Advances in BGP

BRKIPM-3005

Steven Moore

James Ng

HOUSEKEEPING

- We value your feedback, don’t forget to complete your online session evaluations after each session and complete the Overall Conference Evaluation which will be available online from Friday.
- Visit the World of Solutions on Level -01!
- Please remember this is a ‘No Smoking’ venue!
- Please switch off your mobile phones!
- Please remember to wear your badge at all times including the Party!
- Do you have a question? Feel free to ask them during the Q&A section or write your question on the Question form given to you and hand it to the Room Monitor when you see them holding up the Q&A sign.

Session Overview

- New Developments
  - Features completed
  - Features in progress
  - Features on the roadmap
  - Features on the "whiteboard"
- IOS BGP will be the focus
- IOS-XR BGP basics will not be discussed, but are included for your reference in the handout (along with other additional information)
- Assumes you have a strong base knowledge of BGP
  - Attributes
  - Decision algorithm
  - 2547 VPNs
- Questions throughout the presentation are encouraged

Agenda

- Faster Convergence
  - BGP Scanner
  - NHT – Next Hop Tracking
  - FSD - Fast Session Deactivation
  - Event Driven Route Origination
  - MRAI – Min Route Advertisement Interval
  - TCP PMTU – Path MTU Discovery
  - Software Improvements
- BGP → TCP Enhancements
- 4-byte AS
- NSR – Non Stop Routing
- OER - Optimized Edge Routing
- Whiteboard Features

Faster Convergence

- Increased focus on faster BGP convergence
    - Critical for voice
    - VPN customers want IGP-like convergence
- Several factors influence BGP convergence
    - Detection of Change
    - Propagation of Information
    - Network Topology and Complexity
    - Network Stability

Faster Convergence

- Typically two scenarios where we need faster convergence
- Single route convergence
  - A bestpath change occurs for one prefix
  - How quickly can BGP propagate the change throughout the network?
  - How quickly can the entire BGP network converge?
  - Key for VPNs and voice networks
- Router startup or “clear ip bgp *” convergence
  - Most stressful scenario for BGP
  - CPU may be busy for several minutes
  - Limiting factor in terms of scalability
  - Key for any router with a full Internet table and many peers

Convergence Basics – BGP Scanner

- BGP Scanner plays a key role in convergence
- Full BGP table scan happens every 60 seconds
  - `bgp scan-time X`
  - Lowering this value is not recommended
- Full scan performs multiple housekeeping tasks
  - Validate nexthop reachability
  - Validate bestpath selection
  - Route redistribution and network statements
  - Conditional advertisement
  - Route dampening
  - BGP Database cleanup
- Import scanner runs once every 15 seconds
  - Imports VPNv4 routes into vrfs
  - `bgp scan-time import X`

Convergence Basics – BGP Nexthops

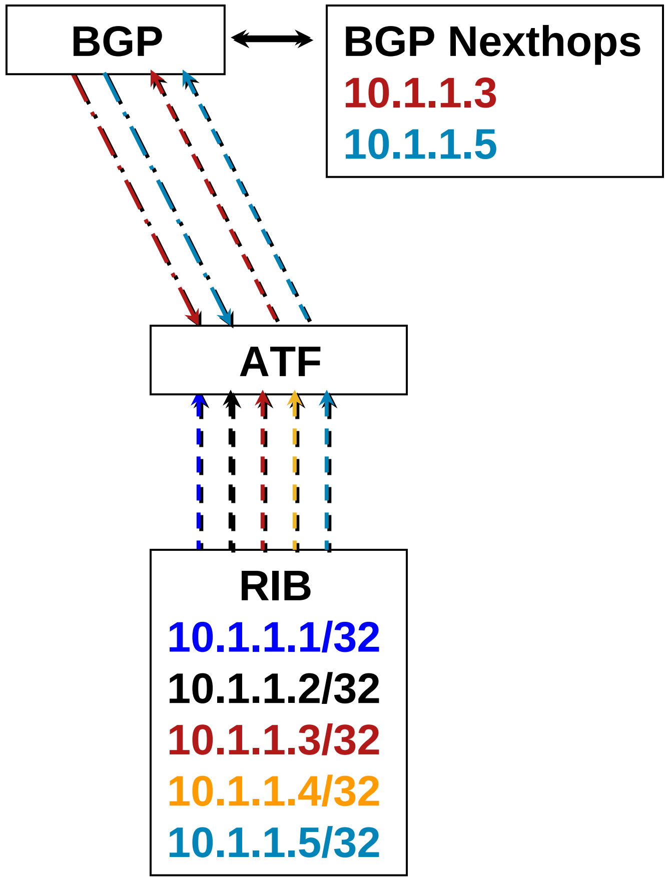
- Every 60 seconds the BGP scanner recalculates bestpath for all prefixes
- Changes to the IGP cost of a BGP nexthop will go unnoticed until scanner's next run
  - IGP may converge in less than a second
  - BGP may not react for as long as 60 seconds ☹
- Need to change from a polling model to an event driven model to improve convergence
  - Polling model – Check each BGP nexthop's IGP cost every 60 seconds
  - Event driven model – BGP is informed by a 3rd party when the IGP cost to a BGP nexthop changes

ATF – Address Tracking Filter

- ATF is a middle man between the RIB and RIB clients
  - BGP, OSPF, EIGRP, etc are all clients of the RIB
- A client tells ATF what prefixes it is interested in
- ATF tracks each prefix
  - Notify the client when the route to a registered prefix changes
  - Client is responsible for taking action based on ATF notification
  - Provides a scalable event driven model for dealing with RIB changes

ATF – Address Tracking Filter

- BGP tells ATF to let us know about any changes to 10.1.1.3 and 10.1.1.5
- ATF filters out any changes for 10.1.1.1/32, 10.1.1.2/32, and 10.1.1.4/32
- Changes to 10.1.1.3/32 and 10.1.1.5/32 are passed along to BGP



NHT – Next Hop Tracking

- BGP Next Hop Tracking
  - Enabled by default
  - 12.0(29)S, 12.3(14)T
  - `[no] bgp nexthop trigger enable`
- BGP registers all nexthops with ATF
  - Hidden command will let you see a list of nexthops
  - `show ip bgp attr nexthop`
- ATF will let BGP know when a route change occurs for a nexthop
- ATF notification will trigger a lightweight “BGP Scanner” run
  - Bestpaths will be calculated
  - None of the other “Full Scan” work will happen

NHT – Next Hop Tracking

- Once an ATF notification is received BGP waits 5 seconds before triggering NHT scan

    ```
    bgp nexthop trigger delay <0-100>
    ```

    May lower default value as we gain experience

- Event driven model allows BGP to react quickly to IGP changes

    No longer need to wait as long as 60 seconds for BGP to scan the table and recalculate bestpaths

    Tuning your IGP for fast convergence is recommended

NHT – Next Hop Tracking

- Dampening is used to reduce frequency of triggered scans
- `show ip bgp internal`

  Displays data on when the last NHT scan occurred

  Time until the next NHT may occur (dampening information)
- New commands

```
bgp nexthop trigger enable
bgp nexthop trigger delay <0-100>
show ip bgp attr next-hop ribfilter
debug ip bgp events nexthop
debug ip bgp rib-filter
```

- Full BGP scan still happens every 60 seconds

  Full scanner will no longer recalculate bestpaths if NHT is enabled

FSD – Fast Session Deactivation

- Register a peer's addresses with ATF
- ATF will let BGP know if there is a change in the route to reach the peer
- If we lose our route to the peer, tear down the session
  - No need to wait for the hold timer to expire!
- Ideal for multihop eBGP peers
- 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
  - Imagine if you lose your IGP route to your RR (Route Reflector) for just 100ms ☹
- Off by default

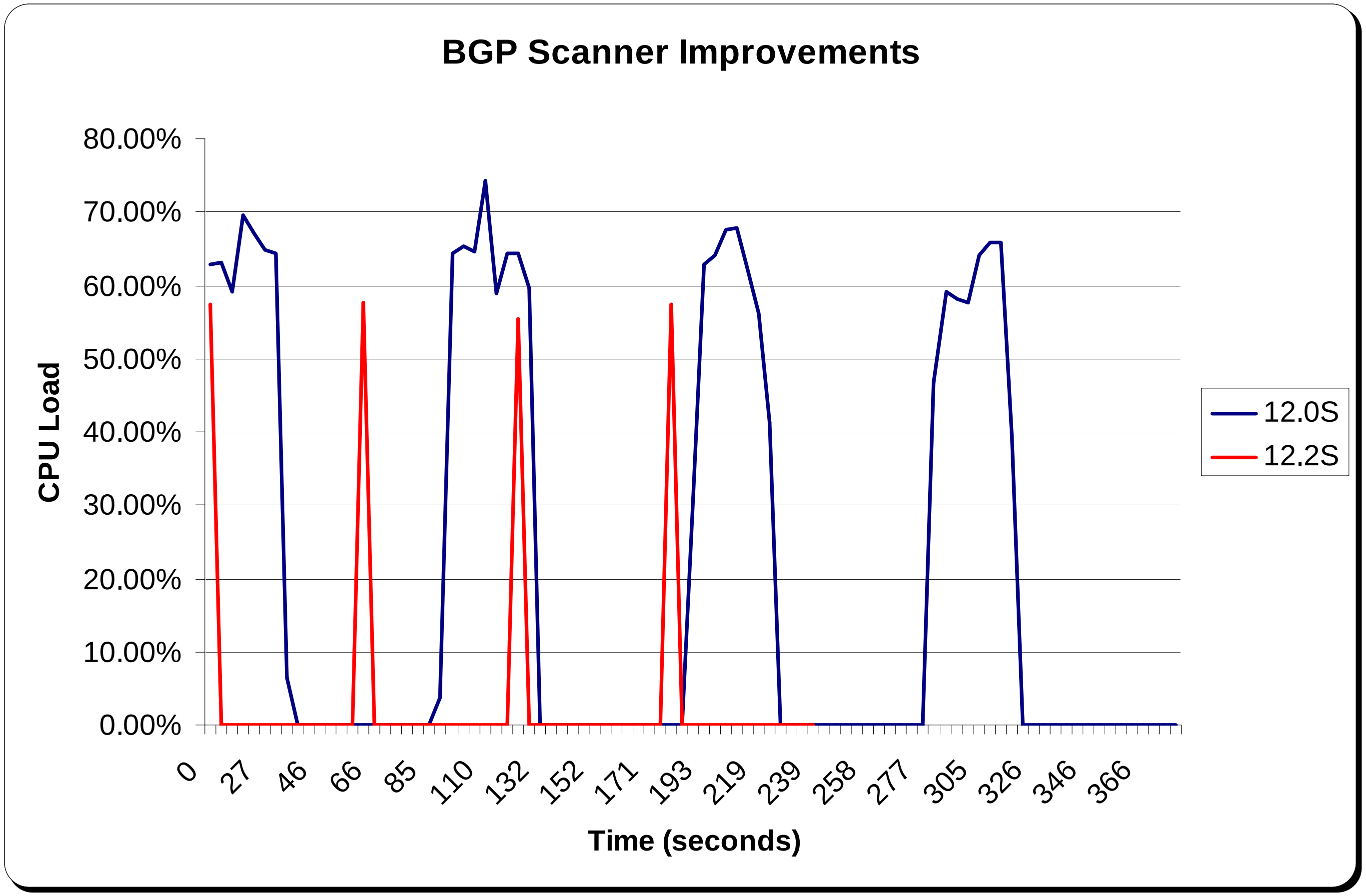
```
neighbor x.x.x.x fall-over
```

- Introduced in 12.0(29)S, 12.3(14)T

Event Driven Route Origination

- Route Origination was also based on a scanner dependant polling model
- Scanner traversed the RIB looking for routes that should be originated
- Traversing the RIB consumes a lot of CPU
- Route origination is now event driven
  - Scanner no longer checks the RIB for routes to redistribute
  - Route redistribution is event driven
  - Network statements are event driven
  - CPU impact of scanner is greatly reduced
- On by default, cannot disable
- Introduced in 12.2(28)S, 12.3(13)T via CSCef51906

Event Driven Route Origination



- 7200 with NPE-G1
- 900k routes in the BGP table
- BGP Scanner in 12.2S uses much less CPU

MRAI – Min Route Advertisement Interval

"…determines the minimum amount of time that must elapse between an advertisement and/or withdrawal of routes to a particular destination by a BGP speaker to a peer. This rate limiting procedure applies on a per-destination basis, although the value of MinRouteAdvertisementIntervalTimer is set on a per BGP peer basis."

