Cisco live!

February 15 – 19, 2016 • Berlin, Germany

We're ready. Are you?

Scaling BGP

Luc De Ghein – Technical Leader Services

BRKRST-3321

Agenda

- Introduction
- Goal
- Scale Challenges
- Memory Utilization
- Full mesh iBGP
- Update Groups
- Slow Peer
- RR Problems & Solutions
- Deployment
- Multi-Session
- MPLS VPN
- OS Enhancements
- Conclusion

"We're Gonna Need a Bigger Boat"

Jaws

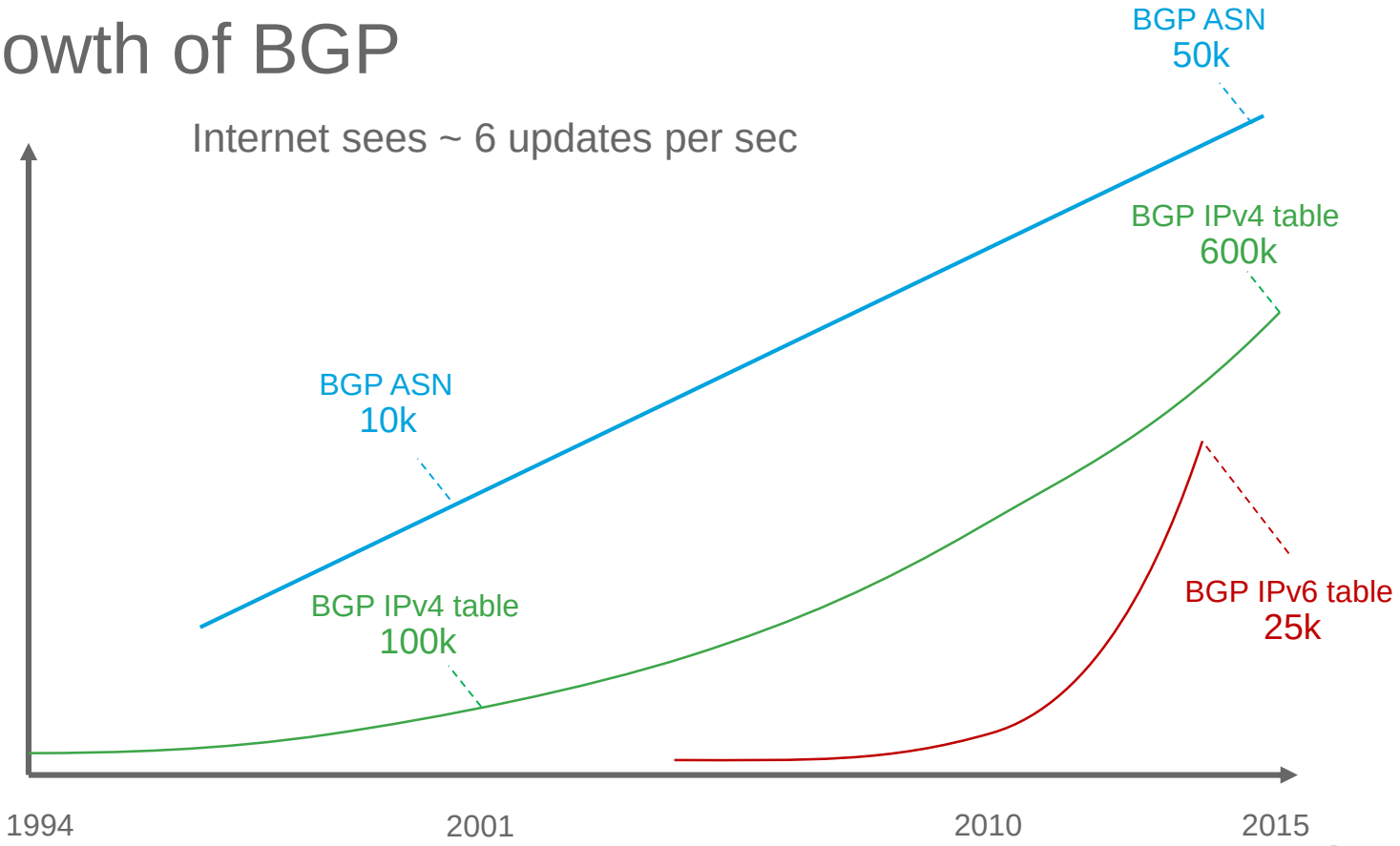
Goal of This session

- Present causes of scale challenges
- Present solutions for scaling
- There are no scaling numbers
- What you can control the most:
  - Buy a bigger box
  - Design the network properly

Scale Challenges

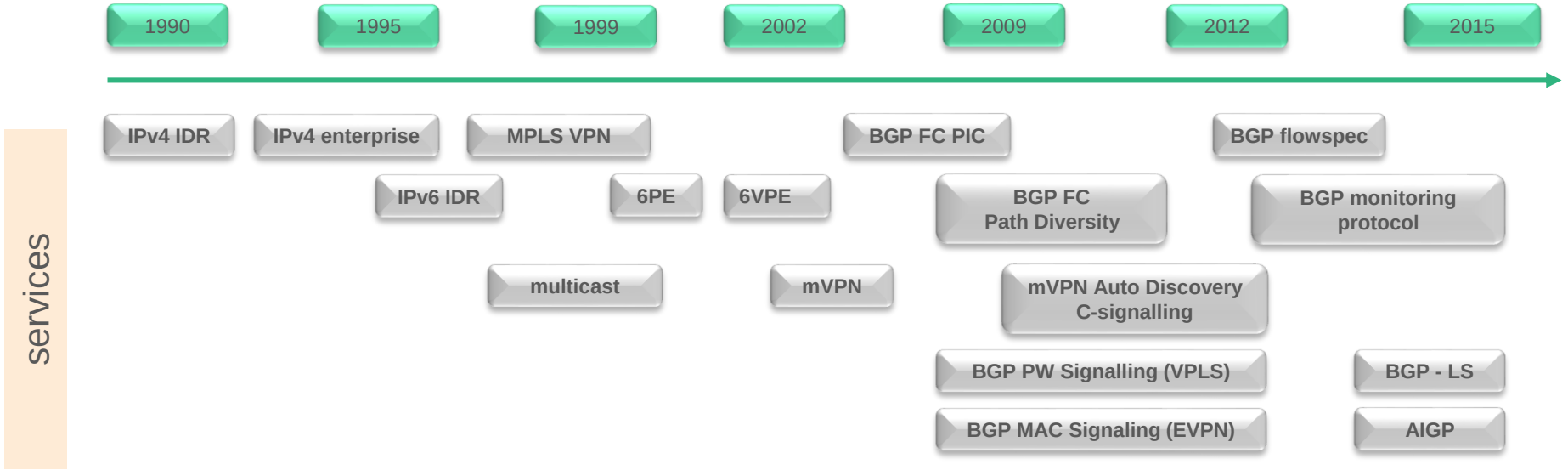
- BGP is robust, simple and well-known
- We need to overcome the following:
  - Newer services: new AFs
  - More prefixes
  - Larger scale: more (BGP) routers
  - More multipath
  - More resilience
    - PIC Edge, Best External, leading to more prefixes/paths

Growth of BGP



Source: bgp.potaroo.net

More Services Using BGP



Service Address Families

For Your Reference

IPv4 unicast

IPv6 multicast

vpn multicast in overlay

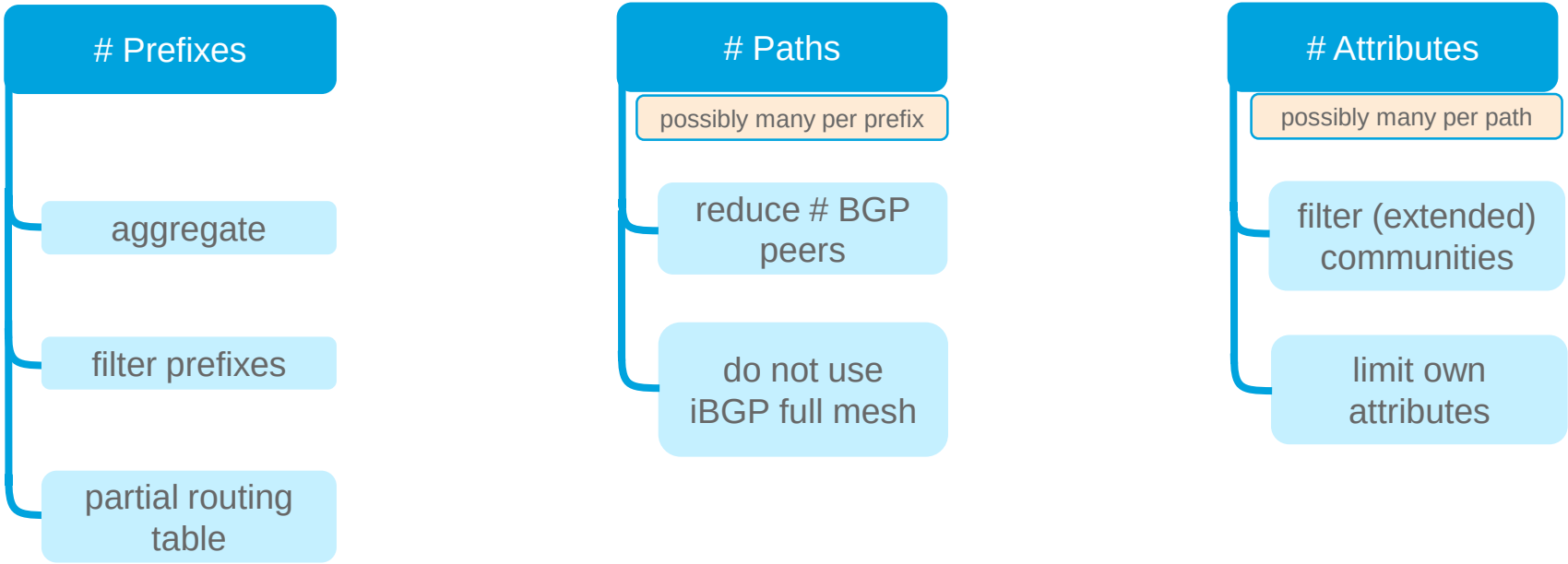
Layer 2 linkstate

IPv4 unicast	IPv6 unicast	vpnv4 unicast	nsap unicast	IPv4 Flowspec
IPv4 multicast	IPv6 multicast	vpnv4 multicast	l2vpn vpls	IPv6 Flowspec
IPv4 MVPN	IPv6 MVPN	vpnv6 unicast	l2vpn evpn	vpnv4 Flowspec
IPv4 MDT		vpnv6 multicast	l2vpn mspw	vpnv6 Flowspec
IPv4 tunnel		rtfilter unicast	linkstate	

Cisco live!

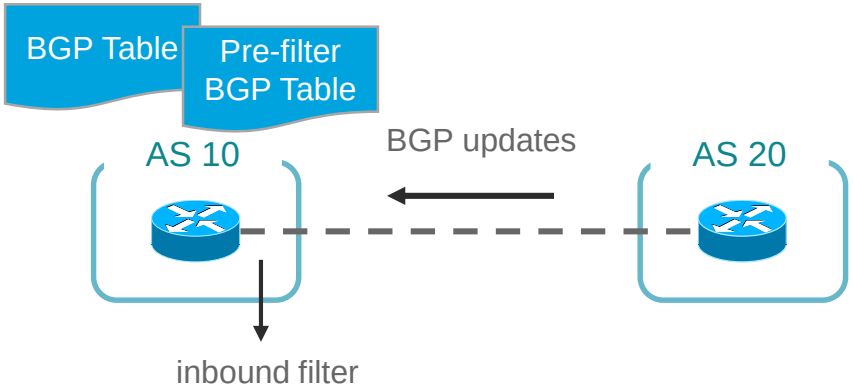
Memory Utilization

High Memory Utilization - Solutions



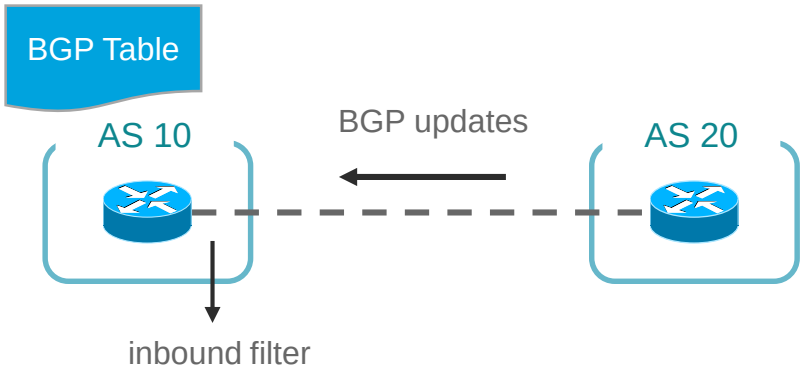
High Memory Utilization

soft reconfiguration inbound



- Filtered prefixes are stored: much more memory used
- Support only on router itself
- Changed filter: re-apply policy to table with filtered prefixes

route refresh



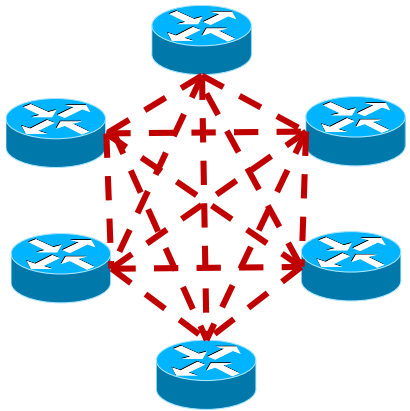
- Filtered prefixes are dropped
- Support needed on peer, but this a very old feature
- Changed filter: router sends out route refresh request to peer to get the full table from peer again

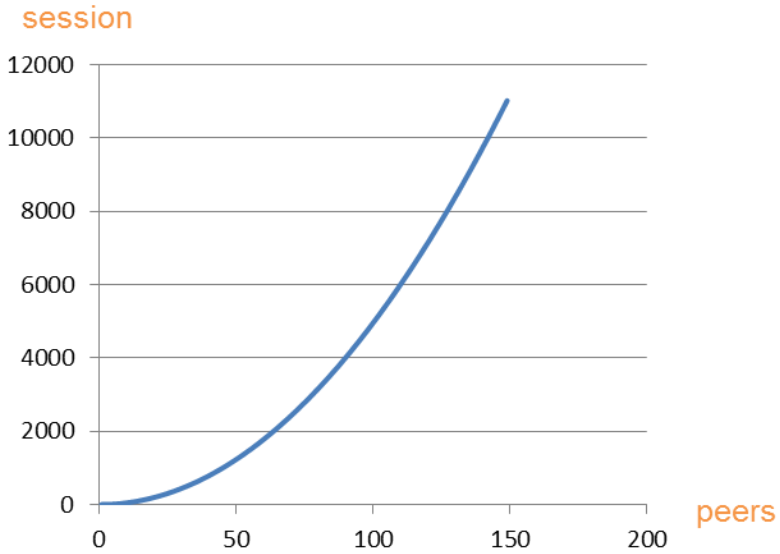
Recommendation: Use soft reconfiguration inbound only on eBGP peering when you need to know what the peer previously advertised and you filtered out

Full Mesh iBGP

Is Full Mesh iBGP Scalable?

- Per BGP standard: iBGP needs to be full mesh
- Total iBGP sessions = n * (n-1) / 2
- Sessions per BGP speaker = n - 1



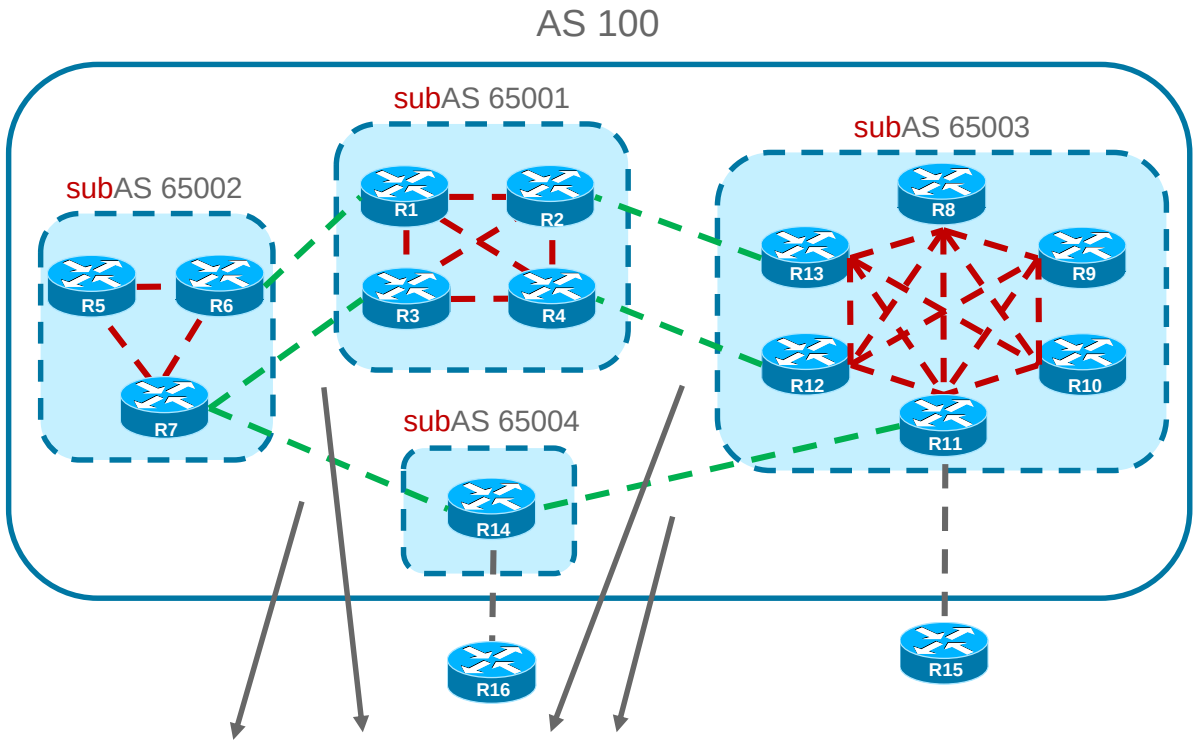


- Two solutions
  1. Confederations
  2. Route reflectors

Confederations

- Create # of sub-AS inside the larger confederation
- Conferation AS looks like normal AS to the outside

- Full mesh iBGP still needed inside subAS
- No full mesh needed between subAS (it's eBGP)
- Every BGP peer needs to be in a subAS
- Each subAS can have different IGP with next-hop-self within confed
- No connectivity needed between any subAS's

