
    ROAs announced: 536784 IPv4  129137 IPv6
    ROAs withdrawn:  15897 IPv4    5835 IPv6
    Error Reports :      0 sent       0 rcvd
```

10 min refresh timer is the default and the recommended value.
This is a high value, which prevents frequent route refreshes towards the BGP peers when an ROA update is received.

Enable Origin Validation

```
router bgp 65001
  address-family ipv4 unicast
   bgp origin-as validation enable
```

enables the Origin Validation

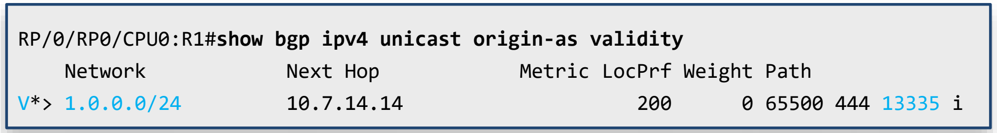
one prefix

```
RP/0/RP0/CPU0:R1#show bgp ipv4 unicast origin-as validity
…
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Origin-AS validation codes: V valid, I invalid, N not-found, D disabled
    Network            Next Hop            Metric LocPrf Weight Path
V*> 1.0.0.0/24         10.7.14.14                             0 65500 444 13335 i
I*> 1.0.0.0/28         10.7.14.14                             0 65500 444 13335 i
N*> 1.0.4.0/25         10.7.14.14                             0 65500 38803 i
N*> 2.0.0.0/12         10.7.14.14                             0 65500 555 1234 i
N*> 2.0.0.0/16         10.7.14.14                             0 65500 555 3215 i
N*> 3.0.0.0/8          10.7.14.14                             0 65500 1 2 3 4 5 6 7
8 9 i
 * i10.0.0.1/32        10.0.0.1                 0    100      0 i
 *>i                   10.0.0.1                 0    150      0 i
```

```
RP/0/RP0/CPU0:RR7#show bgp ipv4 unicast 1.0.0.0/28
…
  65500 444 13335
    10.7.14.14 from 10.7.14.14 (10.7.14.14)
      Origin IGP, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 1, version 53
      Origin-AS validity: invalid
      ASPA validity: not-found
```

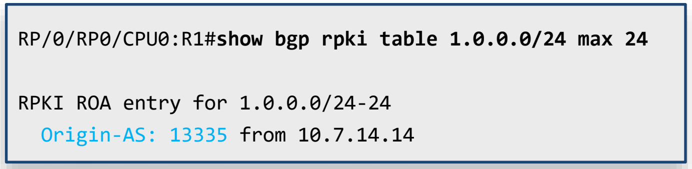
Check Prefix Validation State

Valid



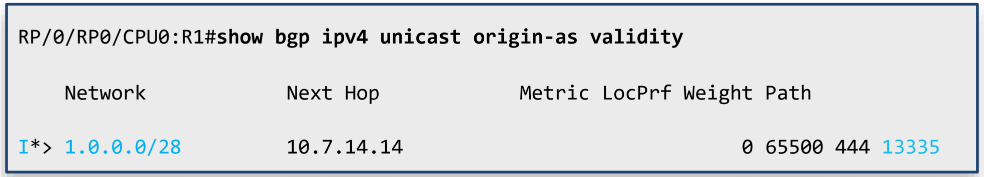
```
RP/0/RP0/CPU0:R1#show bgp ipv4 unicast origin-as validity
    Network            Next Hop            Metric LocPrf Weight Path
V*> 1.0.0.0/24         10.7.14.14                    200      0 65500 444 13335 i
```

exact match in the RPKI table



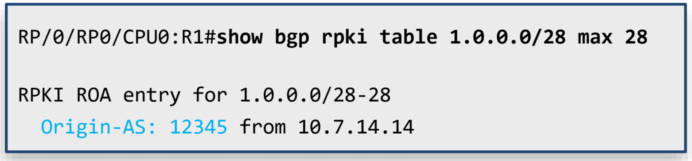
```
RP/0/RP0/CPU0:R1#show bgp rpki table 1.0.0.0/24 max 24

RPKI ROA entry for 1.0.0.0/24-24
  Origin-AS: 13335 from 10.7.14.14
```

Invalid



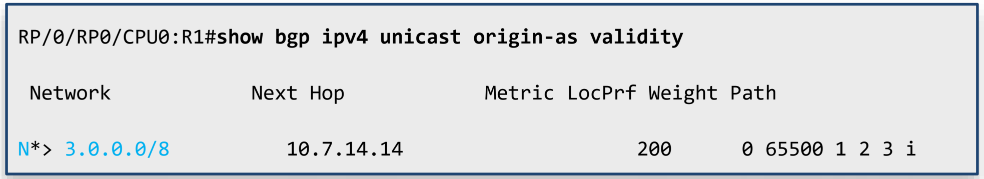
```
RP/0/RP0/CPU0:R1#show bgp ipv4 unicast origin-as validity

    Network            Next Hop            Metric LocPrf Weight Path

I*> 1.0.0.0/28         10.7.14.14                                0 65500 444 13335
```

cover/no match in the RPKI table

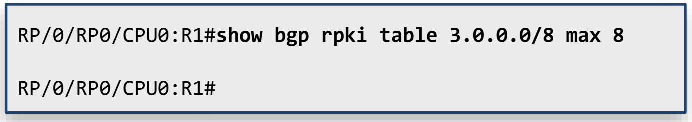


```
RP/0/RP0/CPU0:R1#show bgp rpki table 1.0.0.0/28 max 28

RPKI ROA entry for 1.0.0.0/28-28
  Origin-AS: 12345 from 10.7.14.14
```

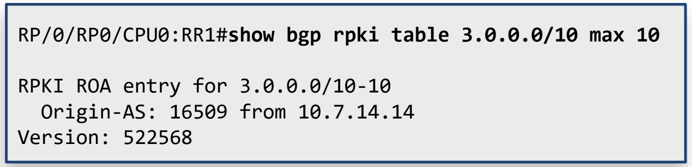
Not-Found