RFC 4271

Section 9.2.1.1

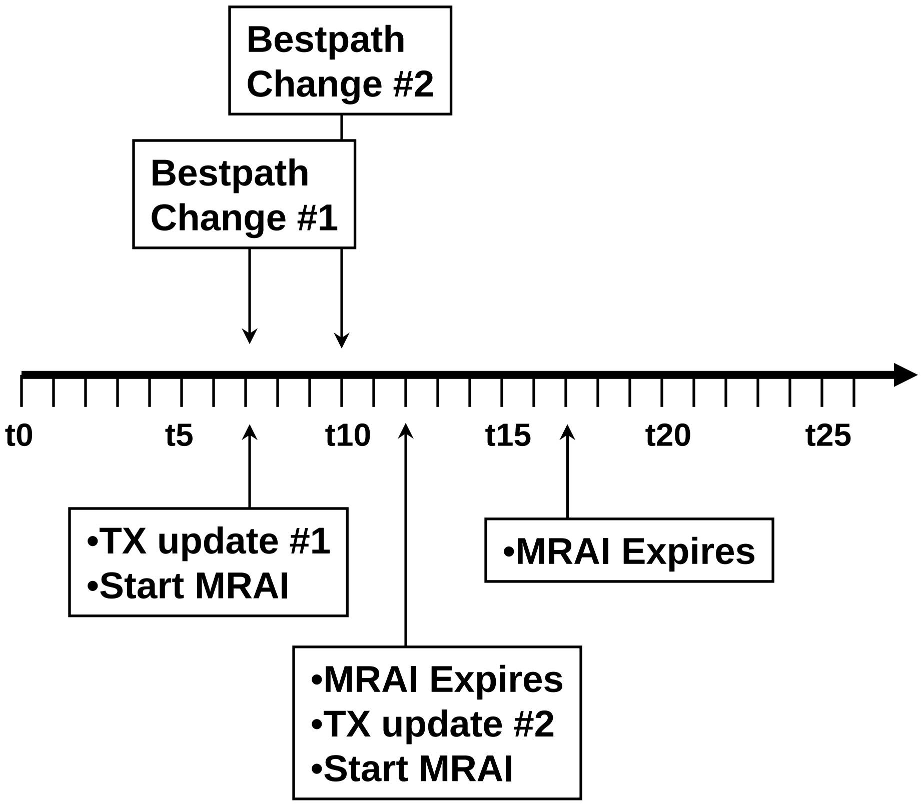
MRAI – Basics

- MRAI timers are maintained per peer
  - iBGP – 5 seconds by default
  - eBGP – 30 seconds by default
  - `neighbor x.x.x.x advertisement-interval <0-600>`
- Popular misconception that withdraws are not affected
- Pros
  - Promotes stability by batching route changes
  - Improves update packing in some situations
- Cons
  - May drastically slow convergence
  - Current defaults are too conservative
  - One flapping prefix can slow convergence for other prefixes

MRAI – Implementation

- How is the timer enforced for peer X?
  - Timer starts when all routes have been advertised to X
  - For the next MRAI (seconds) we will not propagate any bestpath changes to peer X
  - Once X's MRAI timer expires, send him updates and withdraws
  - Restart the timer and the process repeats…
- User may see a wave of updates & withdraws to peer X every MRAI
- User will NOT see a delay of MRAI between each individual update and/or withdraw
  - BGP would probably never converge if this was the case

MRAI – Implementation

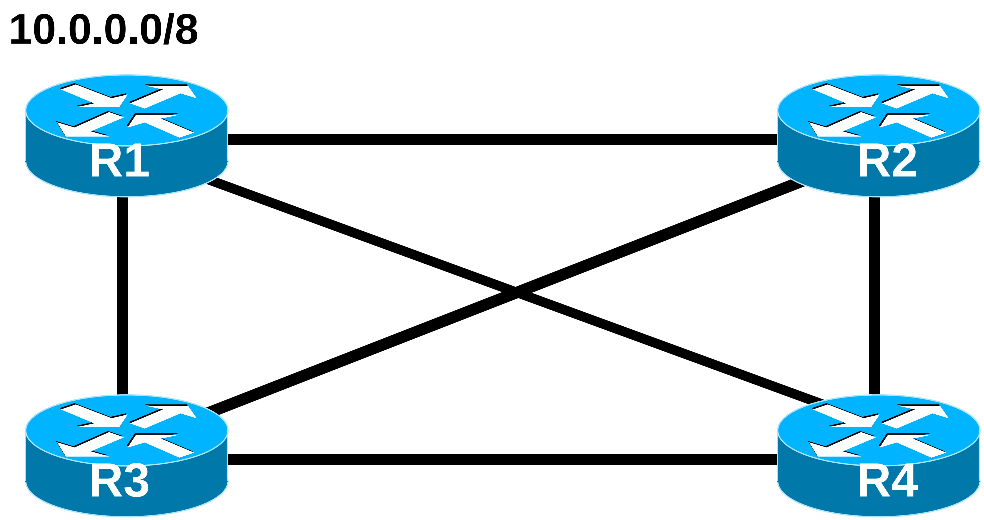


- MRAI timeline for iBGP peer
- Bestpath Change #1 at t7 is TXed immediately
- MRAI timer starts at t7, will expire at t12
- Bestpath Change #2 at t10 must wait until t12 for MRAI to expire
- Bestpath Change #2 is TXed at t12
- MRAI timer starts at t12, will expire at t17
- MRAI expires at t17…no updates are pending

MRAI – Slows Convergence

- BGP is not a link state protocol, but instead is path vector based
- May take several “rounds/cycles” of exchanging updates & withdraws for the network to converge
- MRAI must expire between each round!
- The more fully meshed the network and the more tiers of Autonomous Systems, the more rounds required for convergence
- Think about
  - The many tiers of Autonomous Systems that are in the Internet
  - The degree to which peering can be fully meshed

MRAI – Convergence Example



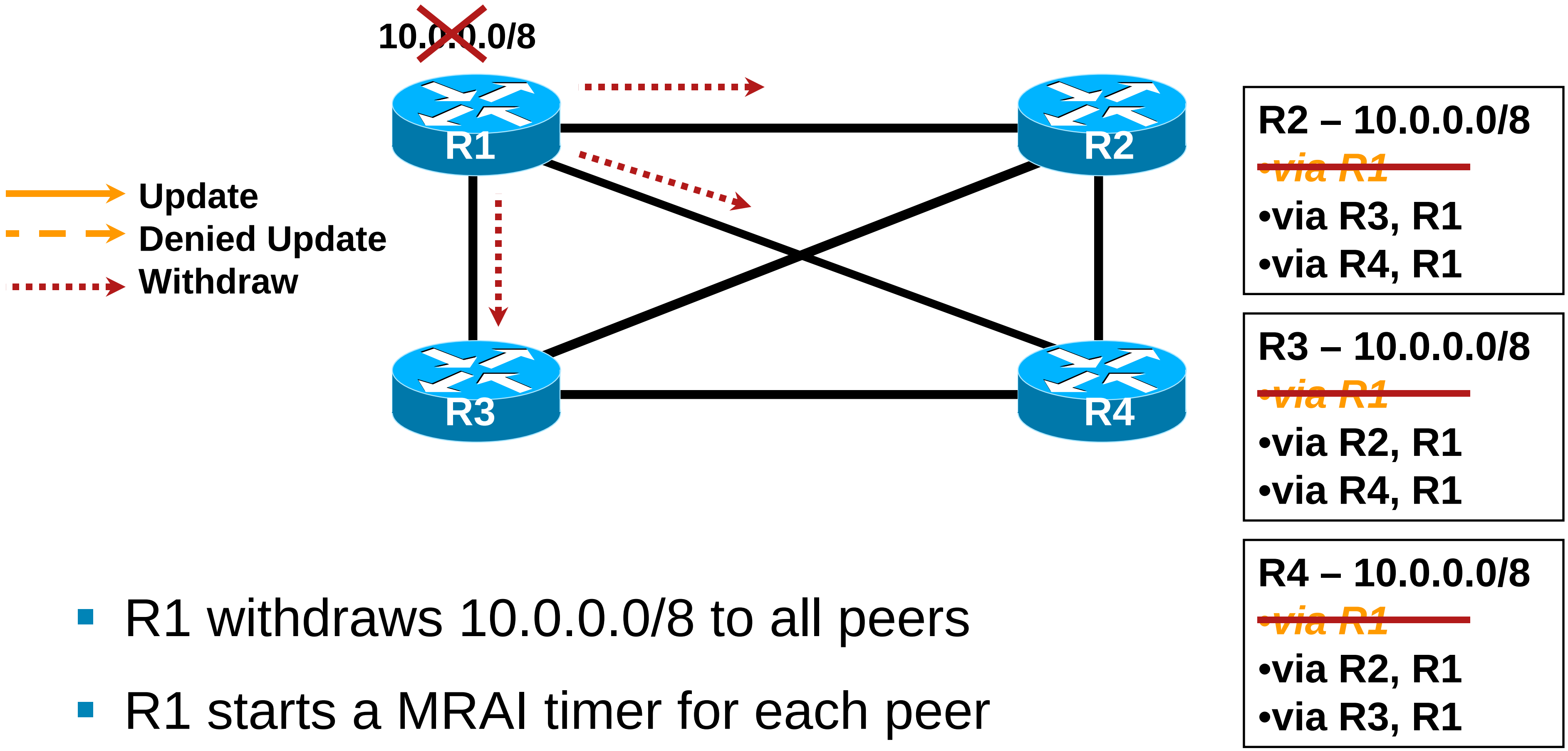
R2 – 10.0.0.0/8
•via R1
•via R3, R1
•via R4, R1