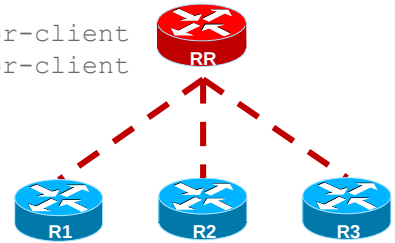


- Flexible confed eBGP peerings
- Redundancy needed vs increased memory/CPU
- But full mesh between subAS's is not needed

Route Reflectors

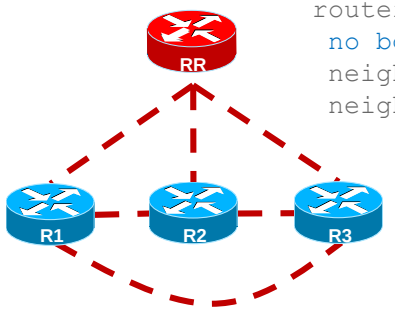
- A route reflector is an iBGP speaker that reflects routes learned from iBGP peers to other iBGP peers, called RR clients
- iBGP full mesh is turned into hub-and-spoke
- RR is the hub in a hub-and-spoke design


```
router bgp 65002
 neighbor R1 route-reflector-client
 neighbor R2 route-reflector-client
```



RR clients are regular iBGP peers

OR



RR clients are interconnected

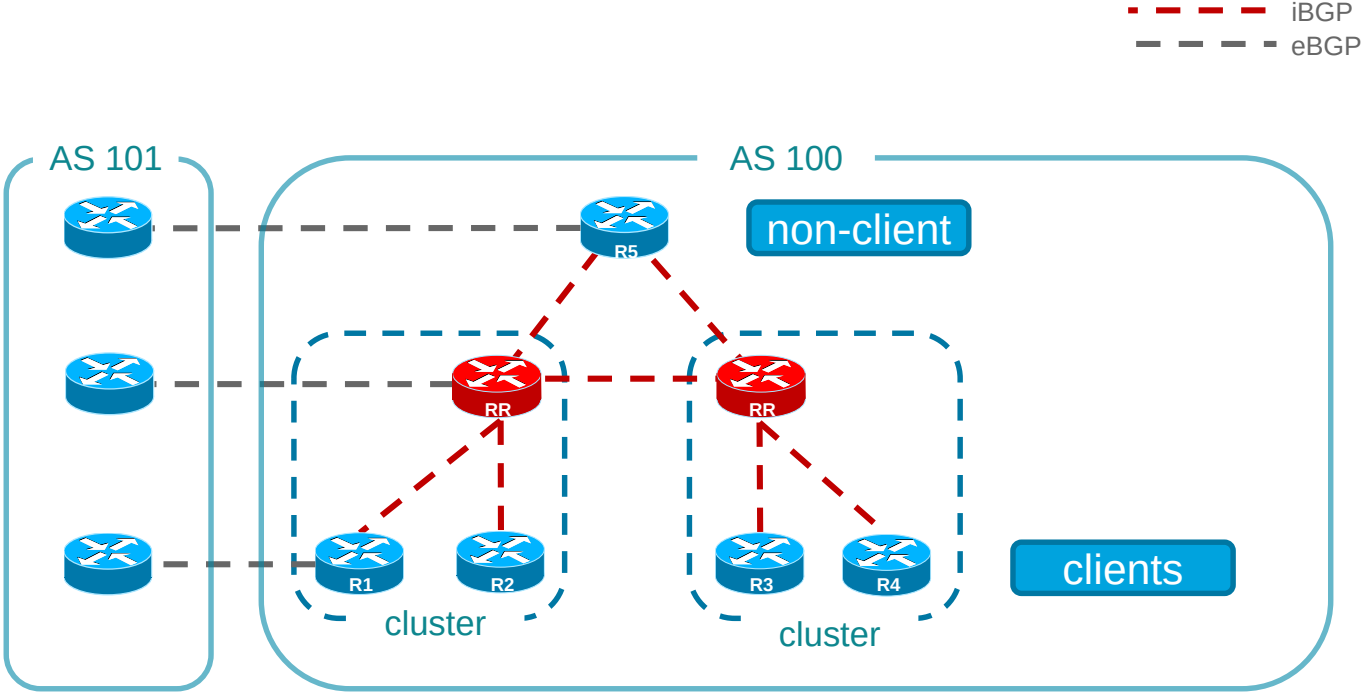
```
router bgp 65002
 no bgp client-to-client reflection
 neighbor R1 route-reflector-client
 neighbor R2 route-reflector-client
```

client-to-client reflection can be turned off

Route Reflector

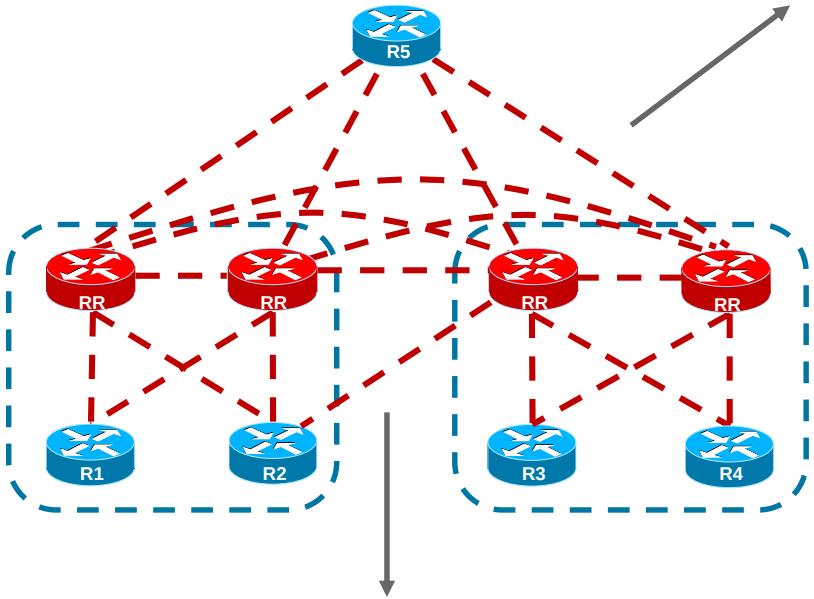
What's Possible?

Any router can peer eBGP



Route Reflector - Cluster

- Redundancy needed, min of 2 RRs per cluster
- Cluster = RR and its clients



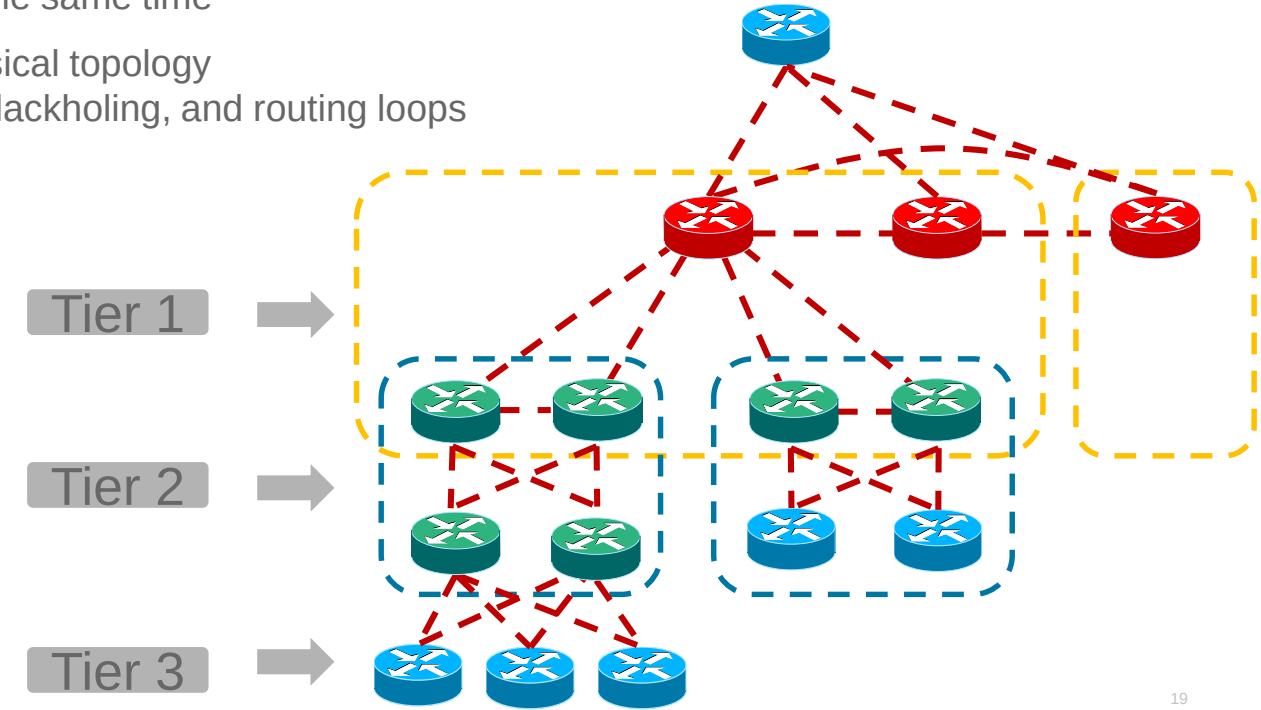
Full mesh between RRs and non-clients should be kept small

Client can peer with RRs in multiple clusters

Hierarchical Route Reflectors

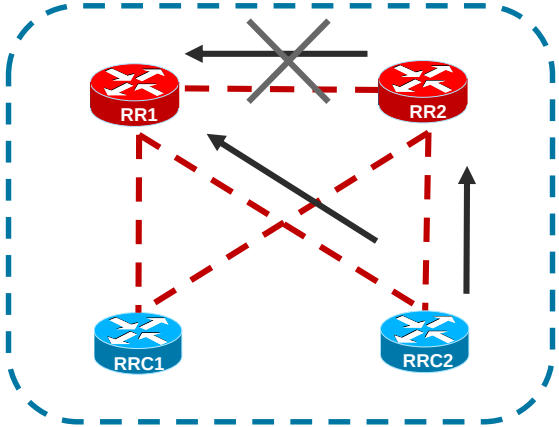
- Chain RRs to keep the full mesh between RRs and non-clients small
- Make RRs clients of other RRs
- An RR is a RR and RR client at the same time
- iBGP topology should follow physical topology
  - Prevents suboptimal routing, blackholing, and routing loops

- RRs in top tier need to be fully meshed
- There is no limit to the amount of tiers



Route Reflector – Same Cluster-ID or Not?

- RR1 has only 1 path for routes from RRC2
  - RR1 and RR2 have the same cluster-ID
- If one link RR to RR-client fails
  - iBGP session remains up, it is between loopback IP addresses
- If different cluster-ID:
  - RR1 stores the path from RR2
  - RR1 uses additional CPU and memory
  - Potentially for many routes
- Using the same or different cluster-ID for RRs in one set?
  - Different cluster-ID
    - Additional memory and processor overhead on RR
  - Same cluster-ID
    - Less redundant paths



Picking RRs

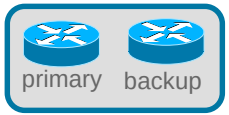
How many?

Redundancy

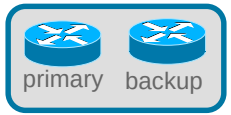
Sets of two

Services

- To scale: sets (per group of) address families



service 1
(one or more AFs)



service 2
(one or more AFs)

Where?

Location

- Geo
- Datacenter
- Region

Which kind?

Dedicated RR

No forwarding (no FIB)
RIB and BGP/IGP

Needed Resources

Memory
CPU

Virtual router

- Mobility
- Manageability
- Same BGP implementation and software version as deployed on the Edge (XE/XR)
- Reduced physical footprint (power/cooling/cabling)
- Performance (multi-core) / memory (64-bit)

7200

ASR1K

CSR1000V

ASR9Kv
(vRR)

Cisco live!

BGP RR Scale - Selective RIB Download

- To block some or all of the BGP prefixes into the RIB (and FIB)
- Only for RR which is not in the forwarding path
- Saves on memory and CPU
- Implemented as filter extension to table-map command

- For AFs IPv4/6
  - not needed for AFs vpnv4/6
- Benefit
  - ASR testing indicated 300% of RR-client session scaling (in order of 1000s)

configuration

```
router bgp 1

 address-family ipv4
  table-map block-into-fib filter

route-map block-into-fib deny 10
```

no BGP prefixes in RIB

```
RR1#show ip route bgp


RR1#
```

no BGP prefixes in FIB

```
RR1#show ip cef


RR1#
```

configuration IOS-XR

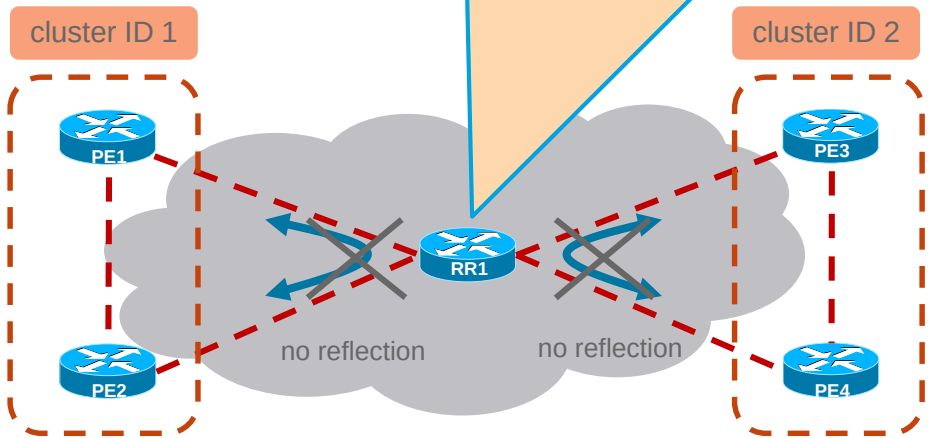
```
route-policy block-into-fib
 if destination in (...) then
   drop
 else
   pass
 end-if
```

```
router bgp 1

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

Multi-Cluster ID

```
router bgp 1
 no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1
 no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.2
```

- An RR can belong to multiple clusters
  - On IBGP neighbor of RR: cluster IDs on a per-neighbor basis
  - The global cluster ID is still there
  - Intra-cluster client-to-client reflection can be disabled, when clients are meshed
  - Can be disabled for all clusters or per cluster
    - More work - sending more updates - for RR clients
    - Less work - sending fewer updates - for RRs



- Each set of peers in cluster ID has its own update group
- Loop-prevention mechanism is modified
  - Taking into account multiple cluster IDs

Comparing Confederation vs. RR

For Your Reference

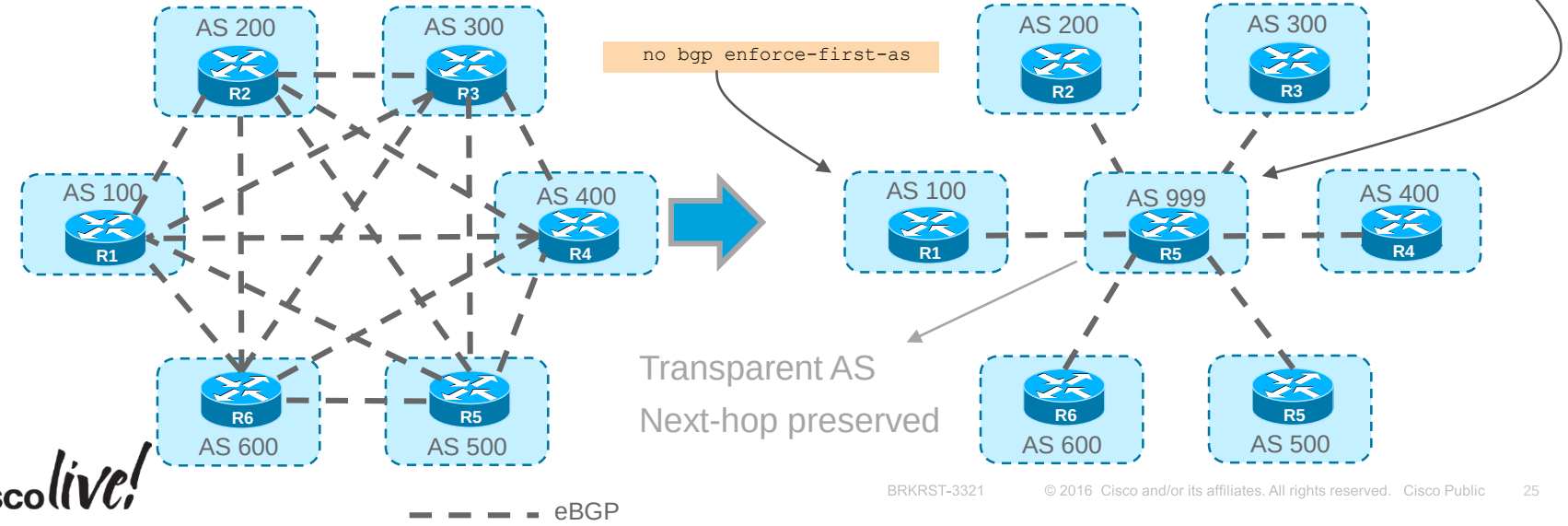
	Confederation	Route Reflector
Loop prevention	AS Confederation Set	Originator/Cluster ID
Break up a single AS	Sub-AS' (possibly with separate IGPs)	Clusters
Redundancy	Multiple connections between sub-AS'	Client connects to several reflectors
External Connections	Anywhere in the network	Anywhere in the network
Multilevel Hierarchy	RRs within sub-AS'	Clusters within clusters
Policy Control	Along outside borders and between sub-AS' Dampening possible on eBGP confed	Along outside border
Scalability	**Medium; still requires full iBGP within each sub-AS**	**Very high**
Migration	Very difficult (impossible in some situations) But easy when merging two companies	Moderately easy
Deployment of (new) features	Decentralized	Central on RR

Note: Route reflectors can exist inside subAS

BGP Route Server

- Alternative to eBGP full mesh
- Used by IX (Internet eXchange) providers
- Operational simplicity
- Reduces CPU/memory/configuration
- Context policy can be used