```
RP/0/RP0/CPU0:R1#show bgp ipv4 unicast origin-as validity

 Network            Next Hop            Metric LocPrf Weight Path

N*> 3.0.0.0/8          10.7.14.14                    200      0 65500 1 2 3 i
```



```
RP/0/RP0/CPU0:R1#show bgp rpki table 3.0.0.0/8 max 8

RP/0/RP0/CPU0:R1#
```



```
RP/0/RP0/CPU0:RR1#show bgp rpki table 3.0.0.0/10 max 10

RPKI ROA entry for 3.0.0.0/10-10
  Origin-AS: 16509 from 10.7.14.14
Version: 522568
```

no cover: the only prefixes with 3.0.0.0 in the RPKI table have a longer mask

iBGP & RPKI

- iBGP routes are not validated by the router against the ROA database
- iBGP routes gain an RPKI validity from the RPKI extended community
- If the IBGP route is received without this extended community, then its validation-state is set to not-found

```
router bgp 65001
 bgp origin-as validation signal ibgp
```

have iBGP carry the extended community for RPKI

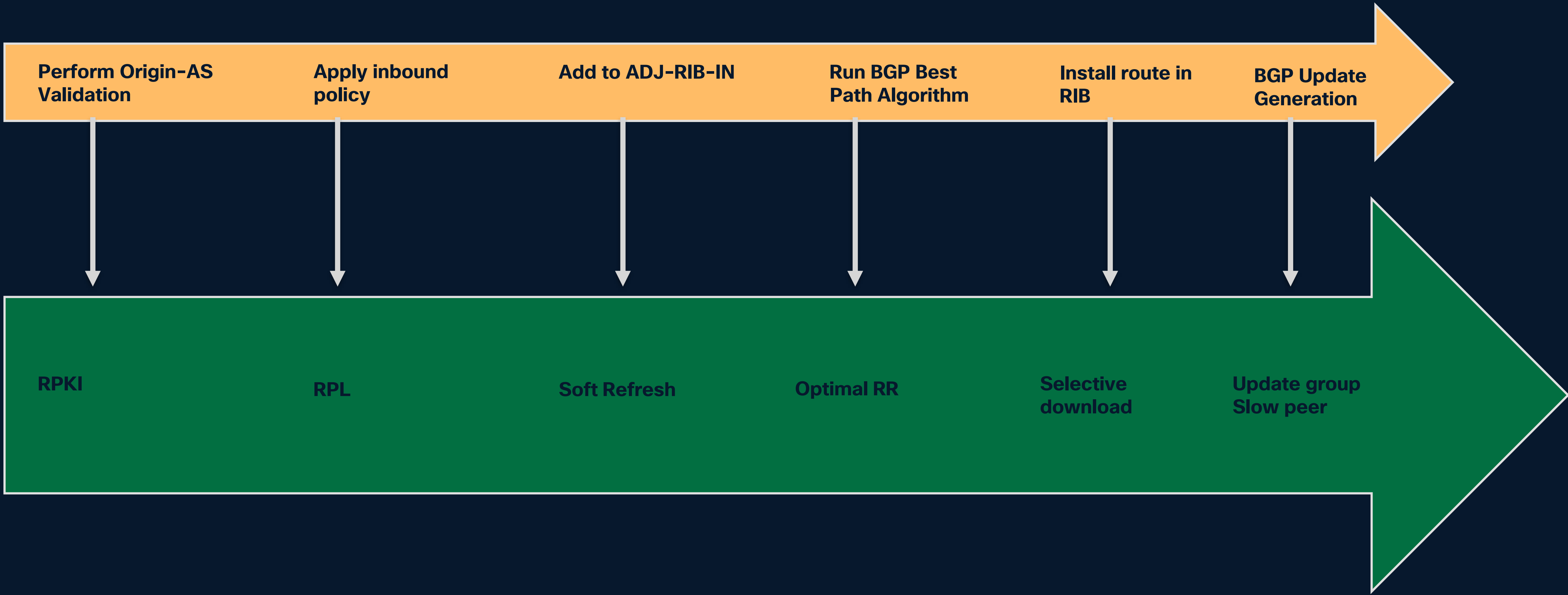
```
RP/0/RP0/CPU0:PE1#show bgp 1.0.0.0/24
…
    10.7.14.14 (metric 10) from 10.0.0.7 (10.0.0.7)
      Origin IGP, localpref 200, valid, internal, best, group-best
      Received Path ID 1, Local Path ID 1, version 598571
      Community: 6500:123
      Extended community: VALIDITY:0
      Origin-AS validity: valid (iBGP signalled)
      ASPA validity: disabled
```

0 - Valid
1 - Not Found
2 - Invalid

Non-Transitive Opaque Extended Community 0x4300
0x0000 : 4 bytes indicating state

Extended Community for RPKI validation state signaling

Summary



Perform Origin-AS Validation

Apply inbound policy

Add to ADJ-RIB-IN

Run BGP Best Path Algorithm

Install route in RIB

BGP Update Generation

RPKI

RPL

Soft Refresh

Optimal RR

Selective download

Update group
Slow peer

Complete your session evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn exclusive prizes!

Complete your surveys in the Cisco Live mobile app.

Continue your education

Visit the Cisco Showcase for related demos

Book your one-on-one Meet the Engineer meeting

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact us at: Mankamana Mishra: mankamis@cisco.com
Serge Krier: sekrier@cisco.com

Thank you

CISCO Live!