R3 – 10.0.0.0/8
•via R1
•via R2, R1
•via R4, R1

R4 – 10.0.0.0/8
•via R1
•via R2, R1
•via R3, R1

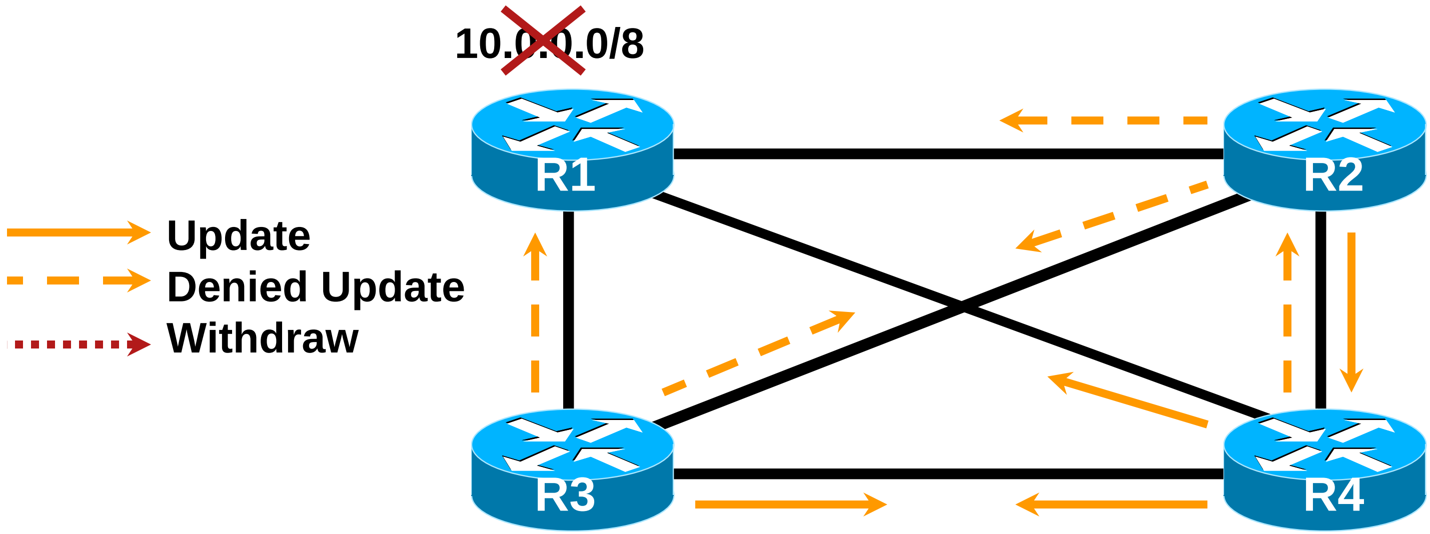
- Full mesh is the worst case MRAI convergence scenario
- R1 will send a withdraw to all peers for 10.0.0.0/8
- Count the number of rounds of UPDATEs & withdraws until the network has converged
- Note how MRAI slows convergence
- *Orange* path is the bestpath

MRAI – Convergence Example



- R1 withdraws 10.0.0.0/8 to all peers
- R1 starts a MRAI timer for each peer

MRAI – Convergence Example



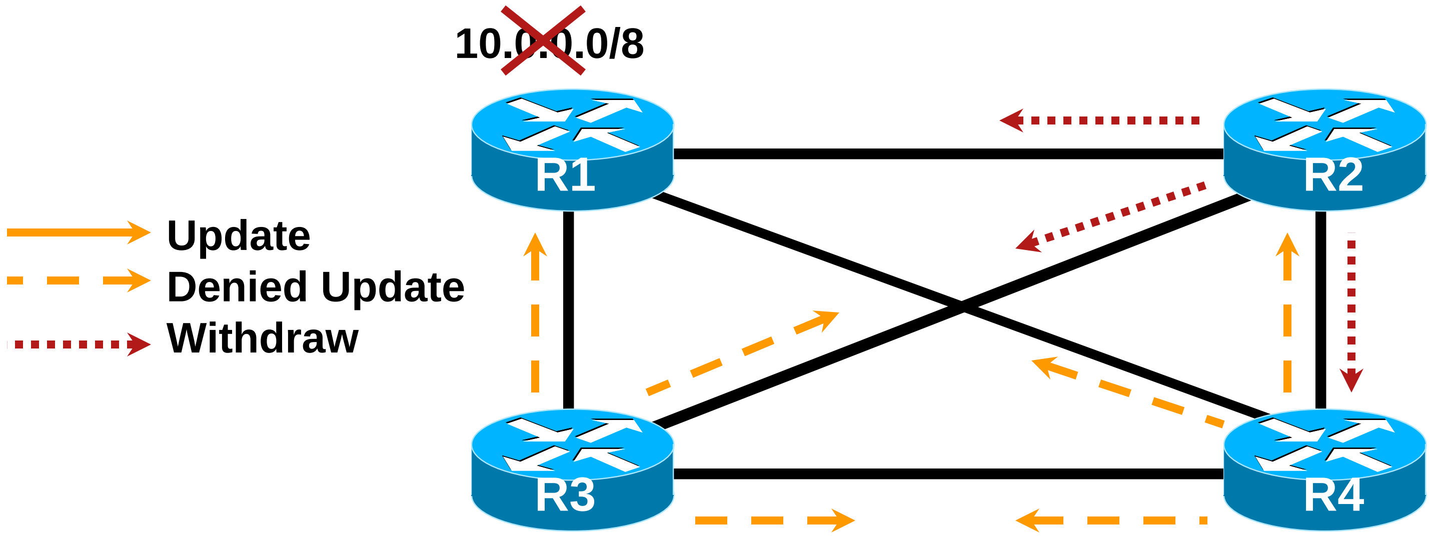
R2 – 10.0.0.0/8
~~•via R1~~
•via R3, R1
•via R4, R1

R3 – 10.0.0.0/8
~~•via R1~~
•via R2, R1
•via R4, R1

R4 – 10.0.0.0/8
~~•via R1~~
•via R2, R1
•via R3, R1

- R2, R3, & R4 recalculate their bestpaths
- R2, R3, & R4 send updates based on new bestpaths
- R2, R3, & R4 start a MRAI timer for each peer
- End of Round 1

MRAI – Convergence Example



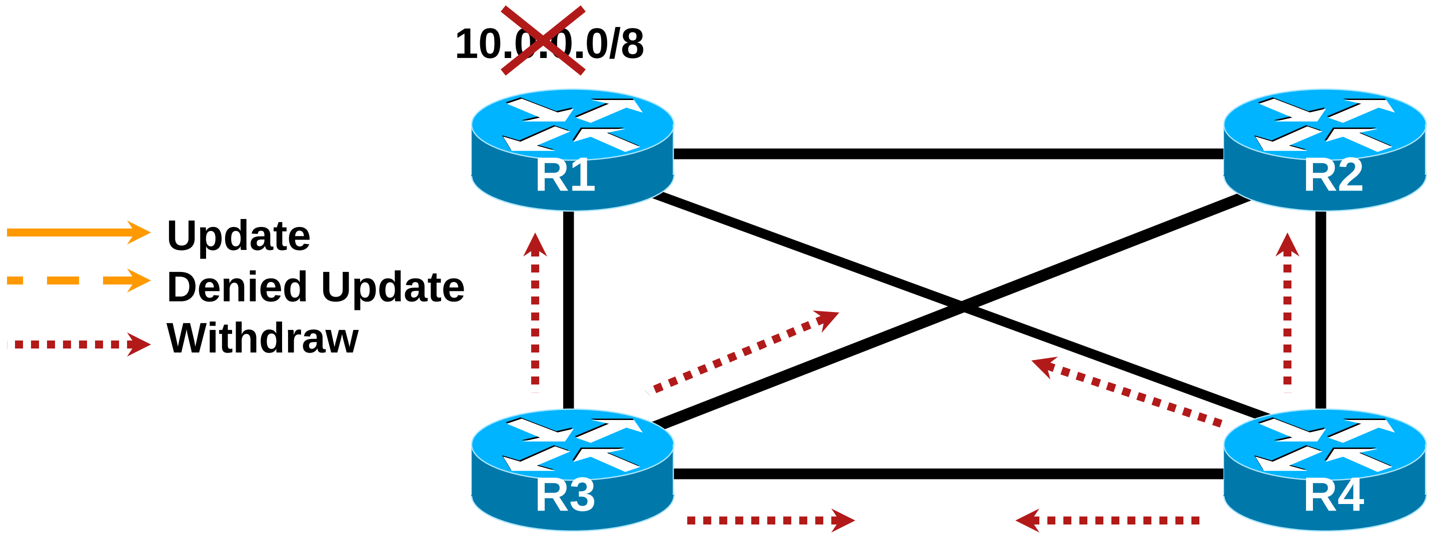
R2 – 10.0.0.0/8
~~•via R1~~
~~•via R3, R1~~
~~•via R4, R1~~

R3 – 10.0.0.0/8
~~•via R1~~
~~•via R2, R1~~
•via R4, R2, R1

R4 – 10.0.0.0/8
~~•via R1~~
~~•via R2, R1~~
•via R3, R2, R1
•via R2, R3, R1

- R2, R3, & R4 recalculate their bestpaths
- R2, R3 & R4 must wait for their MRAI timers to expire!
- R2, R3, & R4 send updates and withdraws based on their new bestpaths
- R2, R3, & R4 restart the MRAI timer for each peer
- End of Round 2

MRAI – Convergence Example



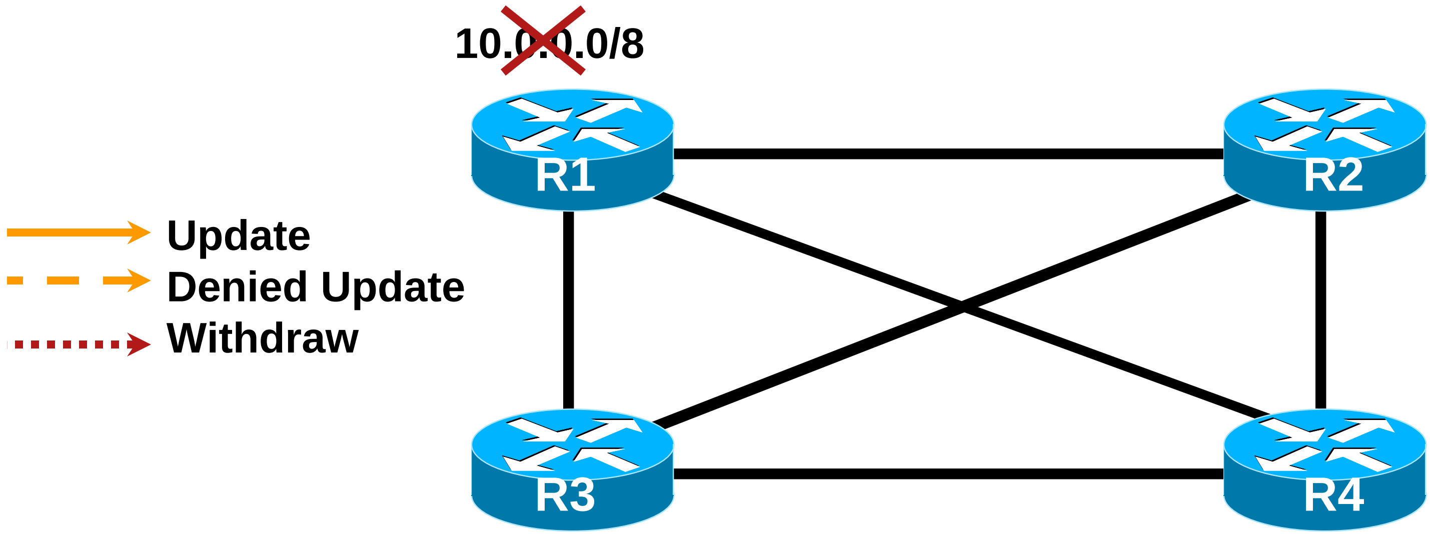
R2 – 10.0.0.0/8
- ~~via R1~~
- ~~via R3, R1~~
- ~~via R4, R1~~

R3 – 10.0.0.0/8
- ~~via R1~~
- ~~via R2, R1~~
- ~~via R4, R2, R1~~

R4 – 10.0.0.0/8
- ~~via R1~~
- ~~via R2, R1~~
- ~~via R3, R2, R1~~
- ~~via R2, R3, R1~~

- R3 & R4 recalculate their bestpaths
- R3 & R4 must wait for their MRAI timers to expire!
- R3 & R4 send updates and withdraws based on their new bestpaths
- R3 & R4 restart the MRAI timer for each peer
- End of Round 3

MRAI – Convergence Example



R2 – 10.0.0.0/8
- ~~via R1~~
- ~~via R3, R1~~
- ~~via R4, R1~~

R3 – 10.0.0.0/8
- ~~via R1~~
- ~~via R2, R1~~
- ~~via R4, R2, R1~~

R4 – 10.0.0.0/8
- ~~via R1~~
- ~~via R2, R1~~
- ~~via R3, R2, R1~~
- ~~via R2, R3, R1~~