```
router bgp 999
 route-server-context rs-context
  !
  address-family ipv4 unicast
   import-map rs-import-map
 !
 neighbor 10.1.1.1 remote-as 100
 !
 address-family ipv4
  neighbor 10.1.1.1 route-server-client context rs-context
!
ip as-path access-list 100 permit ^200$
!
route-map rs-import-map permit 10
 match as-path 100
```



Update Groups

Grouping of BGP Neighbors: Optimization

Configuration/administration

- peer groups
- templates, session-groups, af-groups, neighbor-group

- CLI only

- BGP neighbors with same outbound policy will be put in the same update group regardless if
  - peer-groups are defined
  - templates are defined
  - neighbors are individually defined

Performance/scalability

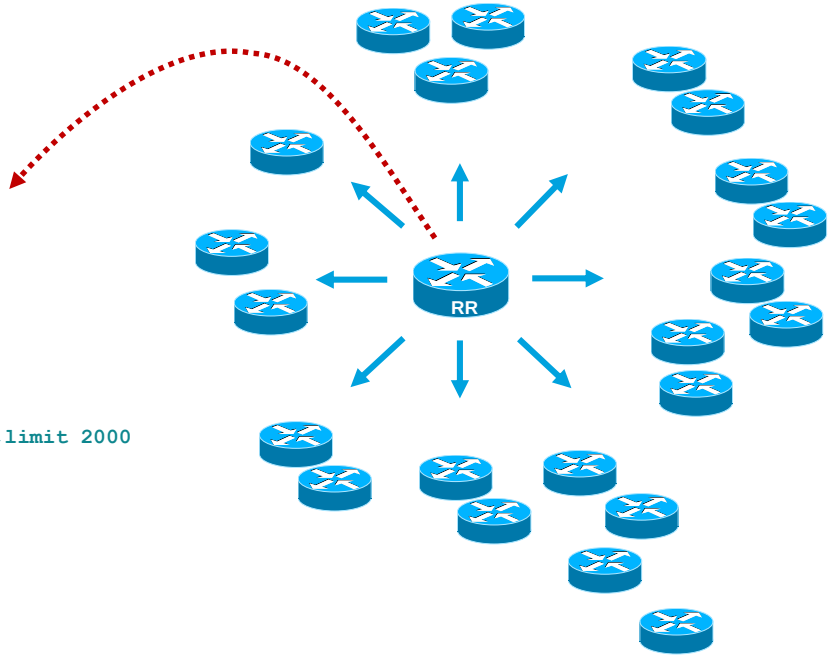
- update groups

- Dynamic grouping BGP of peers according to common outbound policy
- Networks that have the same best-path attributes can be grouped into the same message improving packing efficiency
- BGP formats the update messages once and then replicates to all members of the update group
- replication instead of formatting updates per neighbor: efficiency
- dynamic = policy changes, update group membership changes
- AF independent : a peer can belong to different update groups in different address families

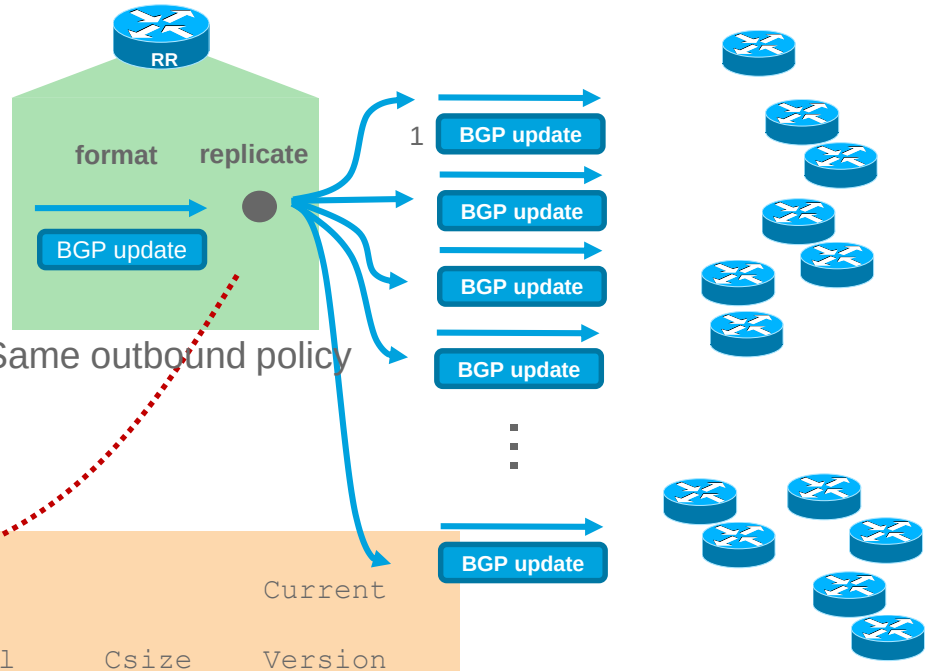
Update Group on RR

- Update groups are very usefull on all BGP speakers
  - but mostly on RR due to
    - # of peers
    - equal outbound policy
- iBGP typically has no outbound policy
  - RRs have large number of iBGP peers in one update group

```
RR#show bgp ipv4 unicast update-group 2
BGP version 4 update-group 2, internal, Address Family: IPv4 Unicast
  BGP Update version : 3201/0, messages 0
  Route-Reflector Client
  Topology: global, highest version: 3201, tail marker: 3201
  Format state: Current working (OK, last not in list)
                Refresh blocked (not in list, last not in list)
  Update messages formatted 2013, replicated 24210, current 0, refresh 0,limit 2000

  Number of NLRIs in the update sent: max 812, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 101 members:
   10.100.1.2       10.200.1.1       10.200.1.10      10.200.1.100
   10.200.1.11      10.200.1.12      10.200.1.13      10.200.1.14
   ...
```



Update Group Replication




```
RR#show ip bgp replication 2

                                                                  Current
Next
Index   Members           Leader          MsgFmt     MsgRepl      Csize     Version
Version
    2       101       10.100.1.2            2013       24210    0/2000        3201/0
```

update group 2	total # of members	formatting according to leader's policy	# of formatted messages	# of replications	size of cache

Adaptive Message Cache Size

- Cache = place to store formatted BGP message, before they are send
- Update message cache size throttles update groups during update generation and controls transient memory usage
- Is now adaptive
  - Variable (change over time) queue depth from 100 to 5000
    - Number of peers in an update groups
    - Installed system memory
    - Type of address family
    - Type of peers in an update group
  - Benefits
    - Update groups with large number of peers get larger update cache
    - Allows routers with bigger system memory to have *appropriately* bigger cache size and thereby queue more update messages
    - vpnv4 iBGP update groups have larger cache size
    - Old cache sizing scheme could not take advantage of expanded memory available on new platforms
    - Results in faster convergence

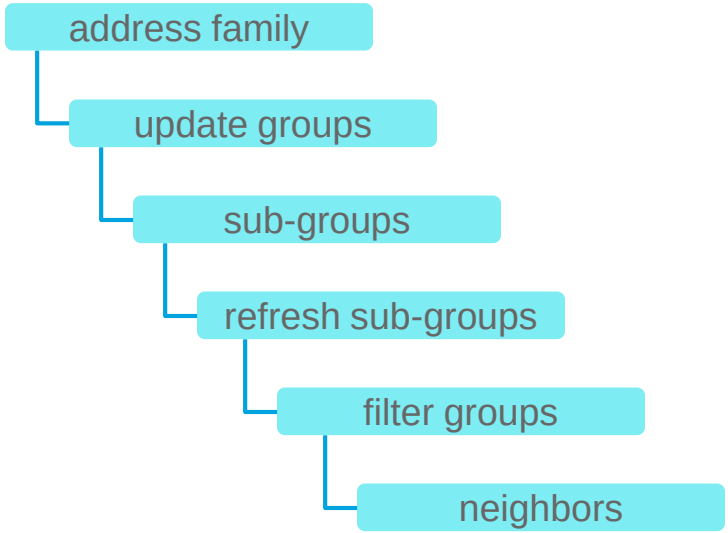
Parallel Processing of Route-refresh (and New Peers)

Original update group handles new transient updates while refresh update group handles re-announcements

- IOS
  - Serving all peers for which route-refresh is needed, at once
  - Tracking the refreshing peers
  - Maintains a copy of neighbor instance that needs to re-announce the table
  - Allows original update group and its members to advertise transient updates without getting blocked
  - Tracked with flags
    - SPE flags to indicate refresh
    - E flag : refresh end marker. Peer is scheduled or participating in a refresh.
    - S flag : refresh has started. If not present: waiting for its refresh to start either because another refresh for the same group is in progress or because the net prepend is not yet complete
    - P flag : refresh is paused waiting for other refresh members that started later to catch up
- IOS-XR: Refresh sub-goup

Update Groups in IOS XR

IOS-XR




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

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
```

Cisco live!

Slow Peer

Slow Peer

IOS

update group 1

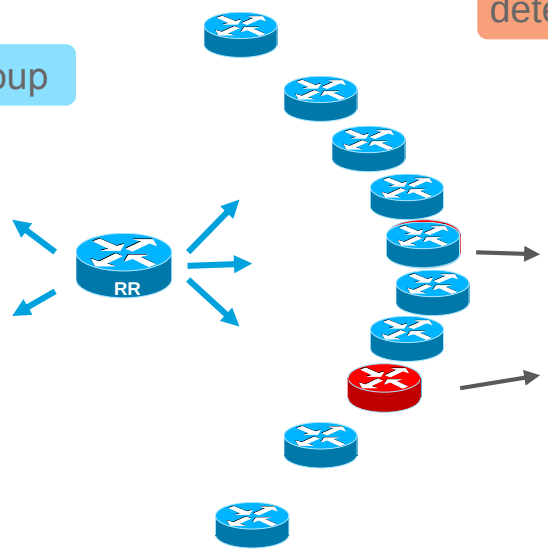
detection phase track peer queue

protection phase

“slow” update group

recovery phase

slow update group is no longer slow



- slow peer = cannot keep up with the rate at which we are generating update messages over a prolonged period of time (order of minutes)
- filled up cache: blocking all peers

Possible causes

- High CPU
- Transport issues (packet loss/loaded links/TCP)

convergence speed of update goup

OK

```
%BGP-5-SLOWPEER_DETECT: Neighbor IPv4 Unicast 10.100.1.1 has been detected as a slow peer
```

```
%BGP-5-SLOWPEER_RECOVER: Slow peer IPv4 Unicast 10.100.1.1 has recovered
```

Allows for fast and slow peers to proceed at the their own speed

Slow Peer CLI

configuration

detection

- per AF
- per VRF
- per peer
- per peer policy template

protection

static

- per AF
- per peer(-group)
- per peer policy template

dynamic

- per VRF
- per peer
- per peer policy template

optional: permanent = peer is not moved back automatically to the update group

show commands

show bgp ... *slow* command

clear commands

clear bgp ... slow command

This is a forced clear of the slow-peer status; the peer is moved to the original update group

Old Slow Peer Solution

Solution before this feature: manual movement

- Create a different outbound policy for the slow peer
- Policy must be different than any other
  - You do not want the slow peer to move to another already existing update group
- Use something that does not affect the actual policy
  - For example: change minimum advertisement interval (MRAI) of the peer (under AF)
  - Also avoiding the cause for a full update (equivalent of a route-refresh)

```
router bgp 1
 address-family vpnv4
  neighbor 10.100.1.1 advertisement-interval 1
```

Slow Peer Mechanism Details

For Your Reference

- Identifying Slow Peer

```
RR#show bgp ipv4 unicast update-group 1 summary
Summary for Update-group 1, Address Family IPv4 Unicast
BGP router identifier 10.100.1.5, local AS number 1
BGP table version is 500001, main routing table version 500001
100000 network entries using 14400000 bytes of memory
BGP using 24373520 total bytes of memory
BGP activity 115574/15574 prefixes, 300000/200000 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd

10.100.1.1      4            1    1257   67368   402061    0 2000 18:56:16        0
10.100.1.2      4            1    1219   23362   402061    0    0 18:23:46        0
10.100.1.3      4            1    1257   23398   402061    0    0 18:56:42        0
10.100.1.4      4            1   10002    1891   402061    0    0 00:01:37   100000
```

convergence is achieved if all peers are at the table version

output queue is not empty, persistently?

Slow Peer Mechanism Details

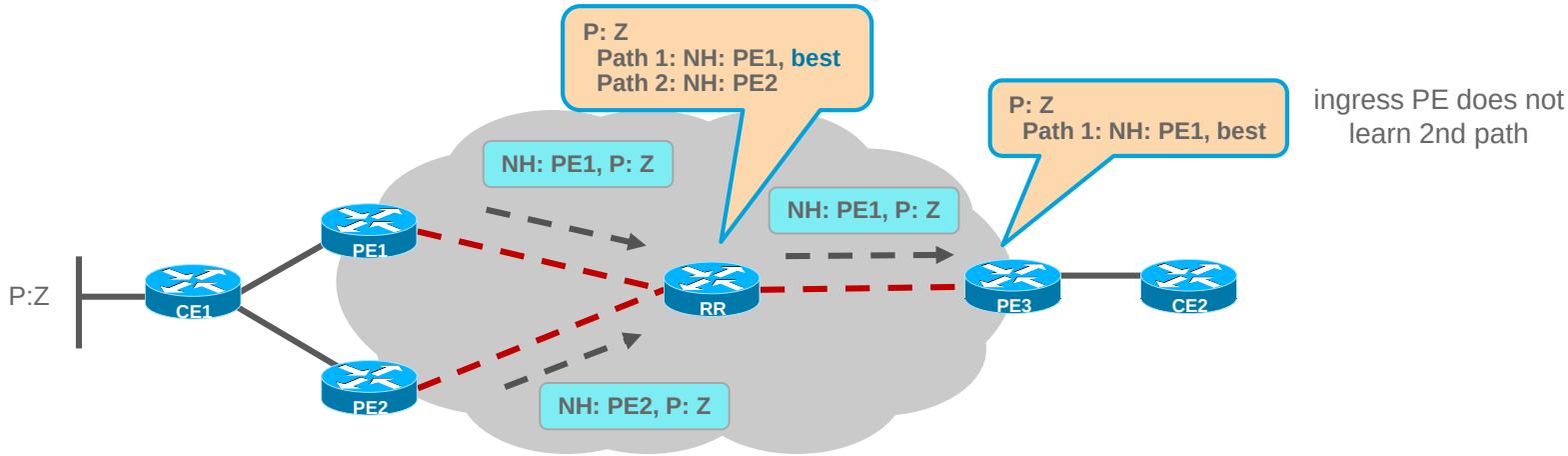
For Your Reference

- Identifying Slow Peer

```
RR#show bgp ipv4 unicast replication 1

                                                                  Current    Next
Index  Members          Leader       MsgFmt    MsgRepl     Csize  Version Version
    1        4      10.100.1.1        78114     144314 2000/2000  402061/500001

RR#show bgp ipv4 unicast update-group 1
BGP version 4 update-group 1, internal, Address Family: IPv4 Unicast
  BGP Update version : 402061/500001, messages 2000
  Topology: global, highest version: 500001, tail marker: 402061 (pending)
  Format state: Current blocked (no message space, last no message space)
                Refresh blocked (not in list, last not in list)
  Update messages formatted 78115, replicated 144318, current 2000, refresh 0, limit 2000
Minimum time between advertisement runs is 0 seconds
  Has 4 members:
   10.100.1.1       10.100.1.2       10.100.1.3       10.100.1.4
```

RR Problems & Solutions

Best Path Selection Route Advertisement on RR



- The BGP4 protocol specifies the selection and propagation of a single best path for each prefix
- If RRs are used, only the best path will be propagated from RRs to ingress BGP speakers
  - Multipath on the RR does not solve the issue of RR only sending best path
- This behavior results in number of disadvantages for new applications and services

Why Having Multiple Paths?

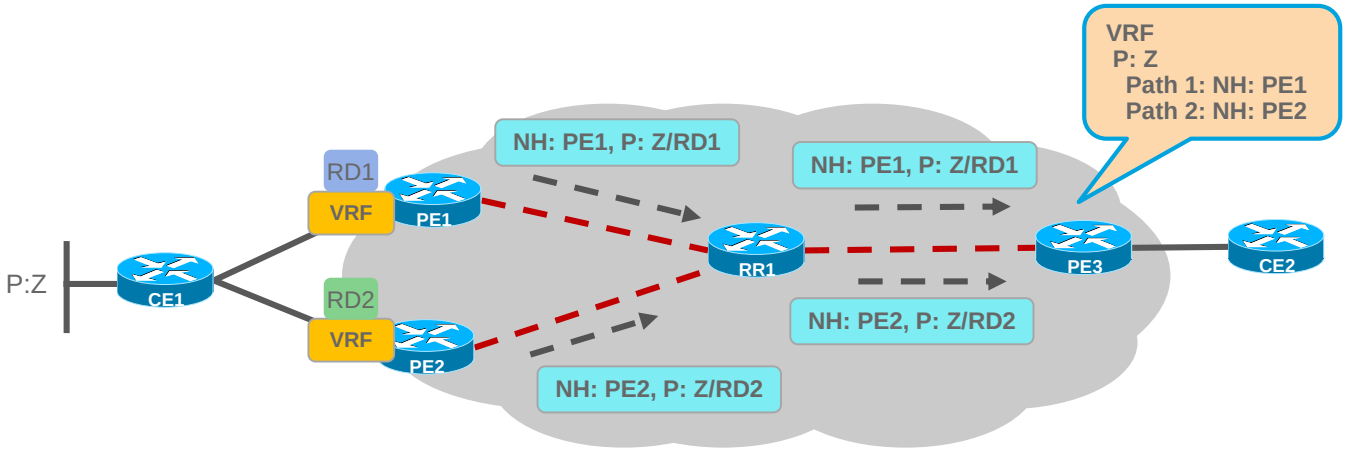
- Convergence
  - BGP Fast Convergence (multiple paths in local BGP table)
  - BGP PIC Edge (backup paths ready in forwarding plane)
- Multipath load balancing
  - ECMP
- Allow hot potato routing
  - = use optimal route
  - The optimal route is not always known on the border routers
- Prevent oscillation
  - The additional info on backup paths leads to local recovery as opposed to relying on iBGP
  - Stop persistent route oscillations caused by comparison of paths based on MED in topologies where route reflectors or the confederation structure hide some paths