- R2, R3, & R4 took 3 rounds of messages to converge
- MRAI timers had to expire between 1st/2nd round and between 2nd/3rd round
- Total MRAI convergence delay for this example

  iBGP mesh – 10 seconds

  eBGP mesh – 60 seconds

MRAI – Tuning

- Internet churn means we are constantly setting and waiting on MRAI timers
  - One flapping prefix slows convergence for all prefixes
  - Internet table sees roughly 1-2 bestpath changes per second
  - Based on Geoff Huston's research:
  - http://www.potaroo.net/presentations/2006-11-03-caida-wide.pdf
- For iBGP and PE→CE eBGP peers
  - `neighbor x.x.x.x advertisement-interval 0`
  - Will be the default in 12.0(32)S
- For regular eBGP peers
  - Lowering to 0 may get you dampened
  - OK to lower for eBGP peers if they are not using dampening

MRAI – Tuning

- Will a MRAI of 0 eliminate batching?
  - Somewhat but not much happens anyway
  - TCP, the operating system, and BGP code provide some batching
    - •Process all message from peer InQs
    - •Calculate bestpaths based on received messages
    - •Format UPDATEs to advertise new bestpaths
- What about CPU load from 0 second MRAI?
  - Internet table has ~1-2 bestpath changes per second
  - This number may differ for you, your mileage may vary.
  - Easy for a router under normal conditions to handle, 5 seconds of delay is not needed

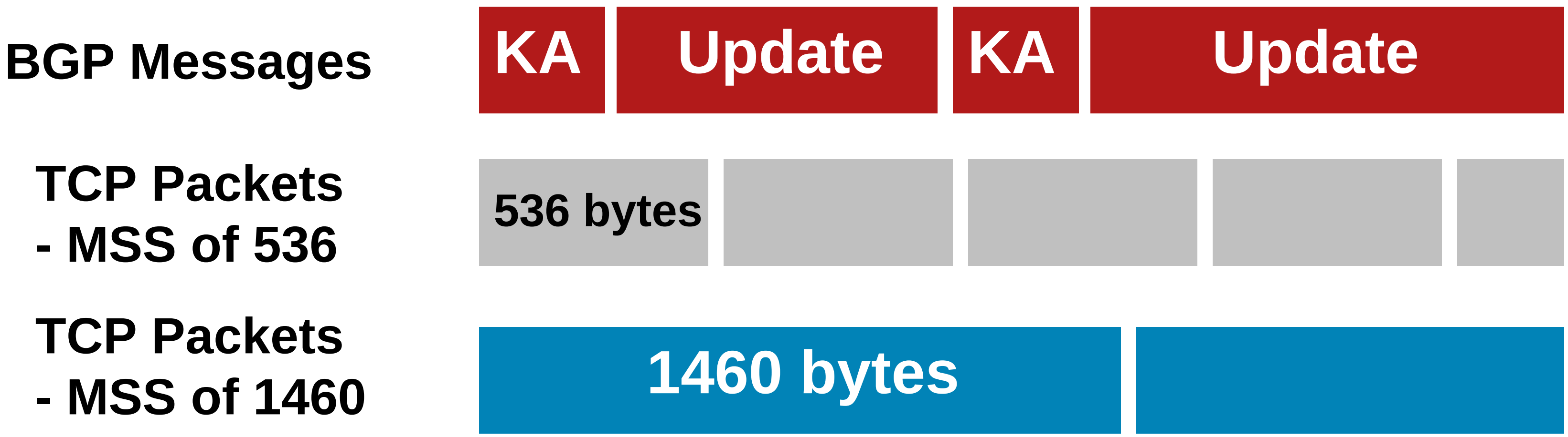
Scalability Update – Overview

- Bootup convergence and “clear ip bgp *” are the biggest challenges
  - Must converge all of our peers from scratch
  - BGP has to build and transmit a ton of data
- Multiple ways to improve bootup convergence and scalability
- Interface input queue drops
  - TCP acks can arrive in waves
  - Dropping a TCP ack is costly
  - If you are getting these drops, increase the size of your interface input queues
- TCP path-mtu-discovery
- Upgrade ☺

TCP Path MTU Discovery

- 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
- 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
- "`ip tcp path-mtu-discovery`"
  - MSS = Lowest MTU between destinations - IP overhead (20 bytes) – TCP overhead (20 bytes)
  - 1460 bytes for Ethernet network
  - 4430 bytes for POS network

TCP Path MTU Discovery



- BGP KAs (Keepalives) are 19 bytes
- BGP Updates vary in size up to 4096 bytes
- The larger the TCP MSS the fewer TCP segments required
- Fewer packets means less overhead and faster convergence
- New knob will allow you to enable/disable per peer

```
[no] neighbor x.x.x.x transport path-mtu-discovery
```

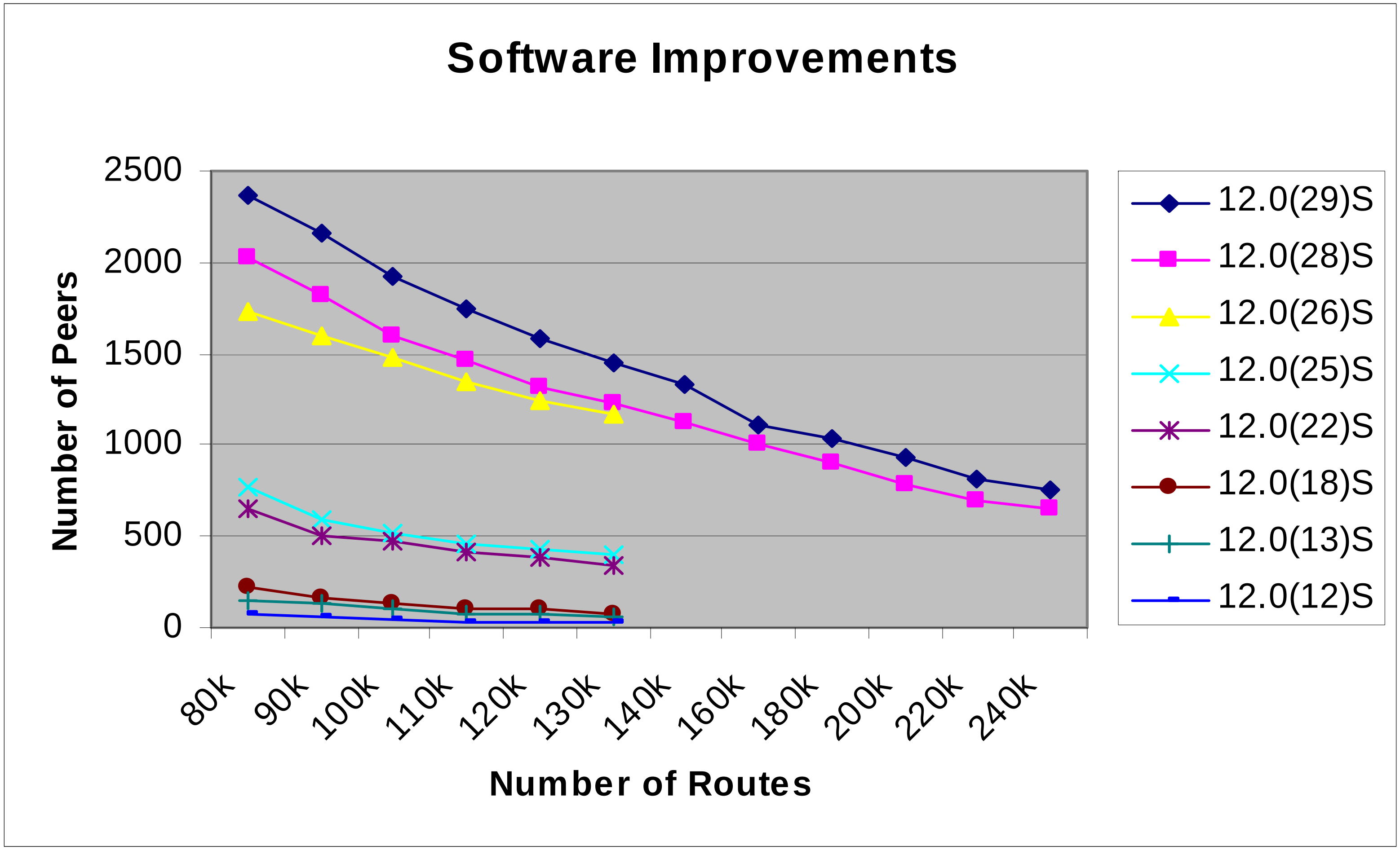

Scalability Update – Software

- Many incremental changes to BGP algorithms to improve convergence
- Most are related to building and replicating updates as efficiently as possible
- Some are related to reducing BGP transient memory usage
- Others involve improving BGP → TCP interaction

Scalability Update – Testing

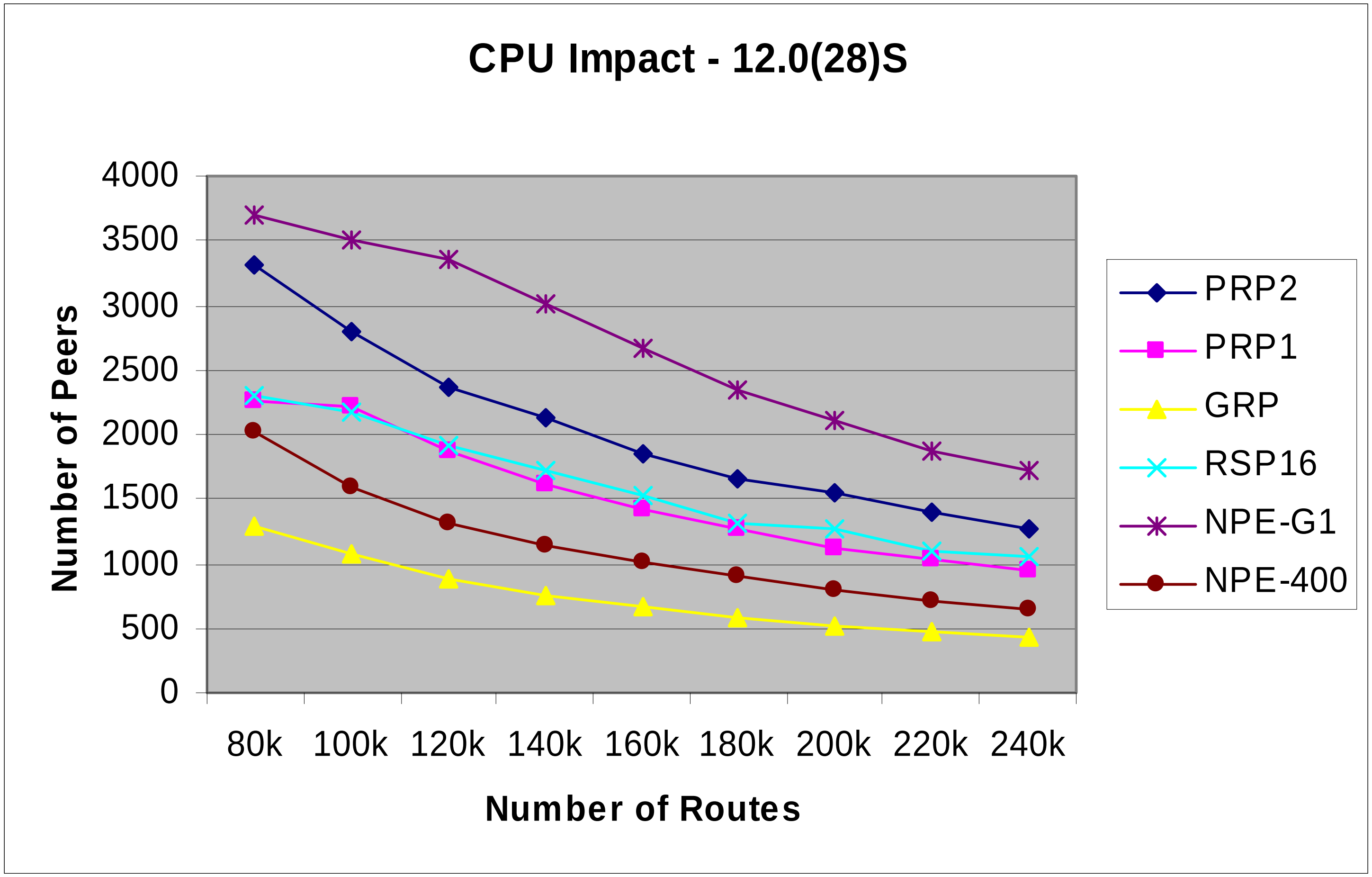
- "How Many Peers" test measures scalability and convergence
- X routes are accepted from one peer
- All routes are advertised to all peers
  - Vanilla BGP configuration, no inbound or outbound policies
  - All peers in one update-group
- If we can converge all peers within 10 minutes increase the number of peers and try again
  - Find the max number of peers for 80k routes, 90k routes, etc
  - Can compare scalability of one version of code vs. another
- "How many peers" graph
  - Displays the number of peers we can converge in 10 minutes (Y-axis) assuming we are advertising X-axis number of routes to each peer