Diverse BGP Path Distribution

Overview

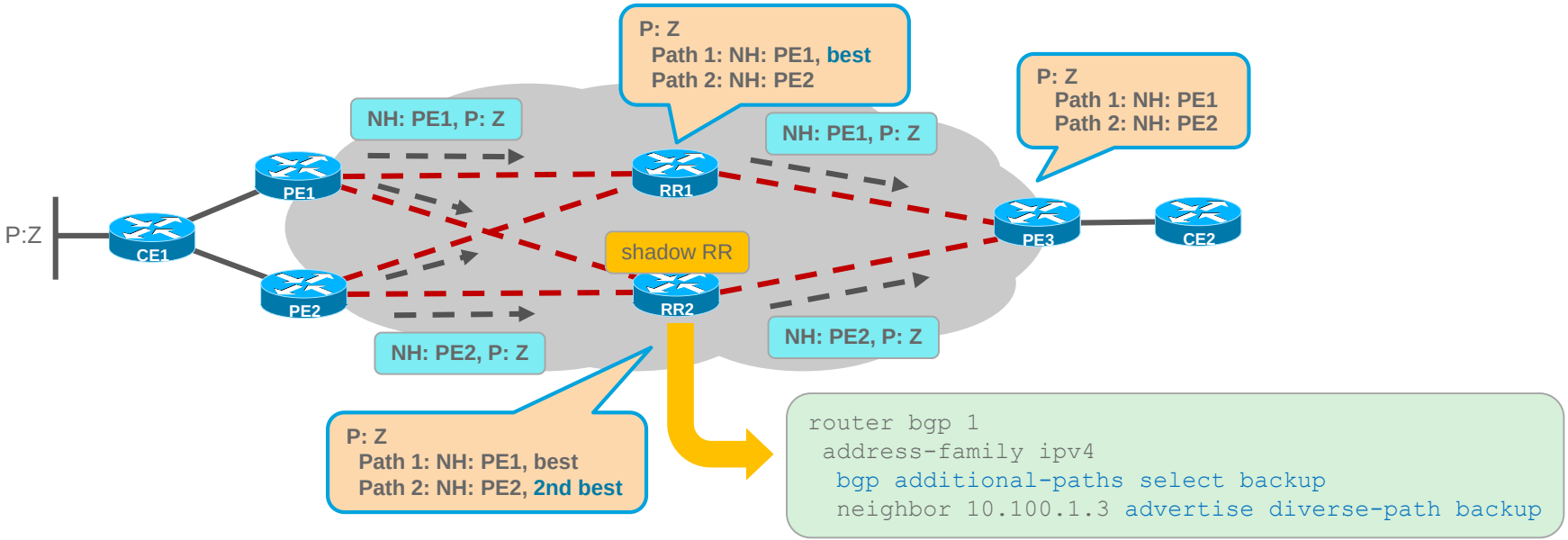
- VPN unique RD (Route Distinguisher)
- BGP Best External
- BGP shadow RR / session
- BGP Add-Path

Unique RD for MPLS VPN



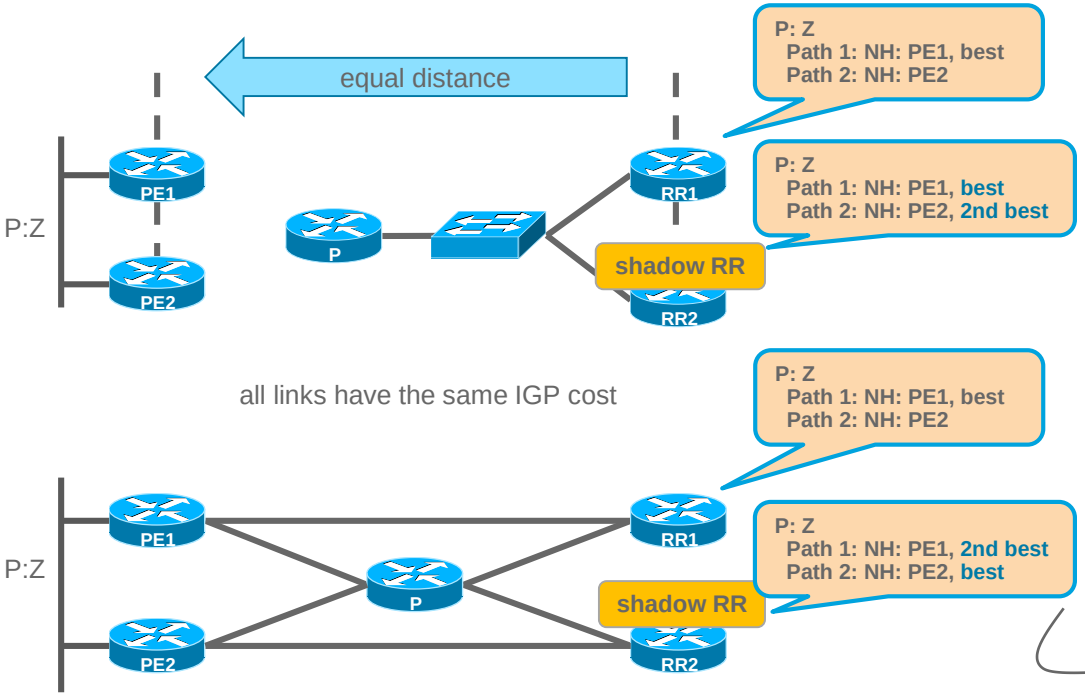
- Unique RD per VRF per PE
- One IPv4 prefix in one VRF becomes unique vpnv4 prefix per VPN per PE
- RR advertises all paths
- Available since the beginning of MPLS VPN, but only for MPLS VPN

Shadow Route Reflector (aka RR Topologies)



- Easy deployment
- One additional “shadow” RR per cluster
- RR2 does announce the 2nd best path, which is different from the primary best path on RR1 by next hop

Shadow Route Reflector



Note: primary RRs do not need diverse path code

RR and shadow RR are co-located.
They're on same vlan with same IGP metric towards prefix.

Note: primary and shadow RRs do not need to turn off IGP metric check

RR and shadow RR are not co-located.

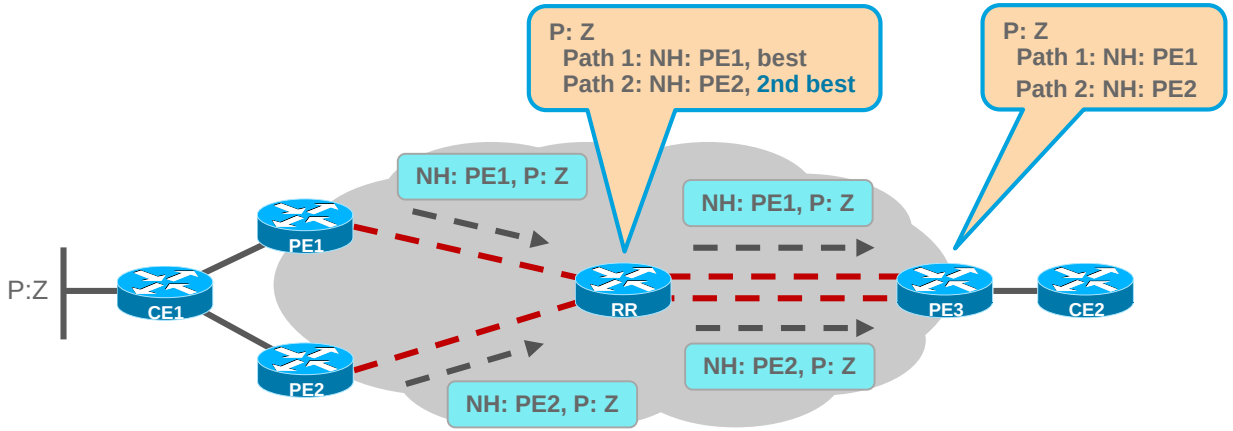
Note: primary and shadow RRs need to turn IGP metric check off.
All RRs to calculate the same best path so that primary and shadow RRs do not advertise the same path

solution	RR(config-router-af)#bgp bestpath igp-metric ignore

RR2 advertises same path as RR1 !

Shadow Session

Note: second session from RR to RR-client (PE3) has diverse-path command in order to advertise 2nd best path



- Easy deployment – only RR needs diverse path code and new iBGP session per each extra path (CLI knob on RR)
- Shadow iBGP session does announce the 2nd best path
  - 2nd session between a pair of routers is no issue (use different loopback interfaces)

BGP PIC (Prefix Independent Convergence) Edge

Problem

- Convergence in flat FIB is prefix dependent
  - More prefixes -> more convergence time
- Classical convergence (flat FIB)
  - Routing protocols react - update RIB - update CEF table (for affected prefixes)
  - Time is proportional to # of prefixes

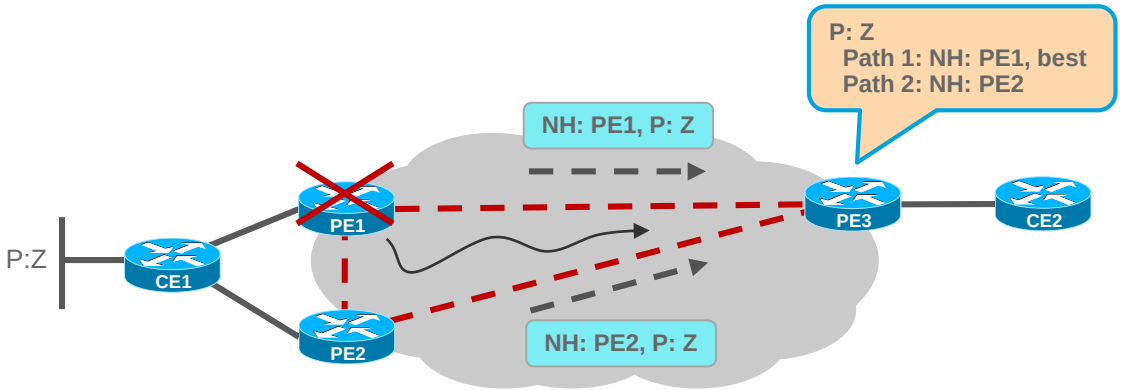
Result

- Improved convergence
- Reduce packet loss
- Have the same convergence time for all BGP prefixes (PIC)

Solution

- The idea of PIC:
  - In both SW and HW:
    - Pre-install a backup path in RIB
    - Pre-install a backup path in FIB
    - Pre-install a backup path in LFIB

MPLS VPN Dual Homed CE - No PIC Edge



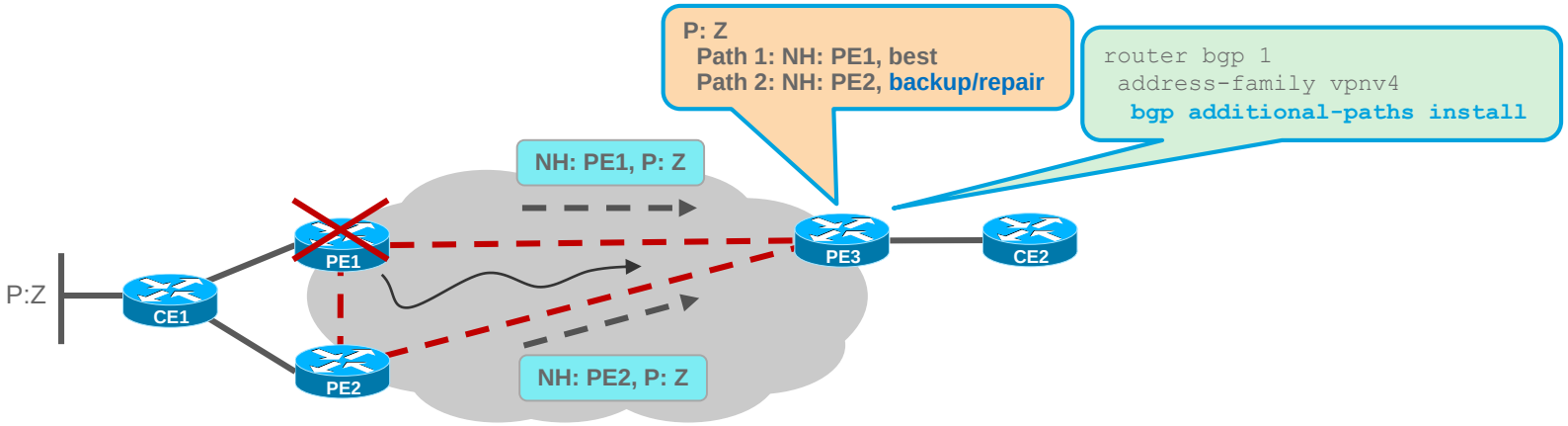
Steps in convergence

1. Egress PE goes down
2. IGP notifies ingress PE in sub-second

Steps in convergence on ingress PE

1. Ingress PE recomputes BGP bestpath
2. Ingress PE installs new BGP bestpath in RIB
3. Ingress PE installs new BGP bestpath in FIB
4. Ingress PE reprograms hardware

MPLS VPN Dual-Homed CE - PIC Edge



Steps in convergence

1. Egress PE goes down
2. IGP notifies ingress PE in sub-second

Steps in convergence on ingress PE

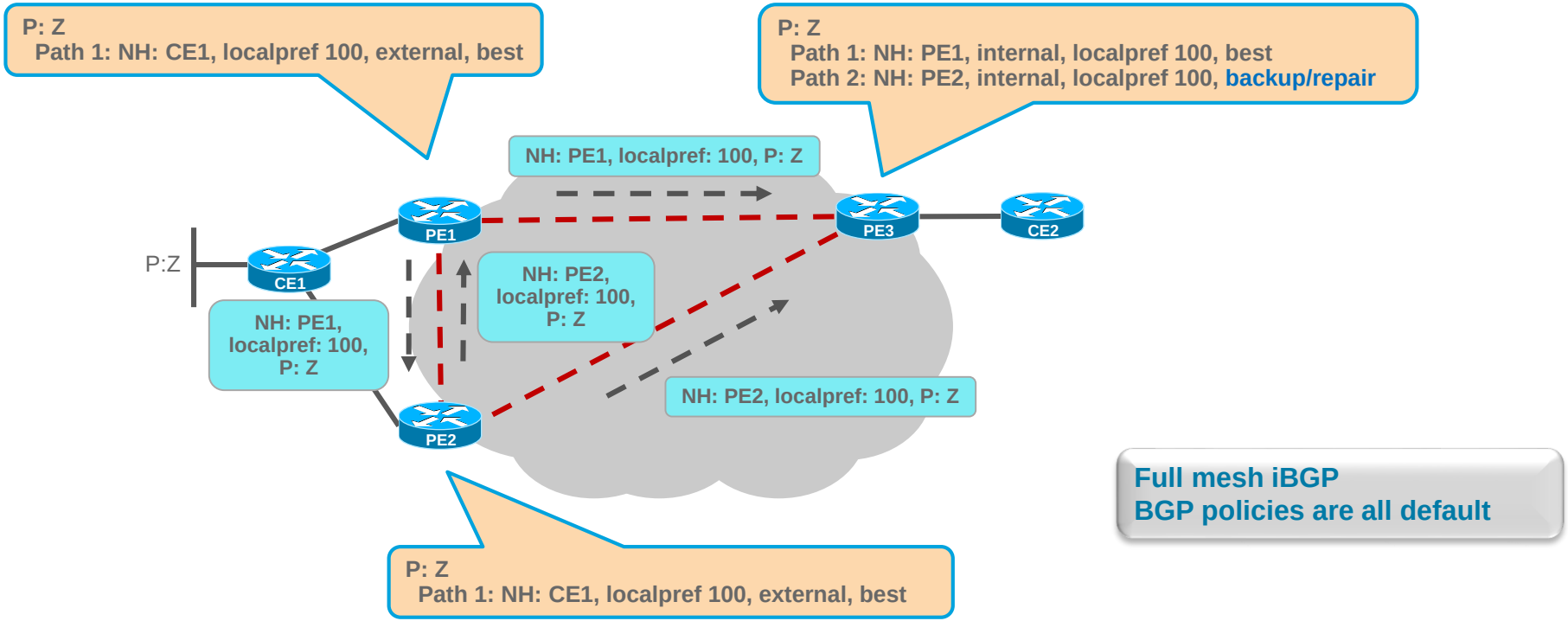
1. Switch to repair path with new Next Hop
2. Ingress PE reprograms hardware

We eliminate convergence dependence on:

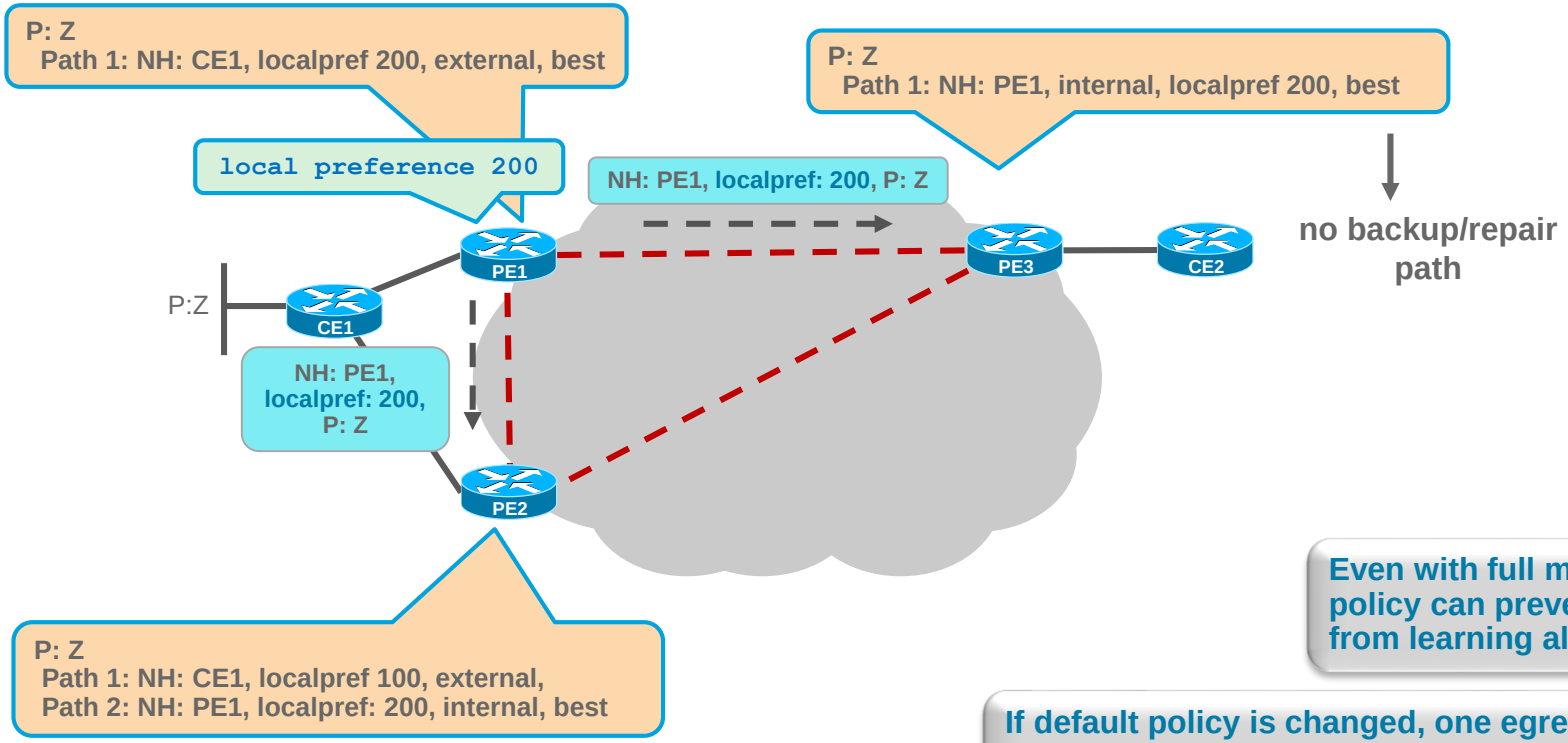
- Scanning of the BGP table
- Bestpath calculation (because there is a pre-computed backup/repair path)
- Time to generate and propagate updates (PE and RR)
- Updating the FIB (with PIC the FIB update is prefix independent)