Scalability Update – Software



- 7200 with NPE-400

Scalability Update - Hardware



- CPU processing power plays a big role

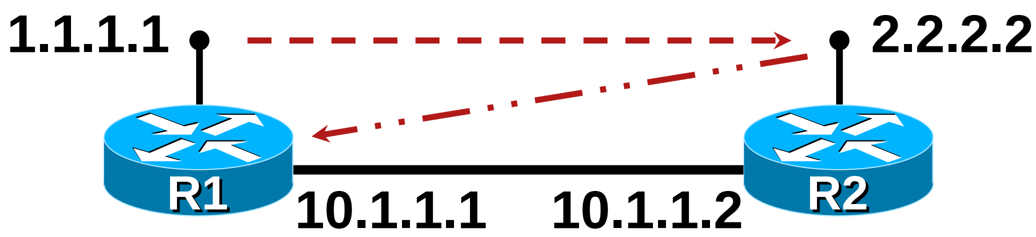
Agenda

- Faster Convergence
- BGP → TCP Enhancements
- 4-byte AS
- NSR – Non Stop Routing
- OER - Optimized Edge Routing
- Whiteboard Features

Source/Destination Address Matching

- Both peers must now agree on peering addresses
- Functionality introduced via CSCdp87864
- IP Addresses
  - Destination IP is specified via “`neighbor x.x.x.x`”
  - Source IP is outbound interface by default
  - Source IP may be specified via “`neighbor x.x.x.x update-source` *`interface`*”
- TCP port numbers
  - Destination will be port 179
  - Source port is random for added security

Source/Destination Address Matching



- Both sides must agree on source/destination addresses
- R1 to R2 connection

```
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source loopback 0
```

- R2 to R1 connection

```
neighbor 10.1.1.1 remote-as 100
neighbor 10.1.1.1 update-source loopback 0
```

- R1 and R2 do not agree on what addresses to use

  BGP will tear down the TCP session due to the conflict

  Points out configuration problems and adds some security

Source/Destination Address Matching

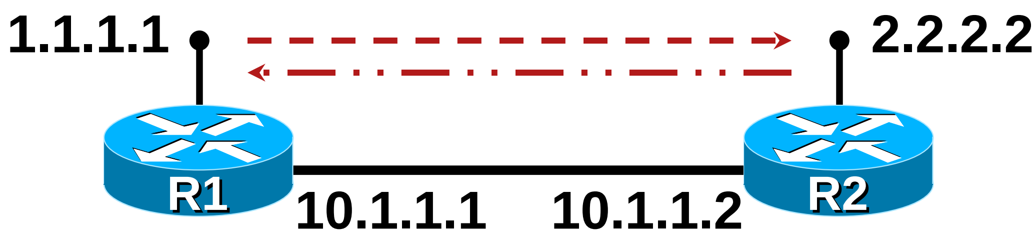
- R2 attempts to open a session to R1

```
BGP: 10.1.1.1 open active, local address 2.2.2.2
```

- R1 denies the session because of the address mismatch
- “debug ip bgp” on R1 shows

```
BGP: 2.2.2.2 passive open to 10.1.1.1
BGP: 2.2.2.2 passive open failed - 10.1.1.1 is not
update-source Loopback0's address (1.1.1.1)
```


Source/Destination Address Matching



- R1 to R2 connection

```
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source loopback 0
```

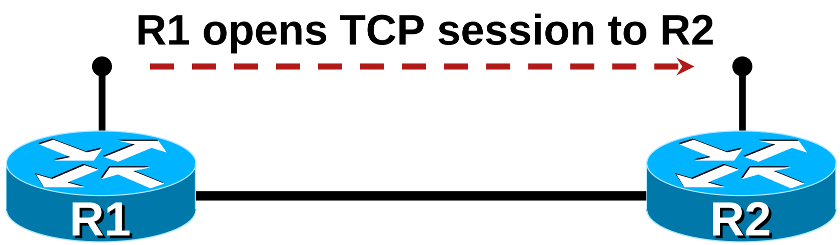
- R2 to R1 connection

```
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 update-source loopback 0
```

- Routers agree on source/destination address

  BGP will accept this TCP session

TCP – Active vs. Passive Session



- Active Session

  If the TCP session initiated by R1 is the one used between R1 & R2 then R1 “actively” established the session.

- Passive Session

  For the same scenario R2 “passively” established the session.

- R1 Actively opened the session
- R2 Passively accepted the session
- Can be configured

```
neighbor x.x.x.x transport connection-mode [active|passive]
```

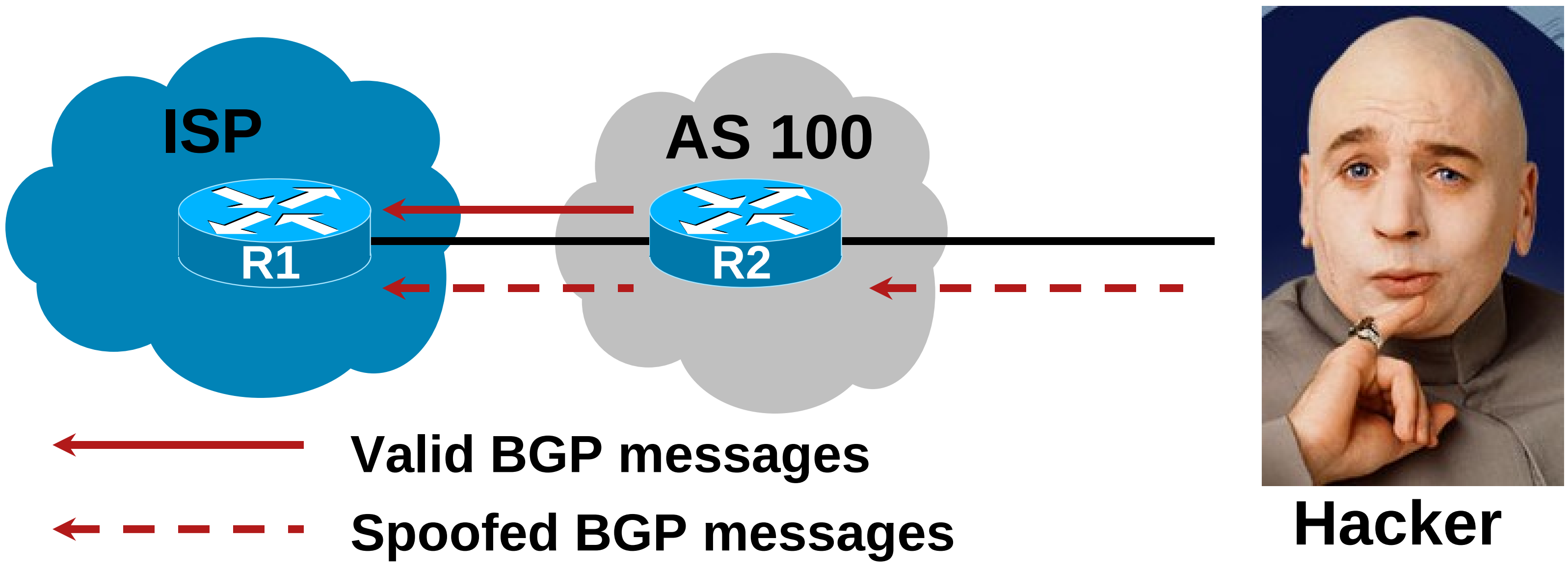

TCP – Active vs. Passive Session

- Use "`show ip bgp neighbor`" to determine if a router actively or passively established a session

```
R1#show ip bgp neighbors 2.2.2.2
BGP neighbor is 2.2.2.2,  remote AS 200, external link
  BGP version 4, remote router ID 2.2.2.2
[snip]
Local host: 1.1.1.1, Local port: 12343
Foreign host: 2.2.2.2, Foreign port: 179
```

- TCP open from R1 to R2's port 179 established the session
- Tells us that R1 actively established the session
  - If the foreign port is 179 then this router actively opened the session
  - If the local port is 179, then this router is the passive peer
- Explicitly Configuring BOTH ends of the session as Active or Passive will NOT work!

BTSH – BGP TTL Security Hack

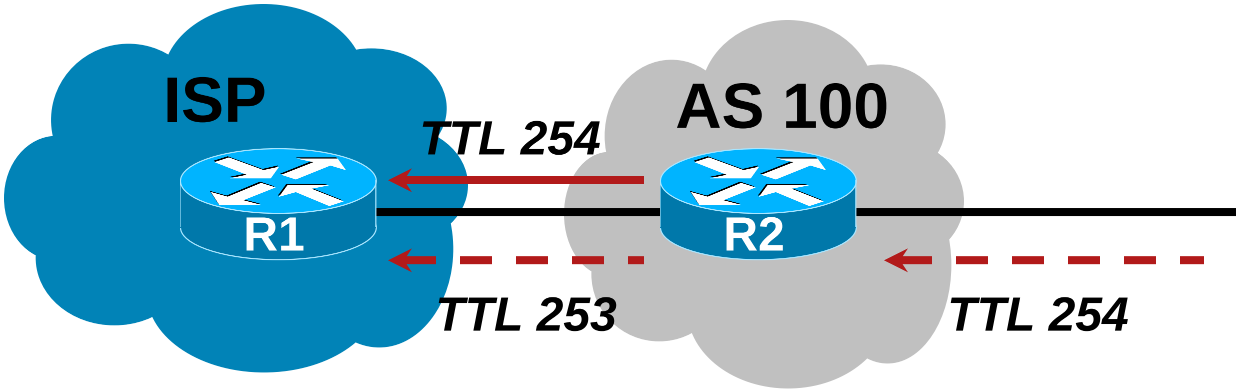


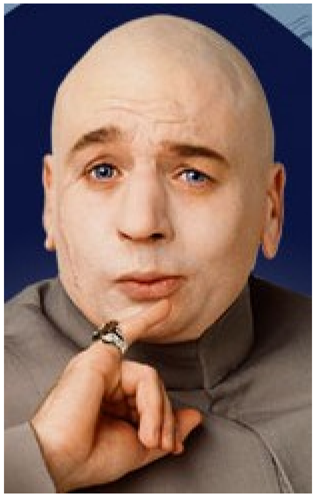
- Hacker spoofs BGP messages to R1 as if he is R2
- R1 must use MD5 to filter out the bogus messages
- MD5 validation must be done on the RP (Route Processor)
- Now known as GTSM - Generic TTL Security Mechanism

BTSH – BGP TTL Security Hack

- If AS 100 sets the TTL to 255 for all BGP messages
- ISP can check for a TTL of 254 for BGP messages from AS 100
- Provides a lightweight mechanism to defend against most BGP spoof attacks
- Does not not prevent attack from the same segment or distance as the configured peer (TTL would be the same)
- Does NOT replace the need for MD5 authentication!
- Introduced in 12.0(24)S