this scales to the number of prefixes

No BGP Best External – Default BGP Policy



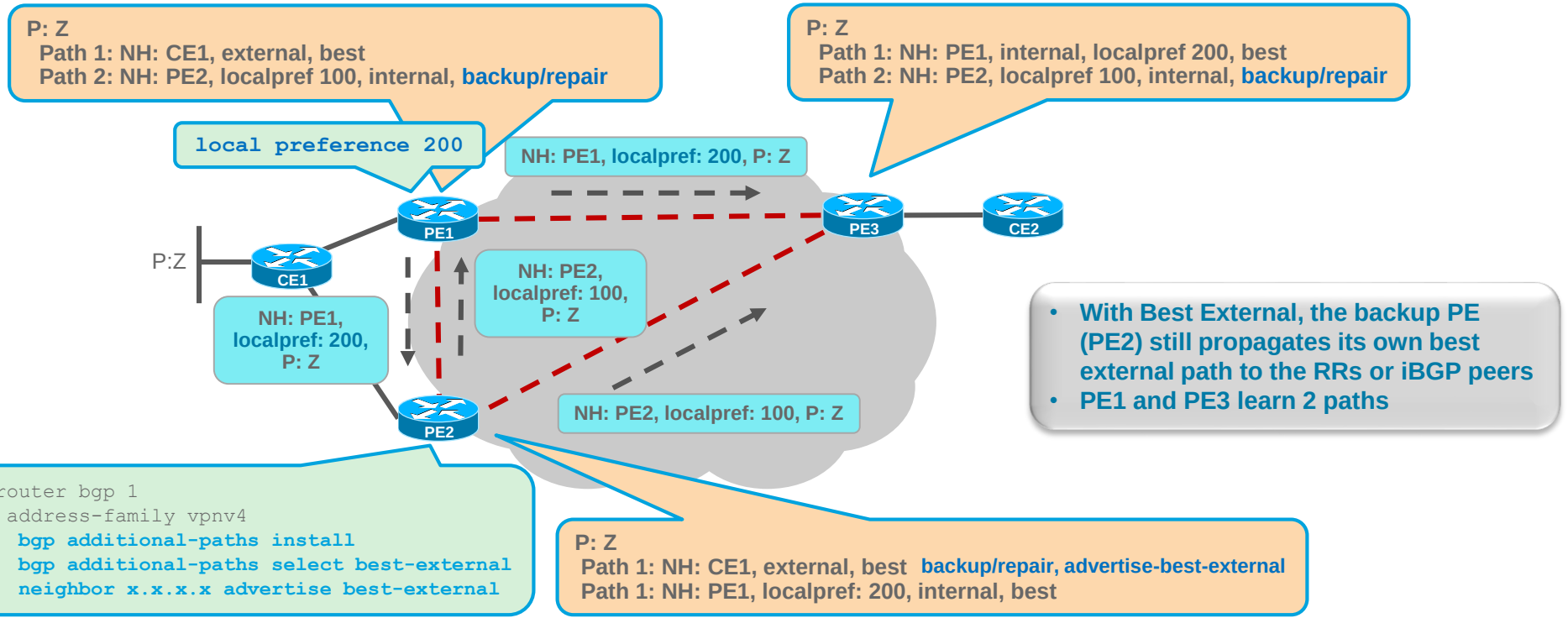
No BGP Best External - Changed BGP Policy



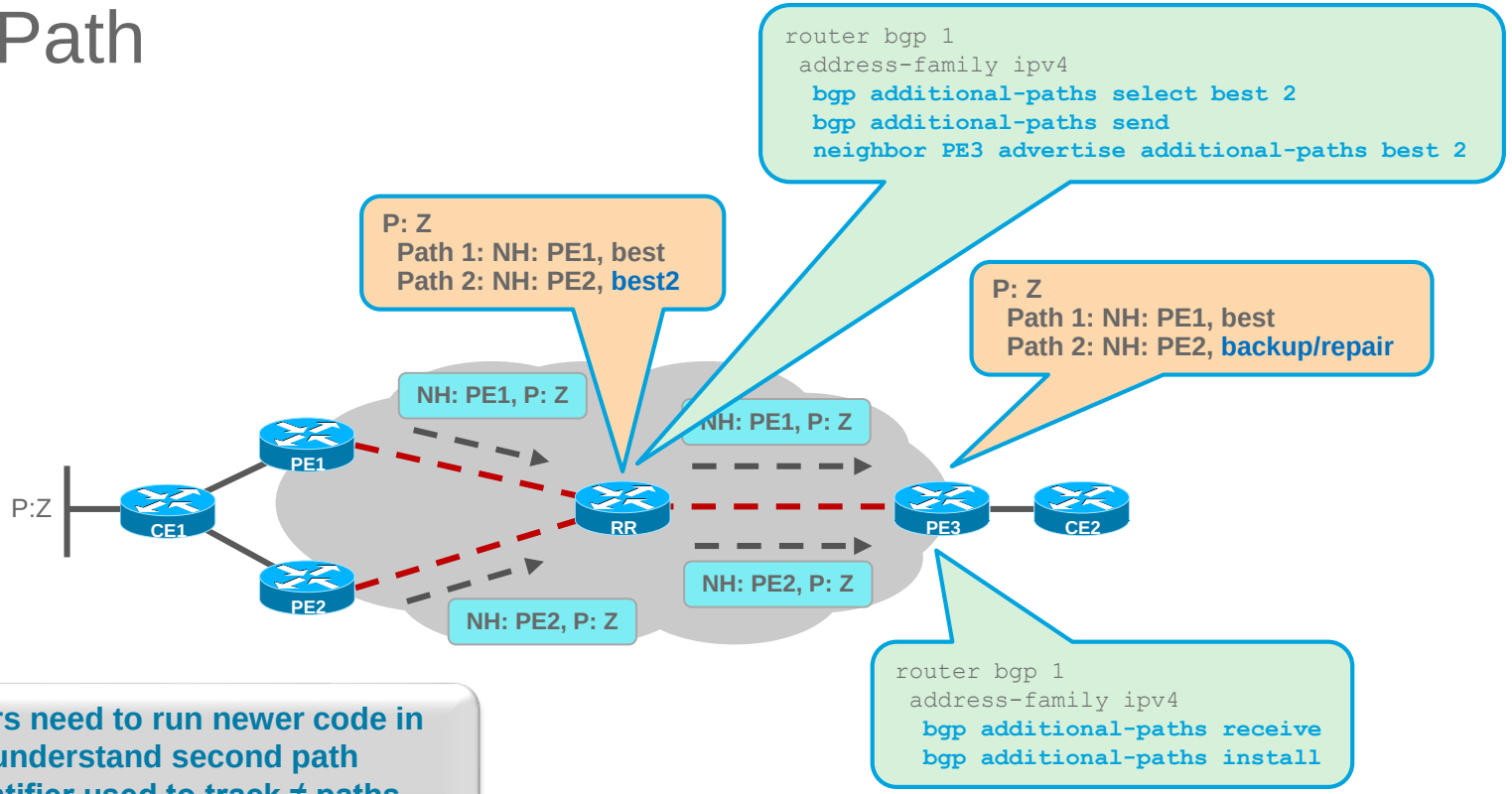
Even with full mesh in iBGP, policy can prevent egress PE from learning all paths

If default policy is changed, one egress PE could have iBGP path to other egress PE as best path and not its own external BGP path

BGP Best External - Changed BGP Policy



ADD Path



- PE routers need to run newer code in order to understand second path
- Path-identifier used to track ≠ paths

Add Path - Possibilities

add-all-path

- RR will do the first best path computation and then send all paths to the border routers
- Pros
  - all paths are available on border routers
- Cons
  - all paths stored
  - more BGP info is exchanged
- Usecase: ECMP, hot potato routing

```
bgp additional-paths select all
```

add-n-path

- RR will do best path computation for up to n paths and send n paths to the border routers
- This is the only mandatory selection mode
- Pros
  - less storage used for paths
  - less BGP info exchanged
- Cons
  - more best path computation
- Usecase: Primary + n-1 backup scenario
  (n is limited to 3 (IOS) or 2 (IOS-XR), to preserve CPU power) = fast convergence

```
bgp additional-paths select best<N>
```

multipath

- RR will do the first best path computation and then send all multipaths to the border routers
- Use case: load balancing and primary + backup scenario

IOS-XR only

Add-Path - IOS-XR

For Your Reference

- Path selection is configured in a route-policy
- Global command, per address family, to turn on add-path in BGP
- Configuration in VPNv4 mode applies to all VRF IPv4-Unicast AF modes unless overridden at individual VRFs

example config

```
router bgp 1
 address-family vpnv4
  additional-paths install backup     (deprecated)
  additional-paths advertise
  additional-paths receive
  additional-paths selection route-policy apx
```

example RPL config

```
route-policy ap1
 if community matches-any (1:1) then
  set path-selection backup 1 install
 elseif destination in (10.1.0.0/16, 10.2.0.0/16)then
 set path-selection backup 1 advertise install
endif

route-policy ap2
 set path-selection all advertise

route-policy ap3
 set path-selection multipath advertise

route-policy ap4
 set path-selection backup 1 install multipath-protect advertise
```

add-n-path

add-all-path

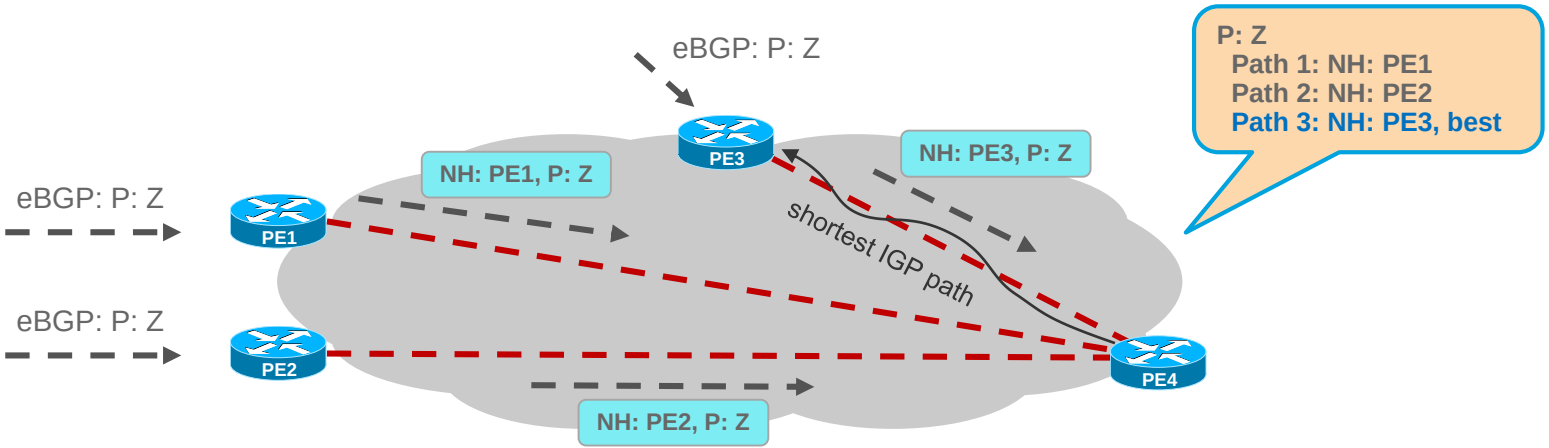
multipath

needed to have a non-multipath path as backup path

Cisco live!

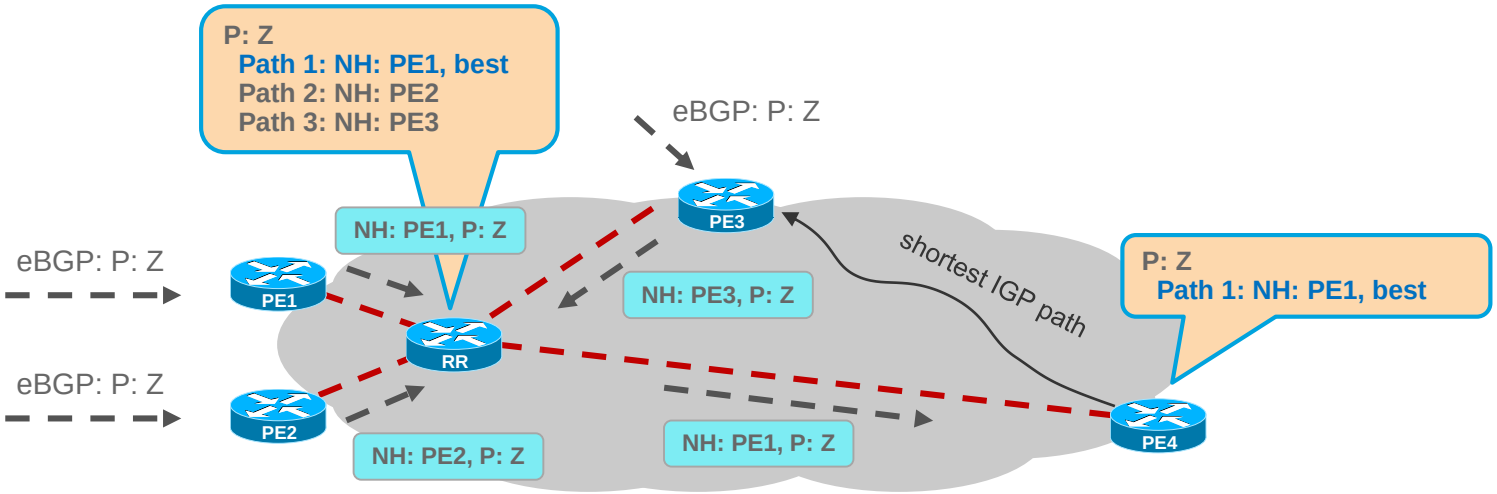
Hot Potato Routing - No RR

- Hot potato routing = packets are passed on (to next AS) as soon as received
- Shortest path though own AS must be used
- In transit AS: same prefix could be announced many times from many eBGP peers

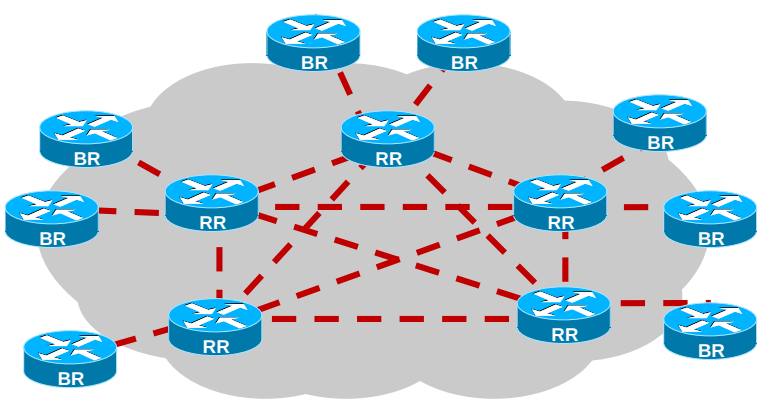


Hot Potato Routing - With RR

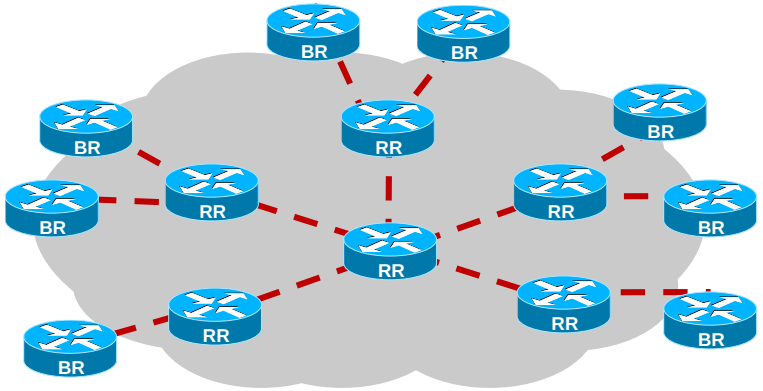
- Introducing RRs break hot potato routing
- Solutions: *Unique RD* for MPLS VPN or *Add Path*



Hot Potato Routing in Large Transit SP



add-path



- Large transit ISPs with full mesh iBGP between regional RRs and hub/spoke between local BR and RR
- Full mesh and global hot potato routing

- *add-all-path* could be deployed between centralized and regional RR's
- Also possible: remove the need for regional RR if all BR routers support add-path

BR Border Router

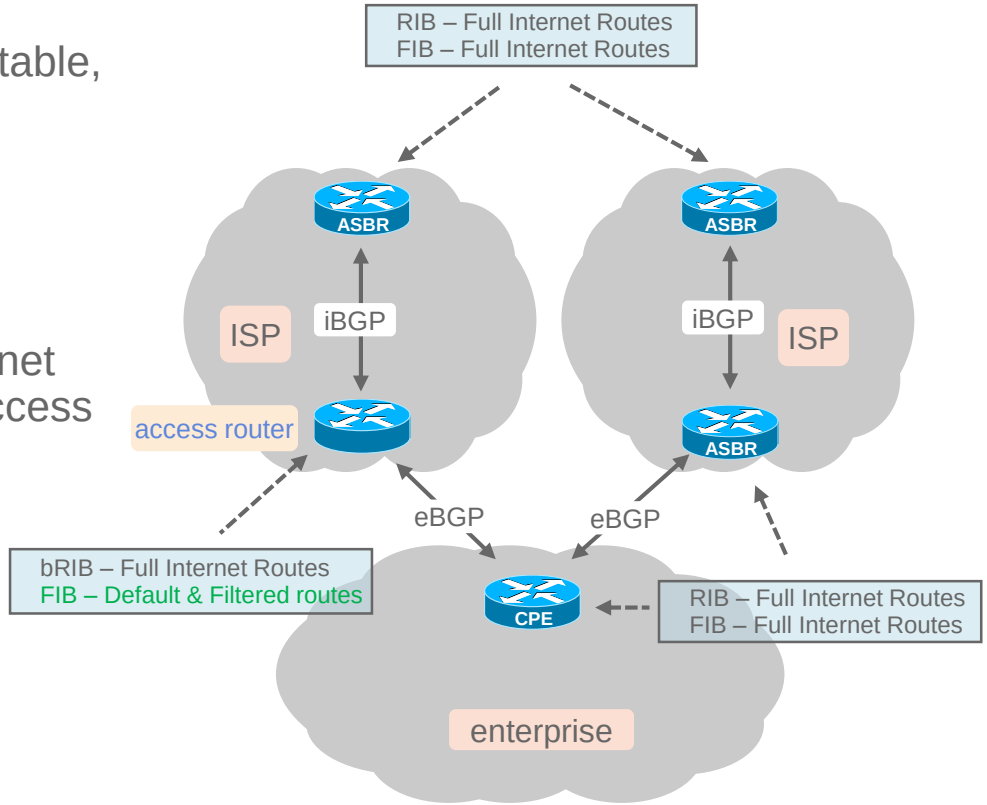
Deployment

BGP Selective Download

- Access router RIB holds full Internet routing table, but fewer routes in FIB
  - Example: ME switches, ASR900
- FIB holds default route and selective more specific routes
- Enterprise CPE devices will receive full Internet routes through their BGP peering with the access router(s)

configuration

```
router bgp 1

 address-family ipv4
  table-map filter-into-fib filter

route-map filter-into-fib deny 10
 match community 100

ip community-list 100 permit 65510:100
```



Path MTU Discovery (PMTUD)

- MSS (Max Segment Size) – Limit on the largest segment that can traverse a TCP session
  - Anything larger must be fragmented & re-assembled at the TCP layer
  - MSS is 536 bytes by default for client BGP without PMTUD
  - Enable PMTU for BGP with
    - Older command "`ip tcp path-mtu-discovery`"
    - Newer command "`bgp transport path-mtu-discovery`" (PMTUD now on by default)

- 536 bytes is inefficient for Ethernet (MTU of 1500) or POS (MTU of 4470) networks
  - TCP is forced to break large segments into 536 byte chunks
  - Adds overheads
  - Slows BGP convergence and reduces scalability

- TCP MSS set per neighbor (IOS-XR 5.4)

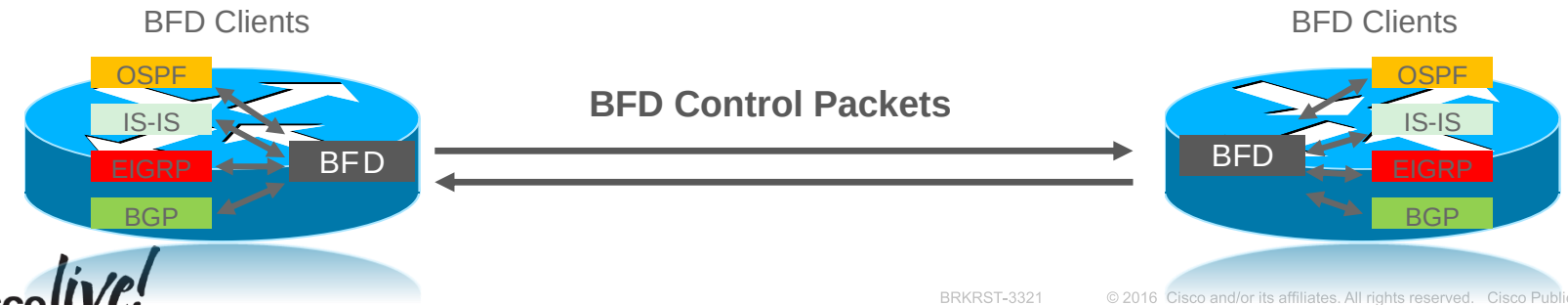
Session/Timers

- Timers = keepalive and holdtime
  - Default is ok
  - Smallest is 3/9 for keepalive/holdtime
  - Scaling <> small timers
- Use BFD
  - Built for speed
  - When failure occurs, BFD notifies BFD client (in 10s of msecs)

- Do not use Fast Session Deactivation (FSD)
  - Tracks the route to the BGP peer
  - A temporary loss of IGP route, will kill off the iBGP sessions
  - Very dangerous for iBGP peers
    - IGP may not have a route to a peer for a split second
    - FSD would tear down the BGP session
  - It is off by default

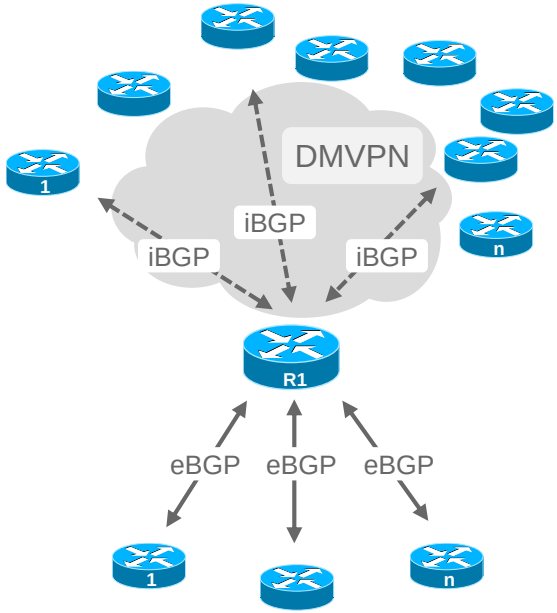
```
neighbor x.x.x.x fall-over
```

  - Next Hop Tracking (NHT), enabed by default, does the job fine



Dynamic Neighbors

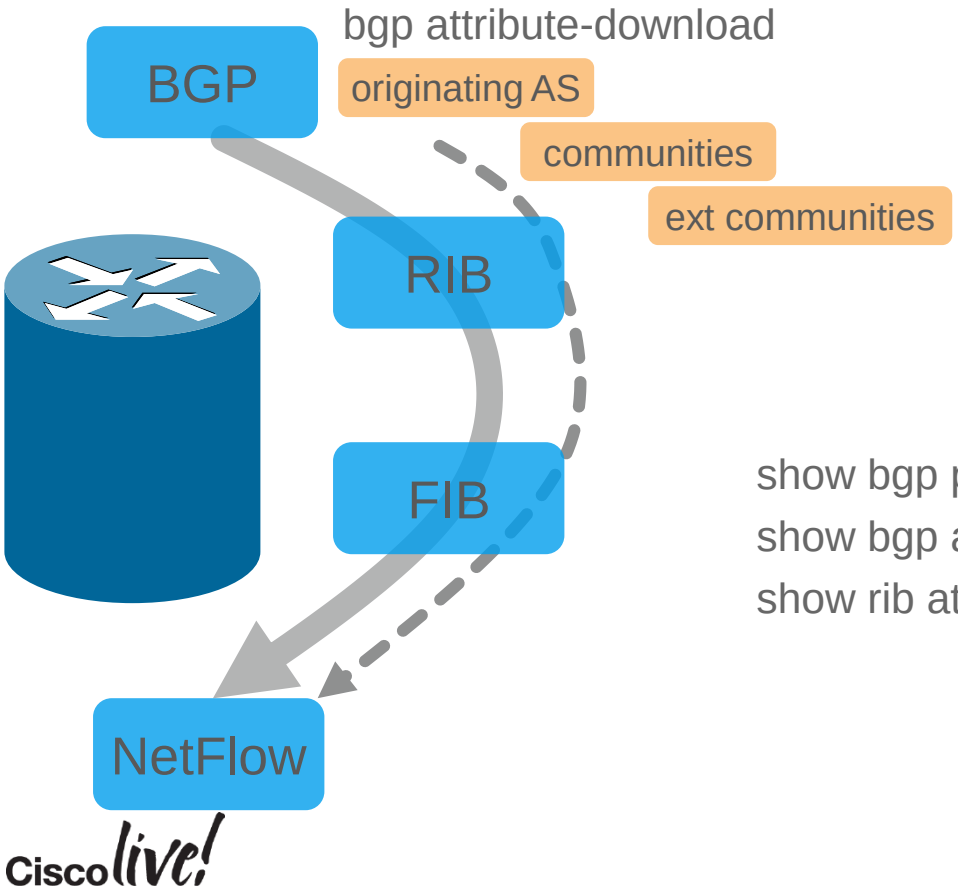
- Remote peers are defined by IP address range
- Less configuration for defining neighbors
- Remote initiate BGP session
- Enterprise networks (DMVPN, ...)
- iBGP and limited eBGP (limited nr of ASNs)



configuration

```
router bgp 1
 bgp listen range 192.168.0.0/16 peer-group 192-16
 bgp listen range 10.1.1.0/24 peer-group 10-24
 bgp listen limit 1000
 neighbor 10-24 peer-group
 neighbor 10-24 remote-as 1
 neighbor 192-16 peer-group
 neighbor 192-16 remote-as 2 alternate-as 3 4 5 6 7
 neighbor 192-16 ebgp-multihop 2
 neighbor 192-16 update-source Loopback0
```

BGP Attribute Download



- Attributes communities, extended communities, and AS-path are downloaded to the RIB & FIB

show bgp process performance-statistics detail

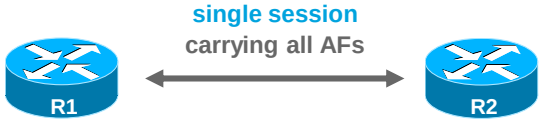
show bgp attribute-key

show rib attributes summary

Multisession

Multisession

- BGP Multisession = multiple BGP (TCP) sessions between 2 BGP speakers
  - Even if there is only one BGP neighbor statement defined between the BGP speakers in the configuration
- Introduced with Multi Topology Routing (MTR)
  - One session per topology
- Now: possibility to have one session per AF/group of AFs
  - Good for incremental deployment of AFs
    - Avoids a BGP reset
    - But multisession needs to be enabled beforehand
  - Good for troubleshooting
  - Good for issues when BGP session resets
    - For example “malformed update”
  - Not so good for scalability
  - IOS only and not enabled by default



Multisession

IOS

For Your Reference

capability

```
BGP: 10.100.1.2 passive rcvd OPEN w/ optional parameter type 2 (Capability)
len 3
BGP: 10.100.1.2 passive OPEN has CAPABILITY code: 131, length 1
BGP: 10.100.1.2 passive OPEN has MULTISESSION capability, without grouping
```

multisession for MTR

```
R2#show bgp ipv4 unicast neighbors
BGP neighbor is 10.100.1.1,  remote AS 1, internal link
…
  BGP multisession with 3 sessions (3 established), first up for 00:05:43
  Neighbor sessions:
    3 active, is multisession capable
    Session: 10.100.1.1 session 1
      Topology IPv4 Unicast
    Session: 10.100.1.1 session 2
      Topology IPv4 Unicast voice
    Session: 10.100.1.1 session 3
      Topology IPv4 Unicast video
```

1 session per topology

multisession without MTR

```
R2#show ip bgp neighbors 10.100.1.1 | include session|address family
  BGP multisession with 3 sessions (3 established), first up for 00:02:29
  Neighbor sessions:
    3 active, is multisession capable
    Session: 10.100.1.1 session 1
    Session: 10.100.1.1 session 2
    Session: 10.100.1.1 session 3
    Route refresh: advertised and received(new) on session 1, 2, 3
    Multisession Capability: advertised and received
  For address family: IPv4 Unicast
    Session: 10.100.1.1 session 1
    session 1 member
  For address family: IPv6 Unicast
    Session: 10.100.1.1 session 2
    session 2 member
  For address family: VPNv4 Unicast
    Session: 10.100.1.1 session 3
    session 3 member
```

1 session per address family

Cisco live!

Multisession

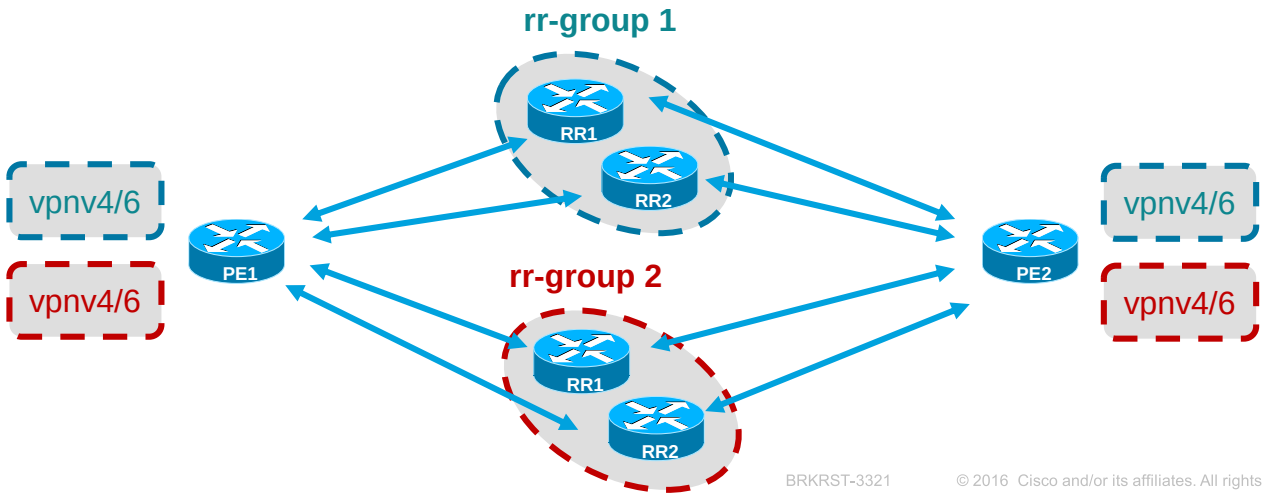
Conclusion

- Increases # of TCP sessions
- Not really needed
- Current default behavior = multisession is off
  - Can be turned on by “neighbor x.x.x.x transport multi-session”
- Makes sense to have IPv4 and IPv6 on seperate TCP sessions
  - IPv6 over IPv4 (or IPv4 over IPv6) can be done, but next hop mediation is needed

MPLS VPN

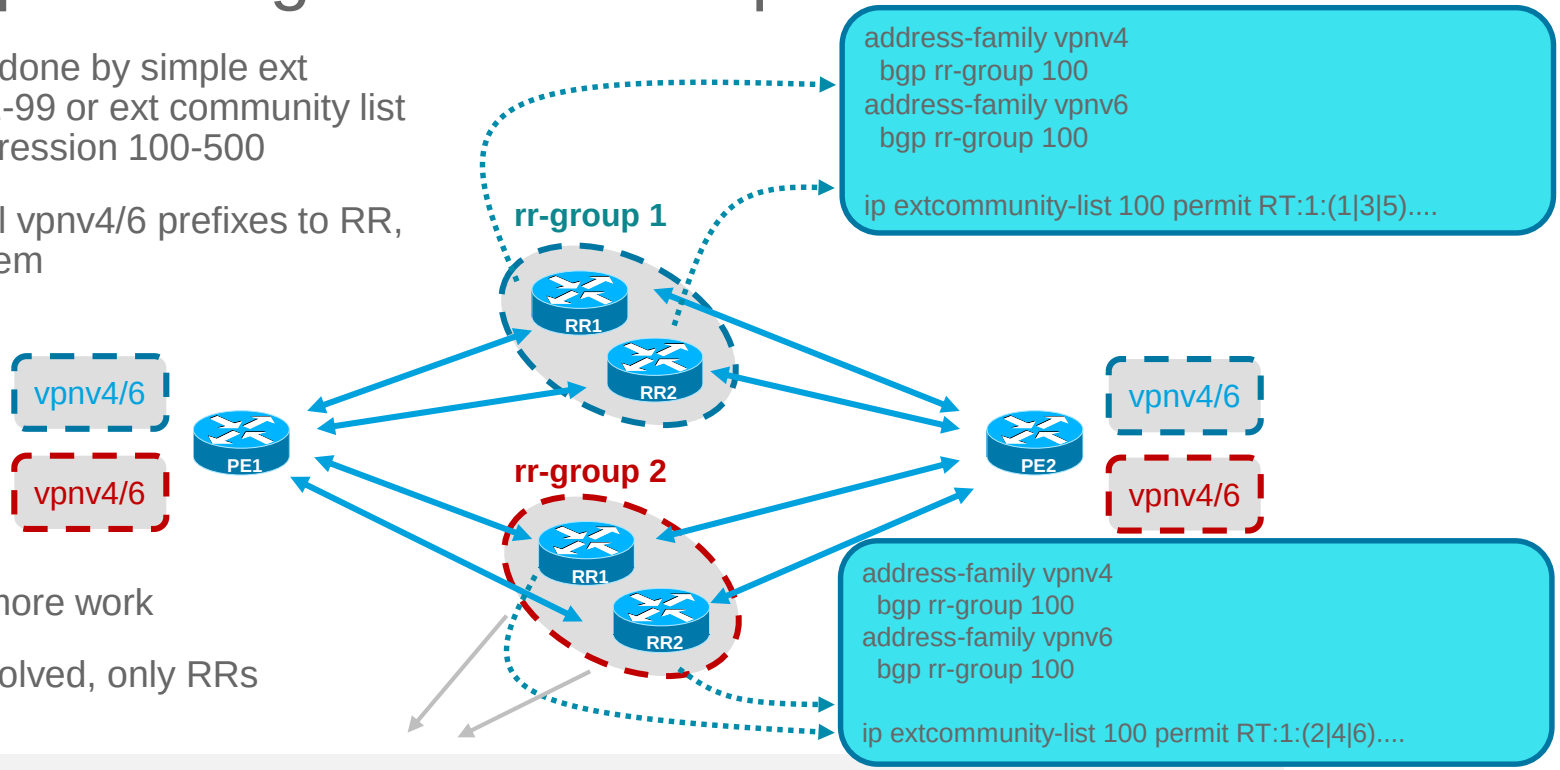
RR-groups

- Use one RR (set of RRs) for a subset of prefixes
  - By carving up range of RTs
- Only for vpnv4/6
  - RR only stores and advertises the specific range of prefixes
- Less storage on RR, but more RRs needed + more peerings



RR-groups Configuration Example

- Dividing of RTs done by simple ext community list 1-99 or ext community list with regular expression 100-500
- PEs still send all vpnv4/6 prefixes to RR, but RR filters them



- Dividing RT = more work
- PEs are not involved, only RRs

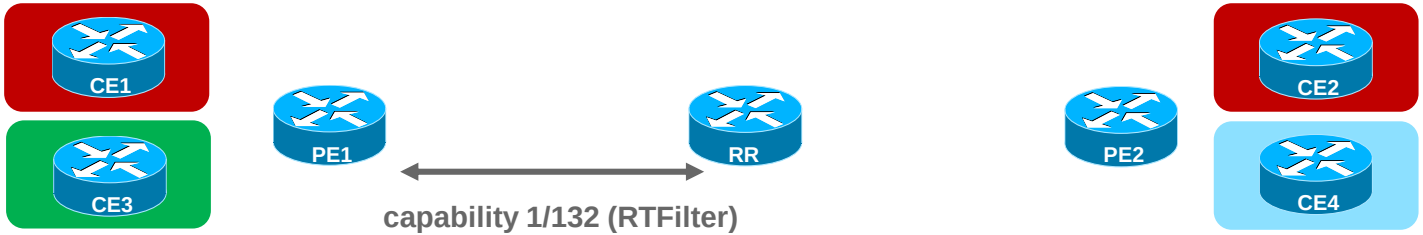
```
BGP(4): 10.100.1.1 rcvd UPDATE w/ attr: nexthop 10.100.1.1, origin ?, localpref 100, metric 0, extended community RT:1:10001
BGP(4): 10.100.1.1 rcvd 1:10001:100.1.1.2/32, label 22 -- DENIED due to:  extended community not supported;
```

Route Target Constraint (RTC)

- Current behavior:
  - RR sends all vpnv4/6 routes to PE
  - PE routers drops vpnv4/6 for which there is no importing VRF
- RTC behavior: RR sends only “wanted” vpnv4/6 routes to PE
  - “wanted”: PE has VRF importing the Route Targets for the specific routes
  - RFC 4684
  - New AF “rtfilter”
- Received RT filters from neighbors are translated into oubound filtering policies for vpnv4/6 prefixes
- The RT Filtering information is obtained from the VPN RT import list
- Result: RR does not send vpnv4/6 unnecessarily prefixes to the PE routers

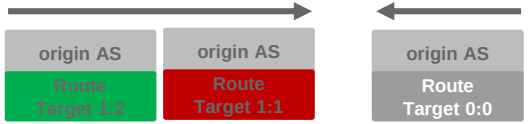
Route Target Constraint (RTC)

BGP capability exchange OPEN message



AF RTFilter exchange

MP_REACH_NLRI
AF RTFilter



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

Cisco live!