BTSH – BGP TTL Security Hack





Hacker

- R1 and R2 both use BTSH
- Both sides must configure the feature

  `neighbor x.x.x.x ttl-security hops 1`

  Valid TTL = 255 - # hops
- Packets from R2 will have a TTL of 254
- Packets generated by the hacker will have a TTL less than 254

  Easy to compare the TTL value vs. the 254 threshold and discard spoofed packets

  Possible to discard packets at the linecard

Agenda

- Faster Convergence
- BGP → TCP Enhancements
- 4-byte AS
- NSR – Non Stop Routing
- OER - Optimized Edge Routing
- Whiteboard Features

4-byte AS

- RFC 4271 defines an AS number as 2-bytes
- Private AS Numbers = 64512 → 65535
- Public AS Numbers = 1 → 64511
  - 39000+ have already been allocated
  - We will eventually run out of AS numbers
- Need to expand AS size from 2-bytes to 4-bytes
  - 4,294,967,295 AS numbers
- Cannot have a "flag day" solution, for example:
  - On Jan 1, 2010 all BGP speakers must support feature X
- Solution must allow for a gradual deployment
- ARIN assigning 4-byte AS upon request after Jan 1, 2007

4-byte AS

- draft-ietf-idr-as4bytes-12.txt
  - “BGP Support for Four-octet AS Number Space”
  - Provides 4-byte AS support in an incremental and backward compatible manner
  - Autonomous System numbers will be assigned in X.Y syntax
- X.Y notation
  - AS #65,536,005 is a mouthful
  - Split the 4-byte value into two 2-byte values
  - 0000001111101000 | 0000000000000101
  - 0000001111101000 = 1000
  - 0000000000000101 = 5
  - 1000.5 is easier to work with

4-byte AS

- 4-byte AS support is advertised within BGP capability negotiation
  - Speakers who support 4-byte AS are known as NEW speakers
  - Those who do not are known as OLD speakers
- New Reserved AS#
  - AS_TRANS = AS #23456
  - 2-byte placeholder for a 4-byte AS number
  - Used for backward compatibility with OLD speakers
- Two new attributes, both are “optional transitive”
  - NEW_AGGREGATOR
  - NEW_ASPATH

4-byte AS

From the perspective of a NEW speaker…

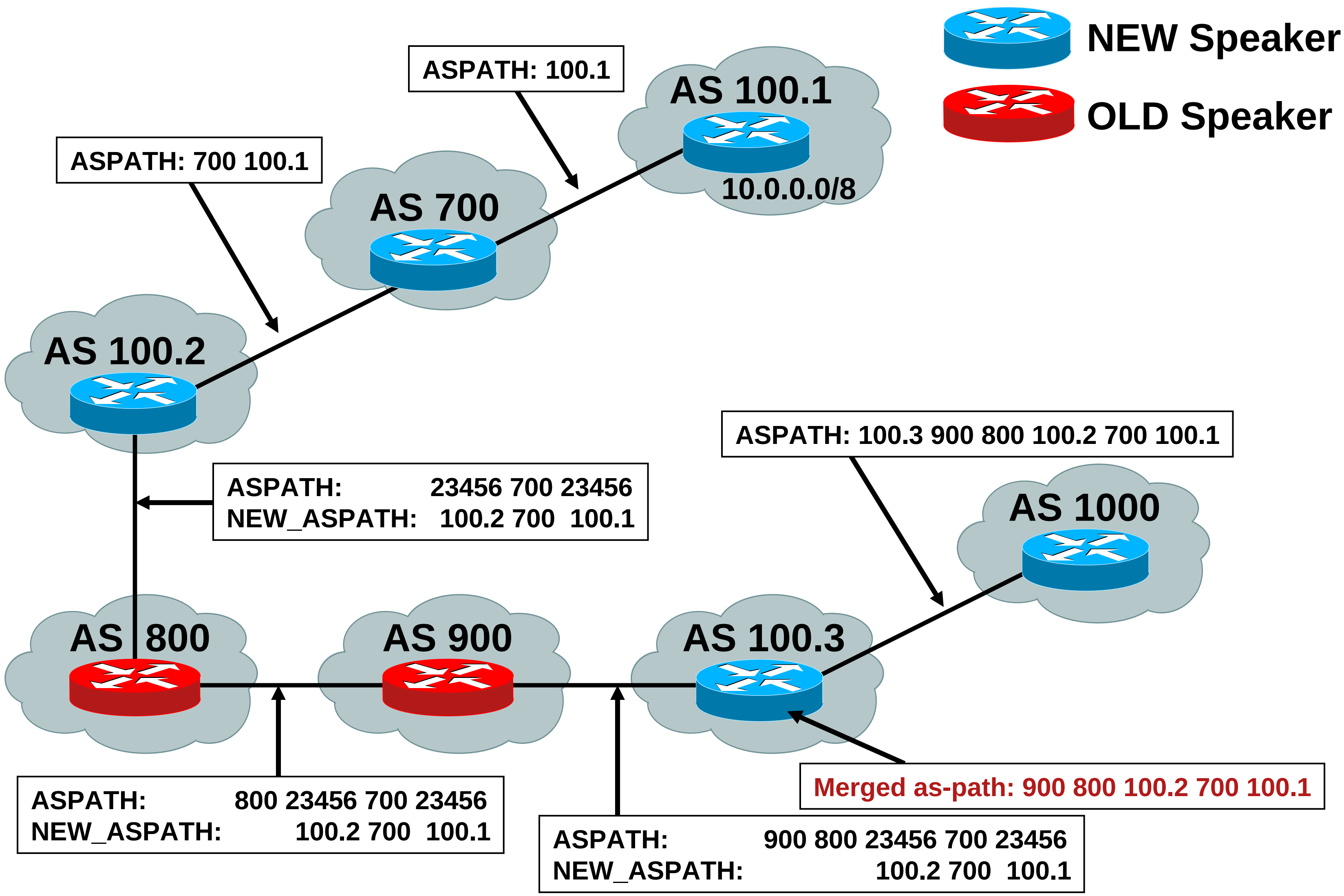
- When Formatting UPDATEs to another NEW speaker
  - Encode each AS number in 4-bytes
  - AS_PATH and AGGREGATOR are the relevant fields for BESTPATH
  - We should not see NEW_ASPATH and NEW_AGGREGATOR
- When Formatting UPDATEs to an OLD speaker
  - If the AGGREGATOR/ASPATH does not contain a 4-byte AS we are fine
  - If it does, substitute AS_TRANS (AS #23456) for each 4-byte AS
  - NEW_AGGREGATOR or NEW_ASPATH will contain a 4-byte encoded copy of the attribute if needed
  - OLD speaker will blindly pass along NEW_AGGREGATOR and NEW_ASPATH attributes

4-byte AS

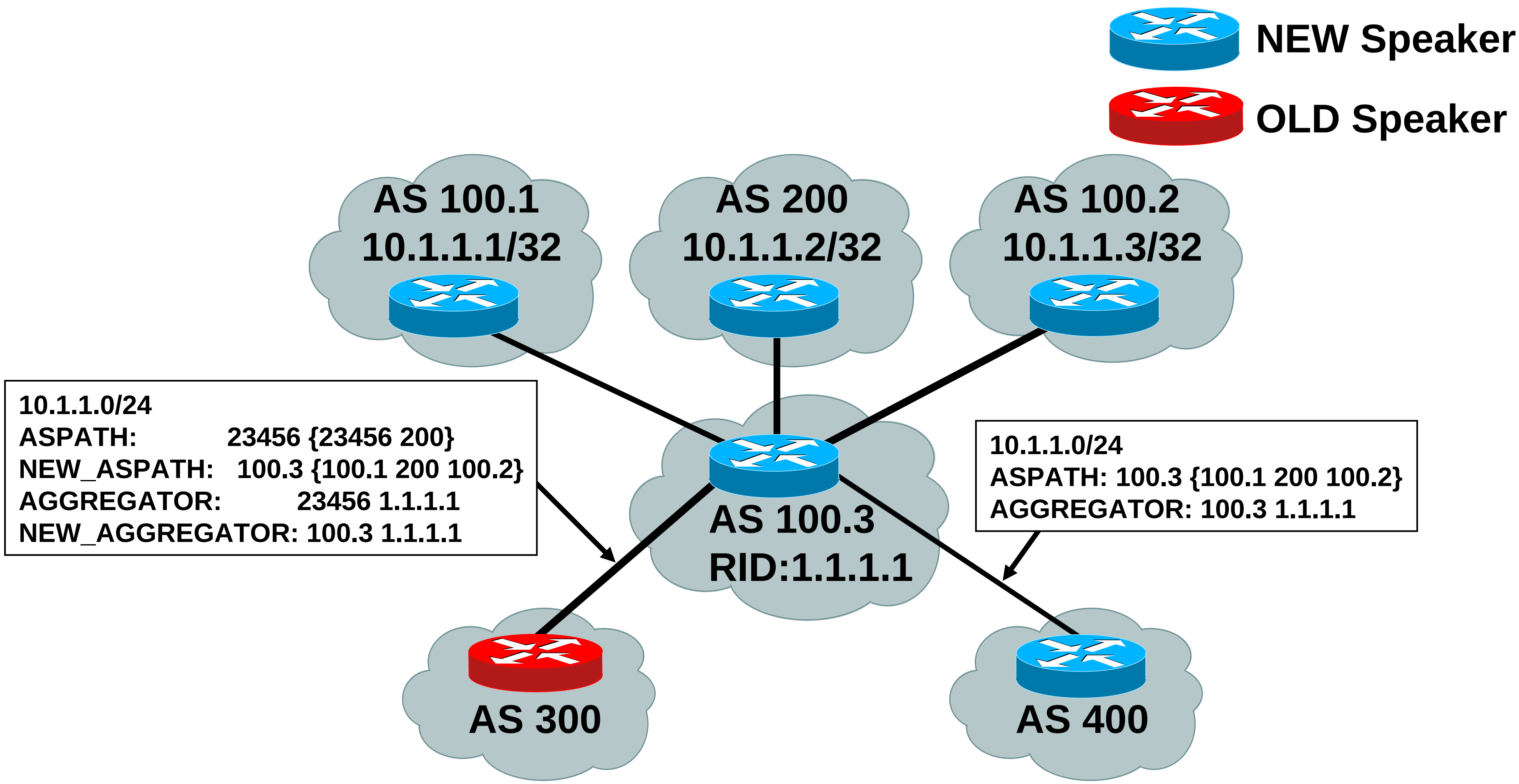
From the perspective of a NEW speaker…

- When Receiving UPDATEs from a NEW speaker
  - Decode each AS number as 4-bytes
  - AS_PATH and AGGREGATOR are encoded as 4-bytes ASN
- When Receiving UPDATEs from an OLD speaker
  - NEW_AGGREGATOR will override AGGREGATOR
  - NEW_ASPATH and ASPATH must be merged to form the correct as-path
- Merging NEW_ASPATH and ASPATH
  - ASPATH – `275 250 225` *`23456 23456`* `200` *`23456`* `175`
  - NEW_ASPATH – *`100.1 100.2`* `200` *`100.3`* `175`
  - Merged as-path – `275 250 225` *`100.1 100.2`* `200` *`100.3`* `175`

4-byte AS – ASPATH & NEW_ASPATH in a mixed environment



4-byte AS – Aggregation



- AS 100.3 creates 10.1.1.0/24 aggregate

Agenda

- Faster Convergence
- BGP → TCP Enhancements
- 4-byte AS
- NSR – Non Stop Routing
- OER - Optimized Edge Routing
- Whiteboard Features