Route Target Constraint (RTC)

- Results
  - Eliminates the waste of processing power on the PE and the waste of bandwidth
  - Number of vpnv4 formatted message is reduced by 75%
  - BGP Convergence time is reduced by 20 - 50%
  - The more sparse the VPNs (few common VPNs on PEs), the more performance gain
- Note: PE and RR need the support for RTC
  - Incremental deployment is possible (per PE)
  - Behavior towards non-RT Constraint peers is not changed
- Note
  - RTC clients of RR with different set of importing RTs will be in the same update group on the RR
    - In IOS-XR, different filter group under same subgroup

Legacy PE RT Filtering

- Problem: If one PE does not support RTC (legacy prefix), then all RRs in one cluster must store and advertise all vpn prefixes to the PE
- Solution: Legacy PE sends special prefixes to mimic RTC behavior, without RTC code

Legacy PE

- Collect import RTs
- Create route-filter VRF (same RD for all these VRFs across all PEs)
- Originate special route-filter route(s) with
  - the import RTs attached
  - one of 4 route-filter communties
  - NO-ADVERTISE community

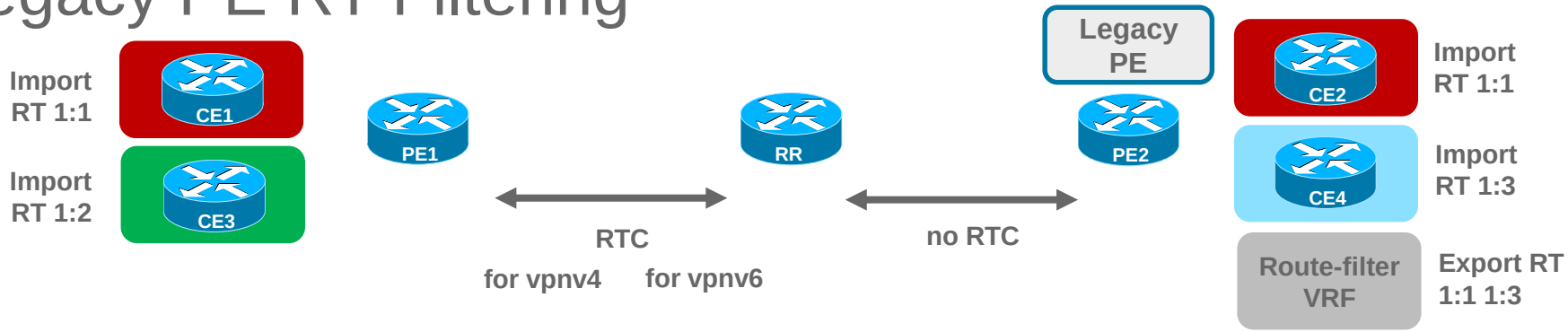
RR

- The presence of the community triggers the RR to extract the RTs and build RT membership information
- RR only advertises wanted vpn prefixes towards legacy PE

4 route-filter communties

```
0xFFFF0002 ROUTE_FILTER_TRANSLATED_v4
0xFFFF0003   ROUTE_FILTER_v4
0xFFFF0004 ROUTE_FILTER_TRANSLATED_v6
0xFFFF0005   ROUTE_FILTER_v6
```

Legacy PE RT Filtering



legacy PE sends route-filter VRF route(s) with unique RD, route-filter community and importing RTs

vpnv4/6 update with prefix(es) RT membership information

```
9999:9999:9.9.9.9/32

Community: 4294901762
Extended Community: RT:0.1.0.0:1
RT:0.1.0.0:3
no-export no-advertise
```

RD:prefix

One of 4 route-filter communities
All import RTs of the legacy PE
NO-ADVERTISE NO-EXPORT community

AF vpnv4/6 prefixes exchange

PE1 sends all its vpnv4/6 prefixes to RR

RR sends only RED (not green) vpnv4/6 prefixes to PE2

Legacy PE RT Filtering - Configuration

For Your Reference

RR config

```
router bgp 1
 address-family vpnv4
  neighbor 10.100.1.2 route-reflector-client
  neighbor 10.100.1.2 accept-route-legacy-rt
```

new code

old code

Import RT 1:1

Import RT 1:3

Route-filter VRF

Export Map 1:1 1:3

Legacy PE config

```
ip vrf route-filter
 rd 9999:9999
 export map SET_RT

router bgp 1
 address-family vpnv4
  neighbor 10.100.1.3 route-map legacy_PE out
 address-family ipv4 vrf route-filter
  network 9.9.9.9 mask 255.255.255.255

ip route vrf route-filter 9.9.9.9 255.255.255.255 Null0
ip prefix-list match_RT_1 seq 5 permit 9.9.9.9/32

route-map SET_RT permit 10
 match ip address prefix-list match_RT_1
 set community 4294901762 (equals 0xFFFF0002)
 set extcommunity rt  0.1.0.0:1 0.1.0.0:3 additive

route-map legacy_PE permit 10
 match ip address prefix-list match_RT_1
 set community no-export no-advertise additive
```

Cisco live!

Full Internet in a VRF?

- Why? Because design dictates it
- Unique RD, so that RR can advertise 2 paths?

PRO

- Remove Internet routing table from P routers
- Security: move Internet into VPN, out of global
- Added flexibility
- More flexible DDOS mitigation

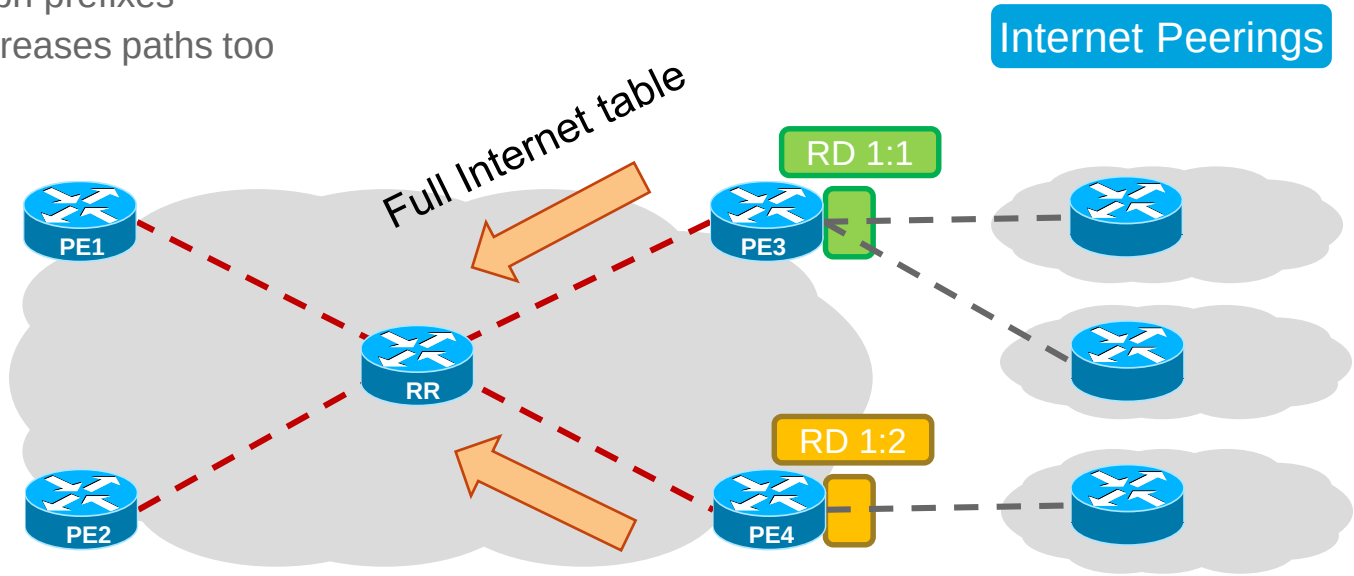
CON

- Increased memory and bandwidth consumption

- Platform must support enough MPLS labels
  - Label allocation is per-prefix by default
  - Perhaps per-ce or per-vrf label allocation is wanted here
  - Now also per-CE and per-VRF label allocation for 6PE (in IOS-XR)

Full Internet in a VRF?

- Considerations
- Two Internet gateways for redundancy
- RRs are present: unique RDs needed
  - Then double # vpn prefixes
  - ADD-PATHS increases paths too

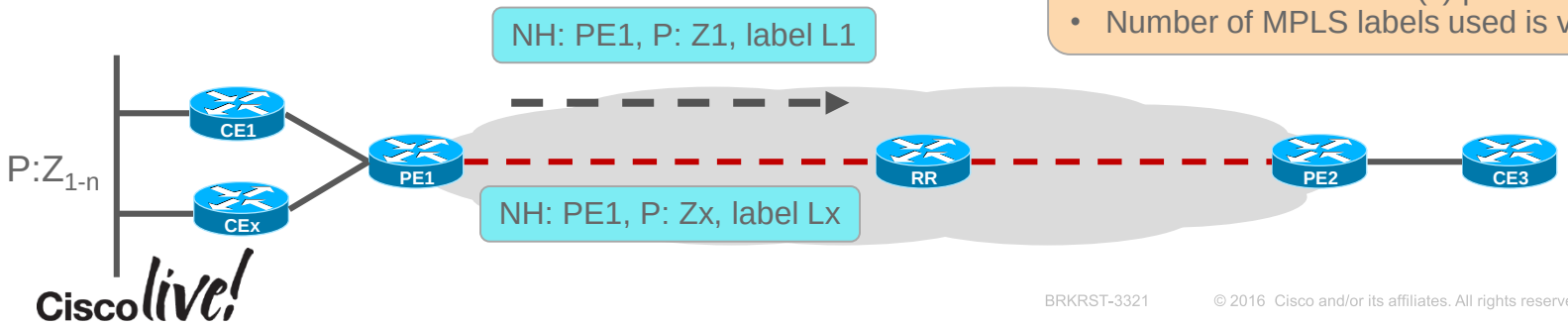


Per-CE Label

- One unique label per prefix is always the default
- Per-CE : one MPLS label per next-hop (so per connected CE router)
- No IP lookup needed after label lookup
- Caveats
  - No granular load balancing because the bottom label is the same for all prefixes from one CE, if platform load balances on bottom label
  - eBGP load balancing & BGP PIC is not supported (it makes usage of label diversity), unless resilient per-ce label
  - Only single hop eBGP supported, no multihop

2 CEs = 2 labels

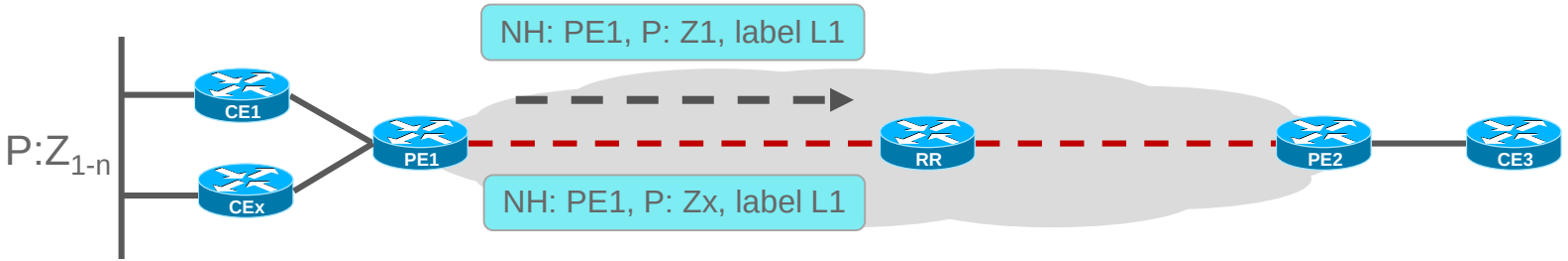
- Number of prefixes (n) is much larger than number of CE routers (x) per VPN
- Number of MPLS labels used is very low



Per-VRF Label

- Per-VRF : one MPLS label per VRF (all CE routers in the VRF)
  - Con: IP lookup needed after label lookup
  - Con: No granular load balancing because the bottom label is the same for all prefixes, if platform load balances on bottom label
  - Potential forwarding loop during local traffic diversion to support PIC
  - No support for EIBGP multipath

Number of MPLS labels used per VRF is 1 !

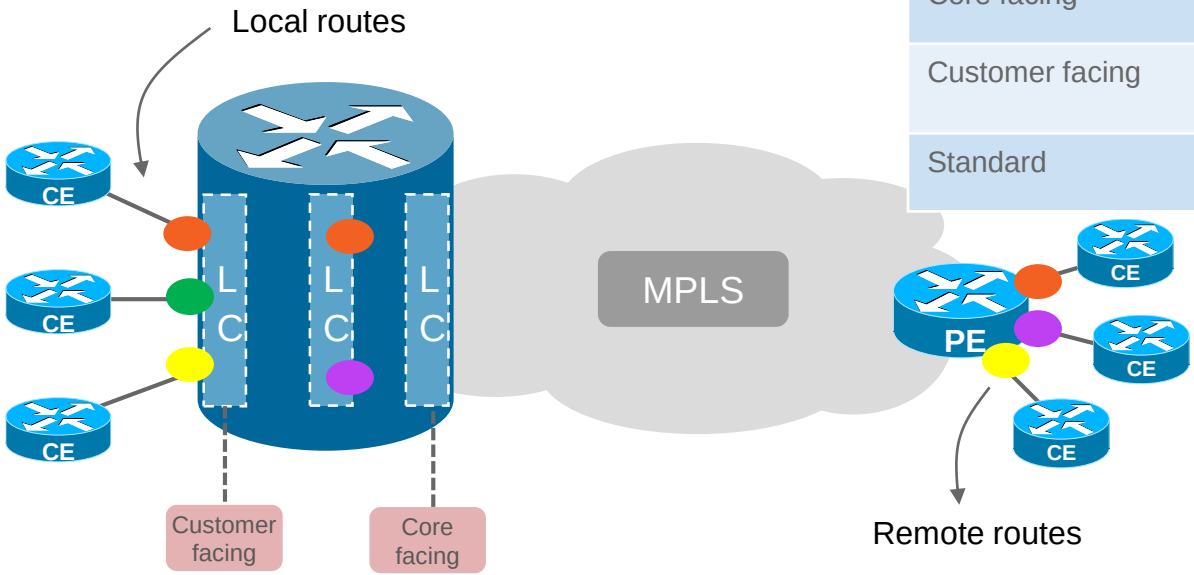


IOS-XR can do selective label mode (prefix | CE | VRF) with RPL

Selective VRF Download (SVD)

- Download to a line card only those prefixes and labels from a VRF that are actively required to forward traffic through that line card
- In IOS-XR 4.2.0 and enabled by default

Linecard Role	Which routes are present?
Core facing	routes for all VRFs, but only the local routes
Customer facing	routes only for VRFs which the LC is interested in (local and remote routes)
Standard	all routes are present



OS Enhancements

ASR9K: Scaling Enhancement

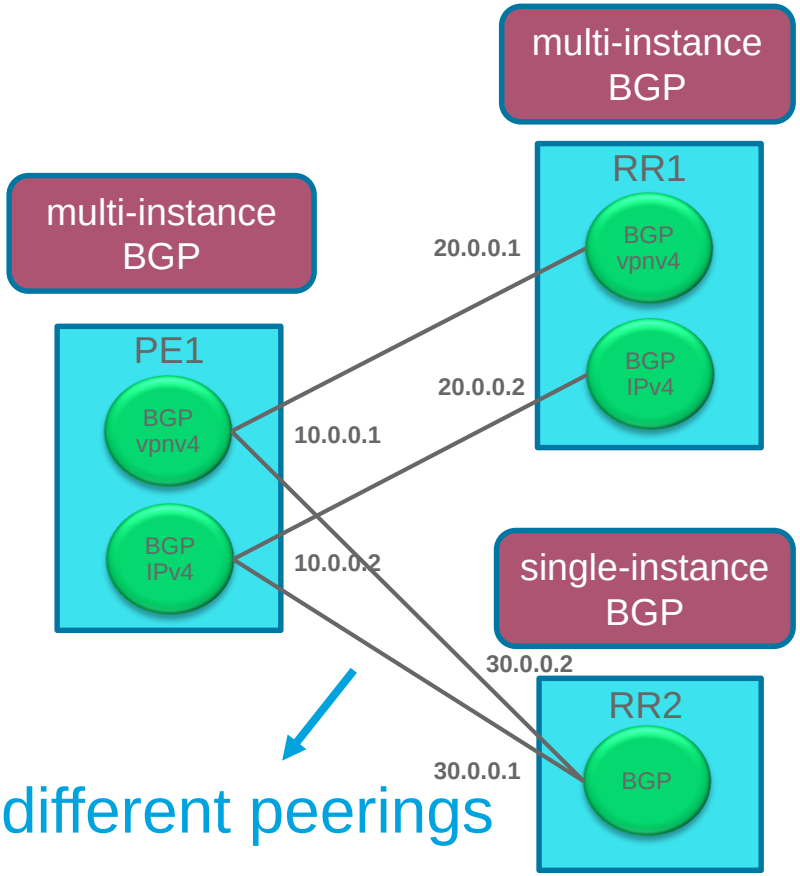
- BGP RIB Scale enhancement in 5.1.1
  - Only for RSP440-SE
  - Reload is needed
- Get more virtual address space for BGP process
  - From 2 GB to 2.5 GB

```
RP/0/RSP1/CPU0:router(admin-config)#hw-module profile scale ?
  default  Default scale profile
  l3       L3 scale profile
  l3xl     L3 XL scale profile
```

Profile	Layer 3 (Prefixes)	Layer 2 (MAC Table)
default	Small (512k)	Large (512k)
l3	Large (1,000k)	Small (128k)
l3xl	Extra large (1,300k)	Minimal
l3xl (5.1.1 RSP3)	Extra large (2,500k)	Minimal

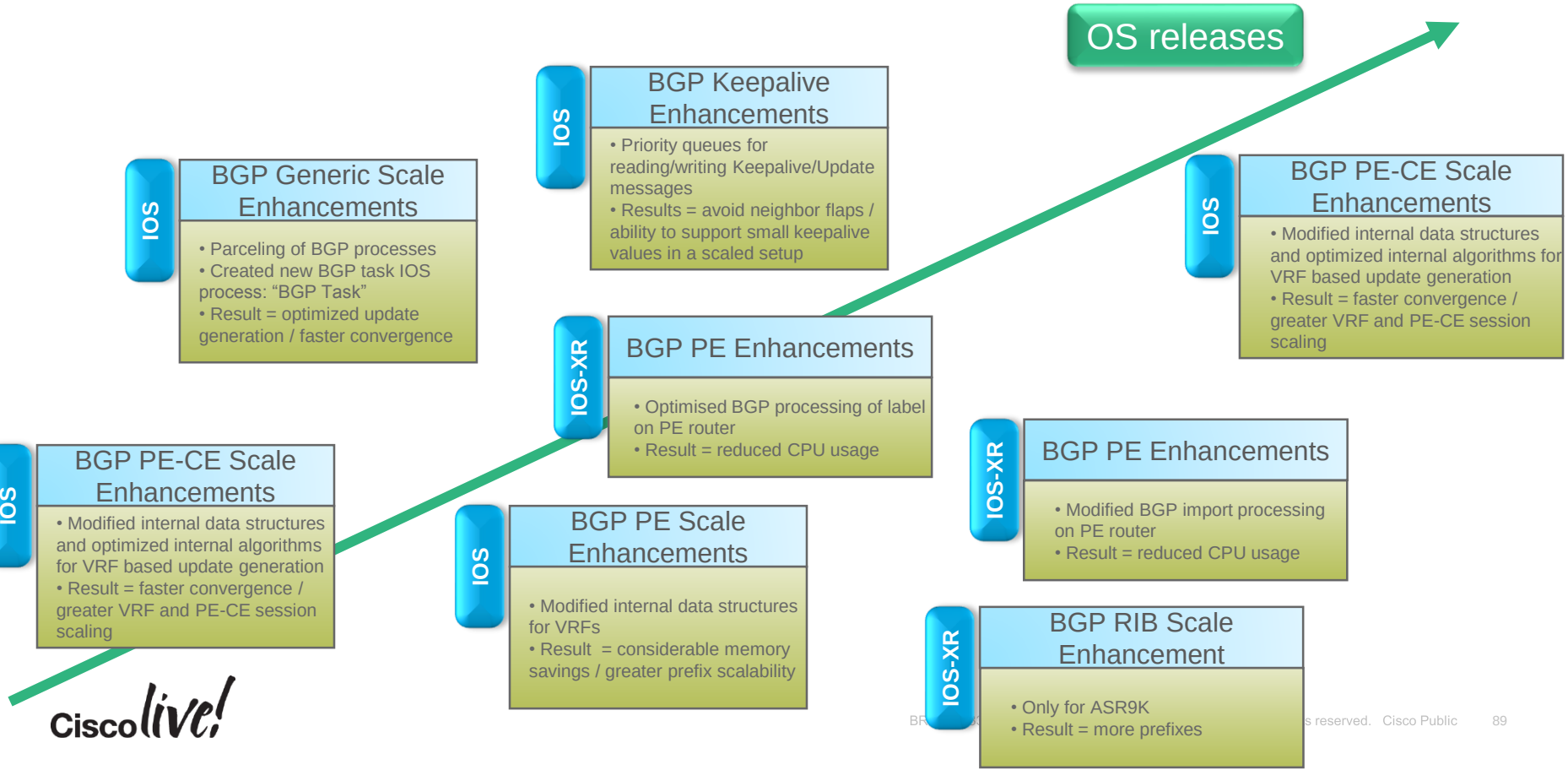
Multi-Instance BGP

- A new IOS-XR BGP architecture to support multiple BGP instances
- Each BGP instance is a separate process running on the same or a different RP/DRP node
- Different prefix tables
- Multiple ASNs are possible
- Solves the 32-bit OS virtual memory limit
- Different BGP routers: isolate services/AFs on common infrastructure
- Achieve higher prefix scale (especially on a RR) by having different instances carrying different BGP tables
- Achieve higher session scale by distributing the overall peering sessions between instances



OS Scaling Enhancements for BGP

For Your Reference



Cisco live!

When is the Boat Not Big Enough?

For Your Reference

Convergence

Measure: Prefix instability, Traffic drops, Table Versions, Timestamps

IOS	IOS-XR	NX-OS
	show bgp convergence	show bgp convergence detail
show bgp all summary	show bgp table	
	show bgp process performance-statistics detail	

Memory

IOS	IOS-XR	NX-OS
show bgp all summary	show bgp table	show bgp internal mem-stats detail - look for “Grand total”, “Private memory”, “Shared memory”
show processes memory sorted	show process memory <job-id> location <>	show system resource
	show watchdog memory-state	
	show memory compare start | end | report	
	show bgp scale	

CPU

IOS	IOS-XR	NX-OS
show processes cpu history show processes cpu | include BGP	show processes cpu show processes bgp show processes cpu | include bgp	show processes cpu history show processes cpu | include bgp show process cpu detailed <bgp pid>

Cisco live!

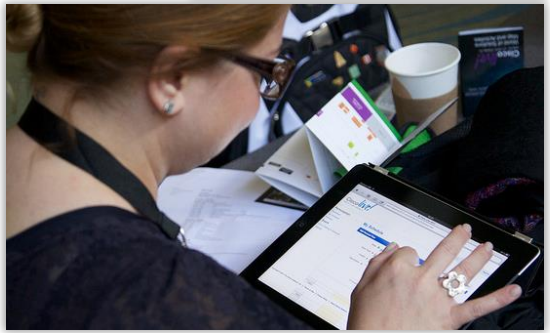
Call to Action

- Visit the World of Solutions for
  - Cisco Campus
  - Walk in Labs
  - Technical Solution Clinics
- Meet the Engineer
- Lunch and Learn Topics
- DevNet zone related sessions

Complete Your **Online Session Evaluation**

- Please complete your online session evaluations after each session. Complete 4 session evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt.

- All surveys can be completed via the Cisco Live Mobile App or the Communication Stations



Cisco live!

Thank you

We’re ready. Are you?

Slow Peer - Configuration

For Your Reference

Phase	Where?	Command
detection	per AF/per VRF	`bgp slow-peer detection [threshold <seconds>]` default is 5 min
	per peer	`neighbor {<nbr-addr>/<peer-grp-name>} slow-peer detection [threshold < seconds >]` `neighbor {<nbr-addr>/<peer-grp-name>} disable`
	per peer policy template	`slow-peer detection [threshold < seconds >]`
protection	static per neighbor/ per peer-group	`neighbor {<nbr-addr>/<peer-grp-name>} slow-peer split-update-group static`
	static via peer policy template	`slow-peer split-update-group static`
	automatic per AF/ per VRF	`bgp slow-peer split-update-group dynamic [permanent]`
	automatic per neighor	`neighbor {<nbr-addr>/<peer-grp-name>} slow-peer split-update-group dynamic [permanent]`
	automatic per peer policy template	`slow-peer split-update-group dynamic [permanent]`

permanent = peer is not moved back automatically to the update group

Slow Peers: Displaying/Clearing

For Your Reference

- CLI to display slow peers
- Applicable to all address families
  - show bgp all summary slow
  - show bgp ipv4 unicast neighbors slow
  - show bgp ipv4 unicast update-group summary slow
- This is a forced clear of the slow-peer status; the peer is moved to the original update group
- Needed when the permanent keyword is configured
  - clear bgp AF {unicast|multicast} * slow
  - clear bgp AF {unicast|multicast} <AS number> slow
  - clear bgp AF {unicast|multicast} peer-group <group-name> slow
  - clear bgp AF {unicast|multicast} <neighbor-address> slow

Slow Peer Mechanism Details

For Your Reference

Clearing

- CLI to clear
- This is a forced clear of the slow-peer status; the peer is moved to the original update group
- Needed when the permanent keyword is configured

- clear bgp AF {unicast|multicast} * slow
- clear bgp AF {unicast|multicast} <AS number> slow
- clear bgp AF {unicast|multicast} peer-group <group-name> slow
- clear bgp AF {unicast|multicast} <neighbor-address> slow

Route-Refresh Update Group: When is a Route Refresh Request Sent?

For Your Reference

- A route refresh request is sent, when:
  - a user types clear ip bgp [AF] {*|peer} in
  - a user types clear ip bgp [AF] {*|peer} soft in
  - adding or changing the inbound filtering on the BGP neighbor
    - via route-map
  - configuring allowas-in for the BGP neighbor
  - configuring soft-configuration inbound on the BGP neighbor
  - in MPLS VPN (for AFI/SAFI 1/128)
    - a user adds a route-target import to a VRF
  - in 6VPE (for AFI/SAFI 2/128)
    - a user adds a route-target import to a VRF

Route Reflector

For Your Reference

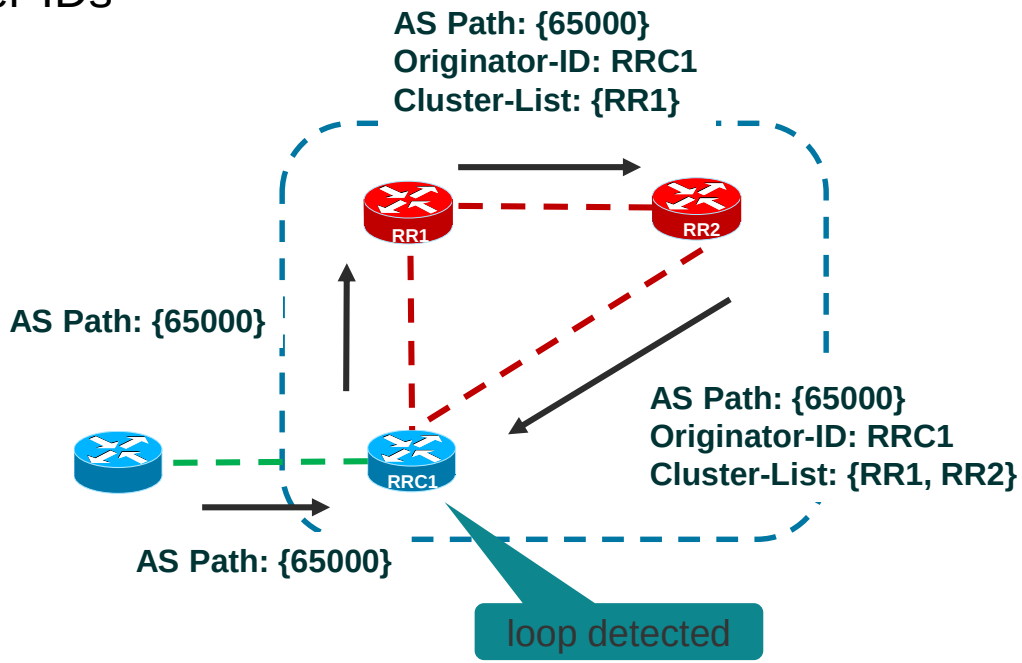
Loop Prevention

- Because we have RRs and same prefix can be advertised multiple times within iBGP cloud: loop prevention needed in iBGP cloud
- Two BGP attributes, two ways
  - Originator ID
    - Set to the router ID of the router injecting the route into the AS
    - Set by the RR
  - Cluster List
    - Each route reflector the route passes through adds their cluster-ID to this list
    - Cluster-id = Router ID by default
    - “bgp cluster-id x.x.x.x” command to set cluster-id
- Router discard routes if:
  - If ORIGINATOR-ID = our ROUTER-ID
  - If CLUSTER_LIST contains our CLUSTER-ID

Route Reflector

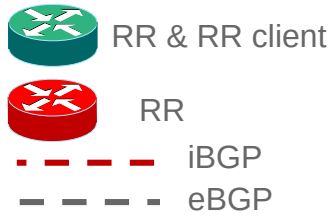
Loop Prevention

- RR1 and RR2 have different cluster-IDs

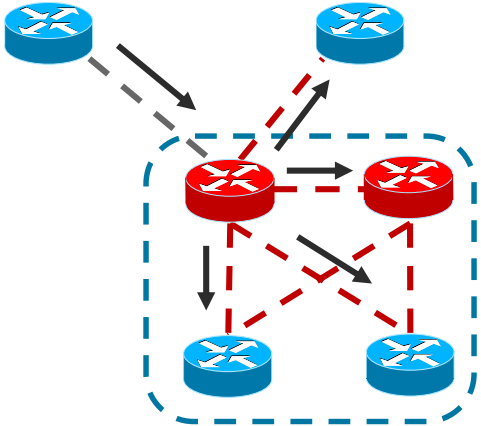


Route Reflector

Route Advertisement

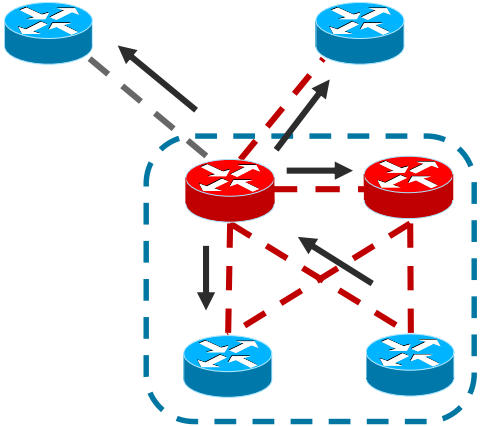


prefix coming from eBGP peer



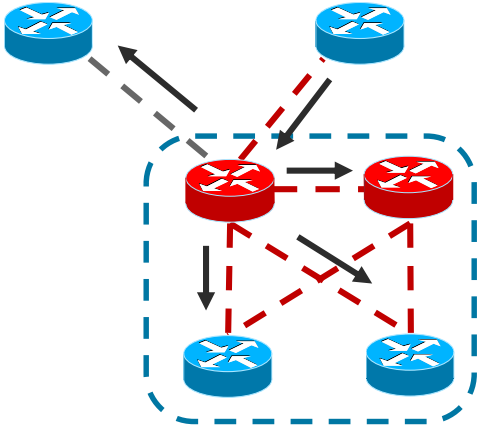
RR sends prefix to clients and non-clients (and sends to other eBGP peers)

prefix coming from RR client



RR reflects prefix to clients and sends to non-clients (and sends to eBGP peers)

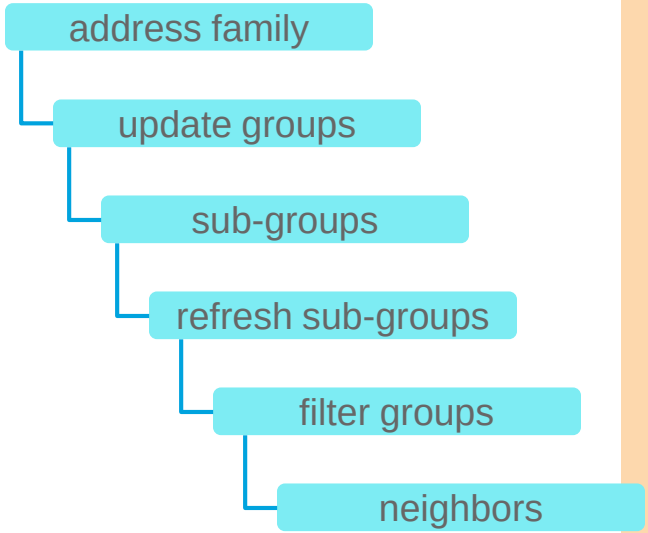
prefix coming from a non-client



RR reflects prefix to clients (and sends to eBGP peers)

Cisco live!

Update Groups in IOS XR

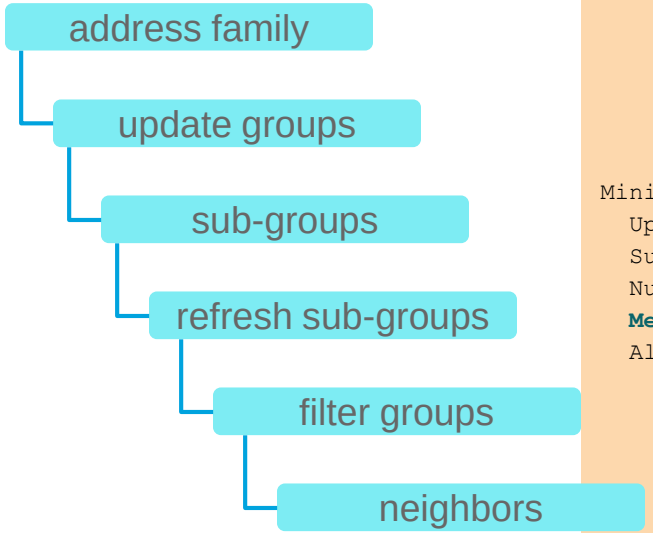


"show bgp ... replication"
in IOS equivalent

```
RP/0/6/CPU0:router#show bgp vpnv4 unicast update out update-group 0.2
VRF "default", Address-family "VPNv4 Unicast"

  Update-group 0.2
    Flags: 0x0010418b
    Sub-groups: 1 (0 throttled)
    Refresh sub-groups: 0 (0 throttled)
    Filter-groups: 3
    Neighbors: 3 (0 leaving)
    Update OutQ:          0 bytes (0 messages)
    Update generation recovery pending ?  [No]

    Last update timer start:   Apr  3 08:44:21.425
    Last update timer stop:    ---
    Last update timer expiry:  Apr  3 08:44:21.435 (1w4d ago)
    Update timer running ?  [No] (0.000 sec remaining; last started for 0.010 sec)

    History:
      Update OutQ Hi:                    3600 bytes (5 messages)
      Update OutQ Cumulative:           38700 bytes (54 messages)
      Update OutQ Discarded:                0 bytes (0 messages)
      Update OutQ Cleared:                  0 bytes (0 messages)
      Last discarded from OutQ:  --- (never)
      Last cleared   from OutQ:  --- (never)
      Update generation throttled 0 times, last event --- (never)
      Update generation recovered 0 times, last event --- (never)
      Update generation mem alloc failed 0 times, last event --- (never)
```

Update Groups in IOS XR

IOS-XR




```
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
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
```

We’re ready. Are you?