NSR – Non Stop Routing

- NSR and NSF (Non Stop Forwarding) are not the same
- NSF in a nutshell
  - Provides forwarding during Active RP failover to Standby RP
  - BGP protocol changes required to recover from failover
  - Peers X & Y must be NSF aware for NSF to work
  - Should not be a challenge within an AS
  - PE → CE is a problem
    - Upgrading CE's is a huge deployment challenge

NSR – Non Stop Routing

- NSR in a nutshell

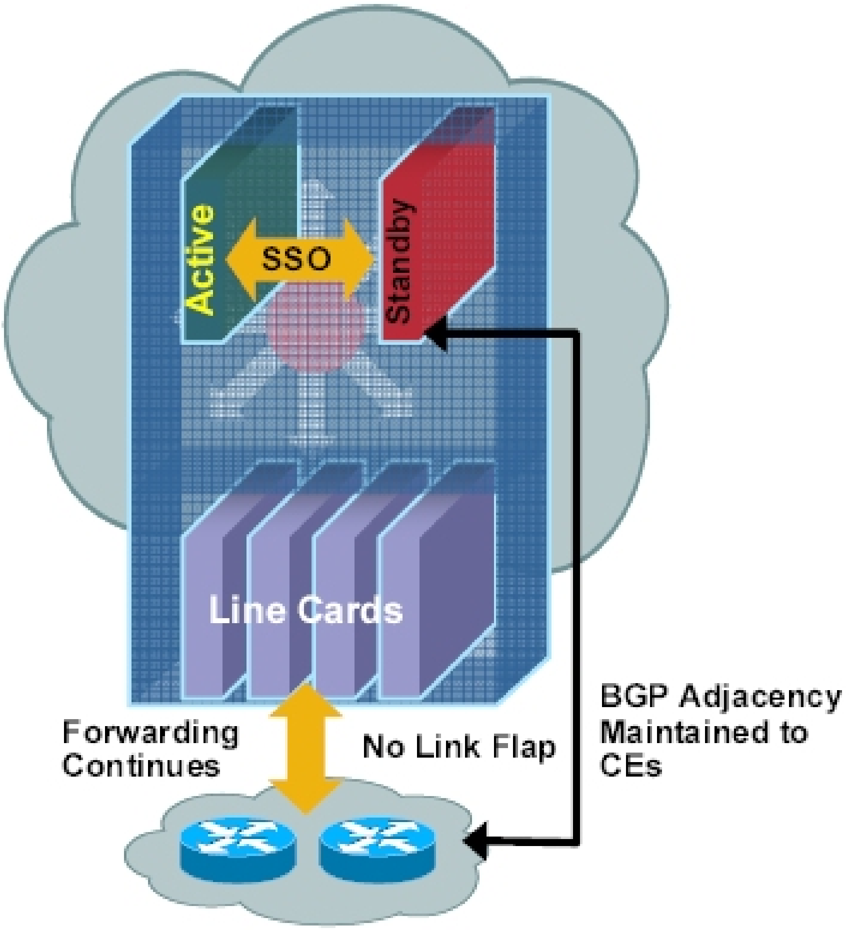
  Provides forwarding and preserves routing during Active RP failover to Standby RP

  BGP protocol changes *ARE NOT* required to recover from failover

  BGP peers' TCP sessions are maintained

  CE's do not need to be upgraded!

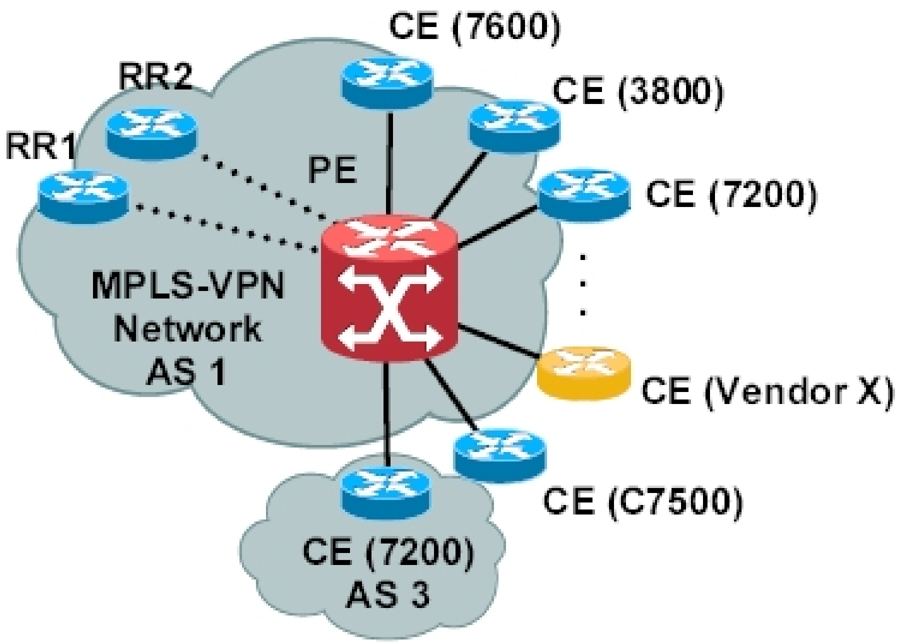
- Introduced in 12.2(28)SB



NSR – Non Stop Routing

- Simplified deployment for service providers
  - Only PEs need to be upgraded to support NSR (incremental deployment on per-peer basis)
  - CEs are not touched! (i.e., no software upgrade required)
- Scaling optimizations & recommendations
  - PE uses NSR with CEs that are not NSF-aware
  - PE uses NSF (Graceful Re-Start) with NSF-aware CEs
  - iBGP sessions to RRs use NSF (Graceful Re-Start)

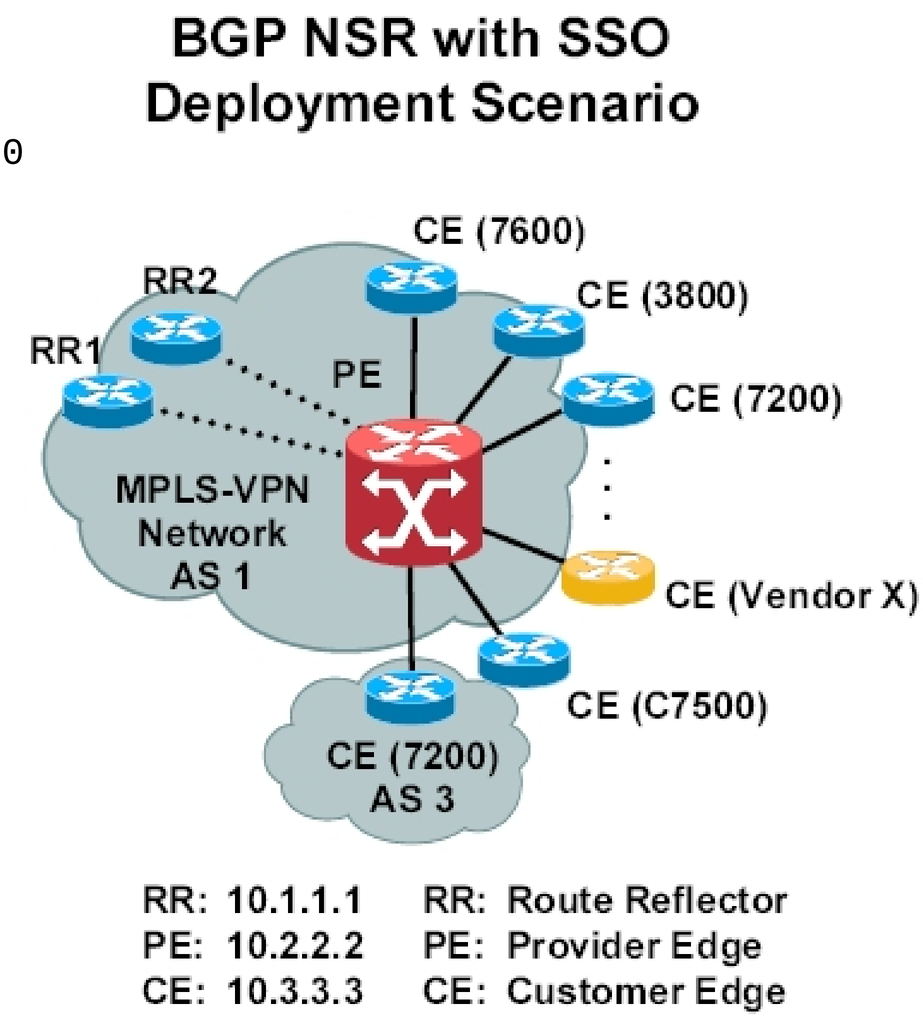
PE Focused Deployment Scenario



RR: 10.1.1.1 RR: Route Reflector
PE: 10.2.2.2 PE: Provider Edge
CE: 10.3.3.3 CE: Customer Edge

NSR – PE Configuration

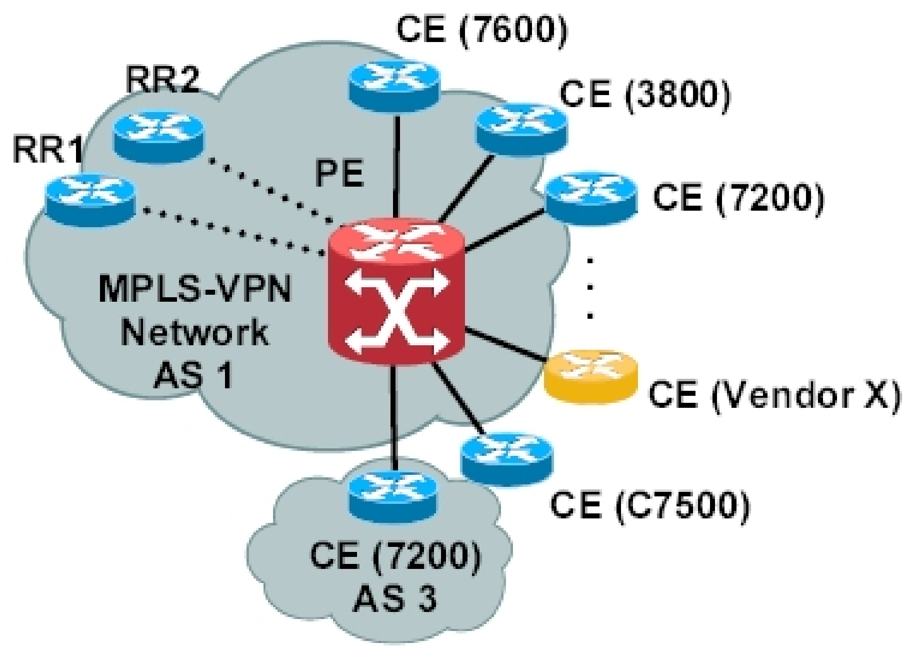
```
router bgp 1
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.1.1.1 remote-as 1
!
 address-family vpnv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf Customer1
  neighbor 10.3.3.3 remote-as 3
  neighbor 10.3.3.3 ha-mode sso
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 as-override
 exit-address-family
 !
```



NSR – CE Configuration

```
router bgp 3
 neighbor 10.2.2.2 remote-as 1
!
```

BGP NSR with SSO Deployment Scenario

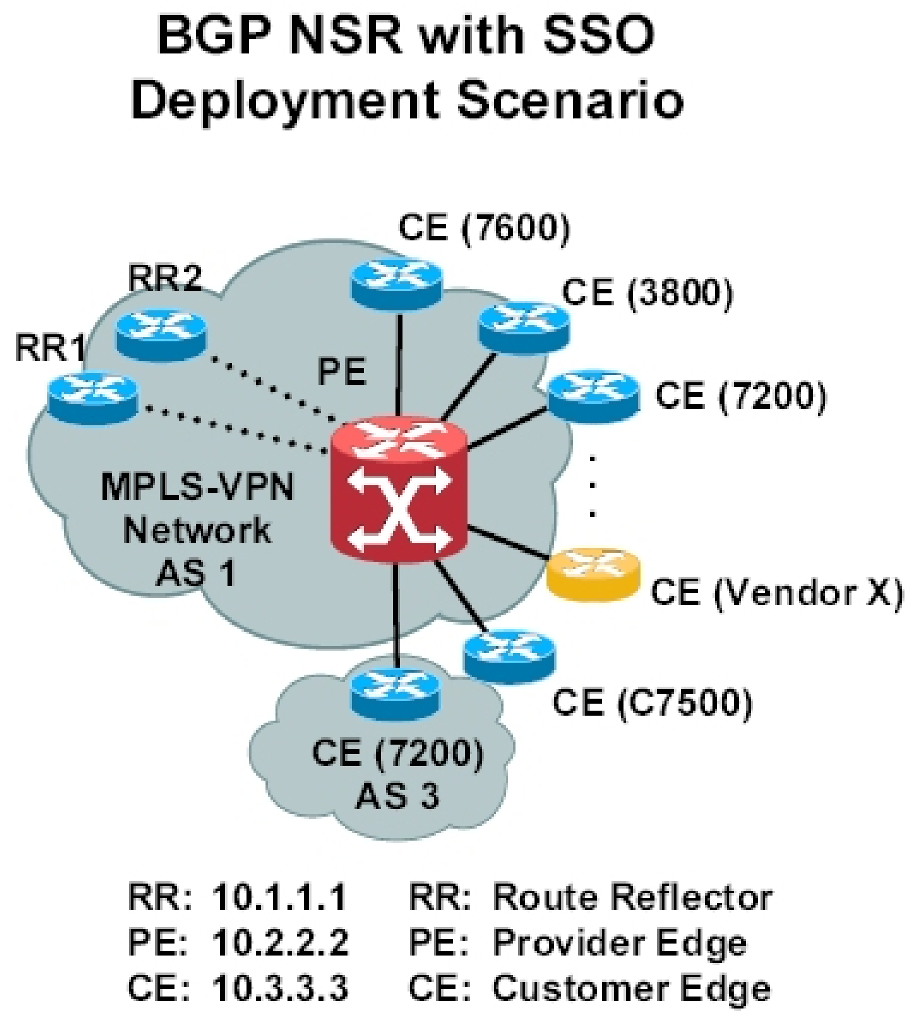


RR: 10.1.1.1 RR: Route Reflector
PE: 10.2.2.2 PE: Provider Edge
CE: 10.3.3.3 CE: Customer Edge

NOTE: No special BGP code or configuration (i.e., NSF-awareness) needed on the CE side to take advantage of the Non Stop Routing capabilities of the PE

NSR – RR Configuration with Graceful Restart

```
router bgp 1
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.2.2.2 remote-as 1
 !
 address-family vpnv4
  neighbor 10.2.2.2 activate
  neighbor 10.2.2.2 route-reflector-clien
  neighbor 10.2.2.2 send-community both
 exit-address-family
!
```



NSR – Related Commands

- `neighbor x.x.x.x ha-mode sso`
- This command is used to configure the router to support SSO towards this BGP neighbor
  - SSO is not enabled by default
  - Configurable via peer, peer-group, and session template

NSR – Related Commands

- `debug ip bgp sso {events | transactions} [detail]`

  Events: Display BGP SSO events

  Transactions: Displays debugging information for BGP speaker interactions between the active RP and the standby RP

  Detail: Displays detailed debugging information

- `debug ip tcp ha {events | transactions} [detail]`

  Events: Display TCP SSO events

  Transactions: Displays debugging information for TCP stack interactions between the active RP and standby RP

  Detail: Displays detailed debugging information

NSR – Related Commands

- `show ip bgp vpnv4 all sso summary`
- Used to display the number of BGP neighbors that are configured for Cisco BGP NSR

```
Router# show ip bgp vpnv4 all sso summary
    Stateful switchover support enabled for 40 neighbors
```


NSR – Related Commands

- `show tcp ha connections`
- Displays connection ID to TCP mapping data

```
Router# show tcp ha connections
SSO enabled for 40 connections
TCP      Local Address Foreign Address (state) Conn Id
71EACE60 2.0.56.1.179  2.0.56.3.58671  ESTAB   37
71EA9320 2.0.53.1.179  2.0.53.3.58659  ESTAB   34
71EA35F8 2.0.41.1.179  2.0.41.3.58650  ESTAB   22
[snip]
```

- Used for Debugging and Troubleshooting the stateful TCP sessions

Agenda

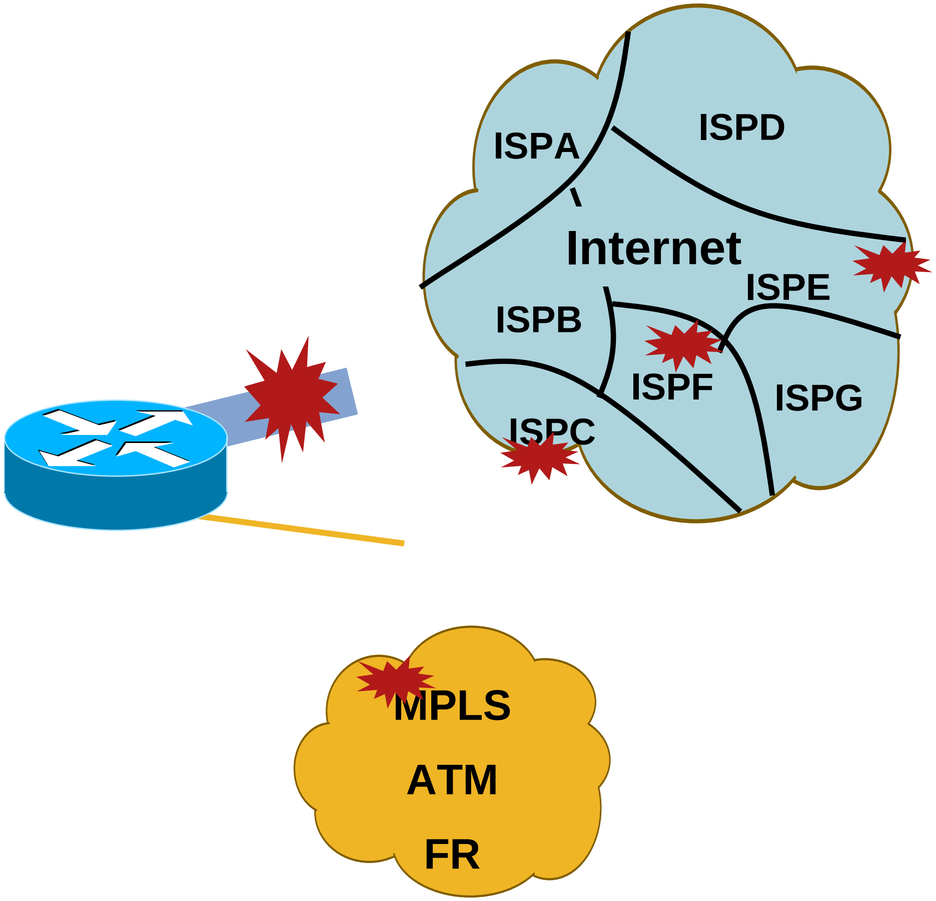
- Faster Convergence
- BGP → TCP Enhancements
- 4-byte AS
- NSR – Non Stop Routing
- OER - Optimized Edge Routing
  - Overview
  - CLI
  - Debug/Show commands
- Whiteboard Features

OER - Overview of Problem Set

"The network is up but are applications working?"

- WAN availability
  - Routing indicates reachability, but:
    - Blackouts
    - Brownouts
    - Congestion
- WAN performance
  - Bestpath not performance based
- Load distribution
  - Over/under utilized links
- Cost management
  - Need to control/limit transport cost

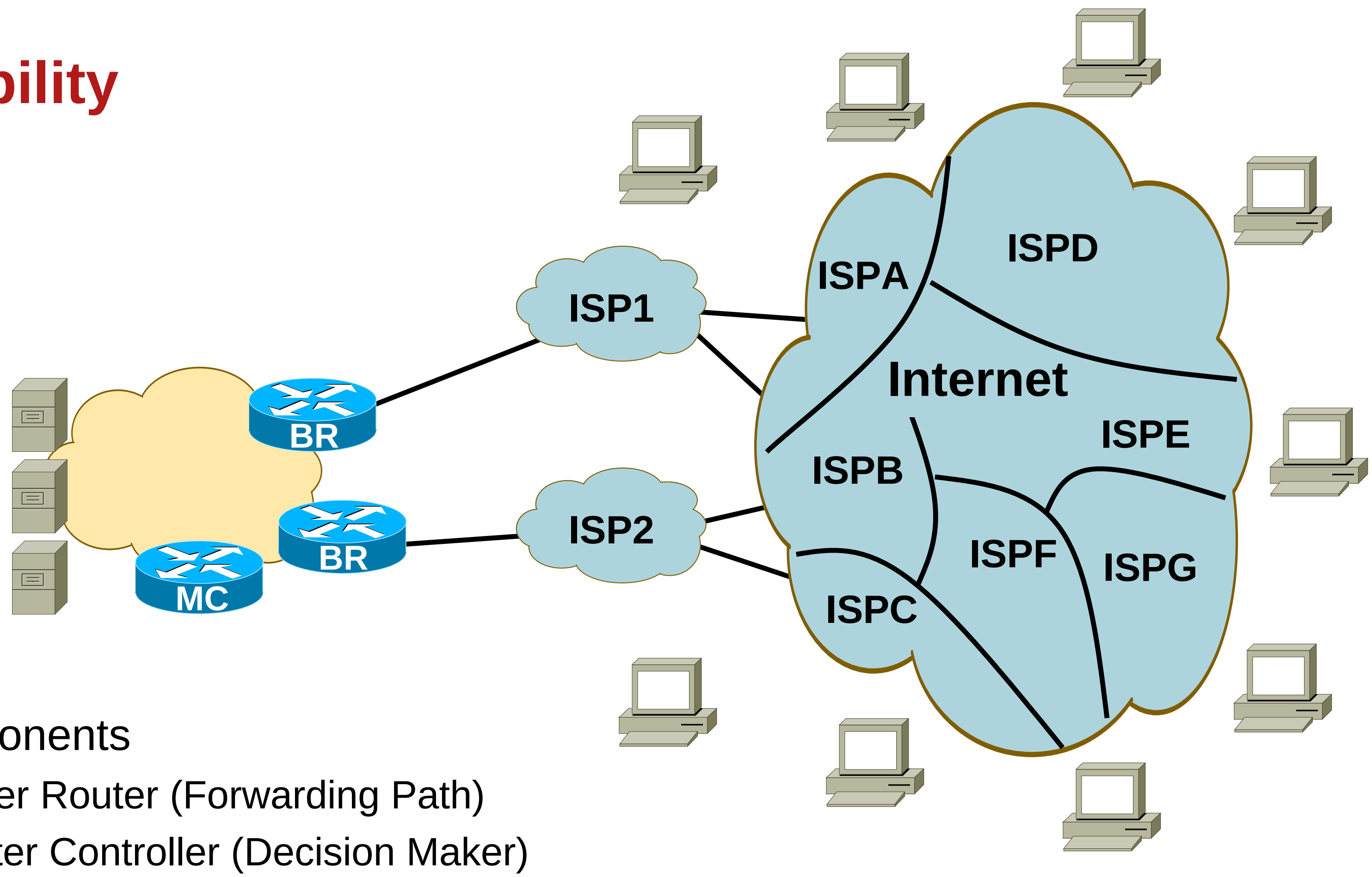
$$$$$$$



Optimized Edge Routing (OER)
Performance Based Routing for Internet Edge

Exit and/or Entrance Selection Criteria

Reachability
Delay
Loss
Jitter
MOS
Load
Cost



OER Components

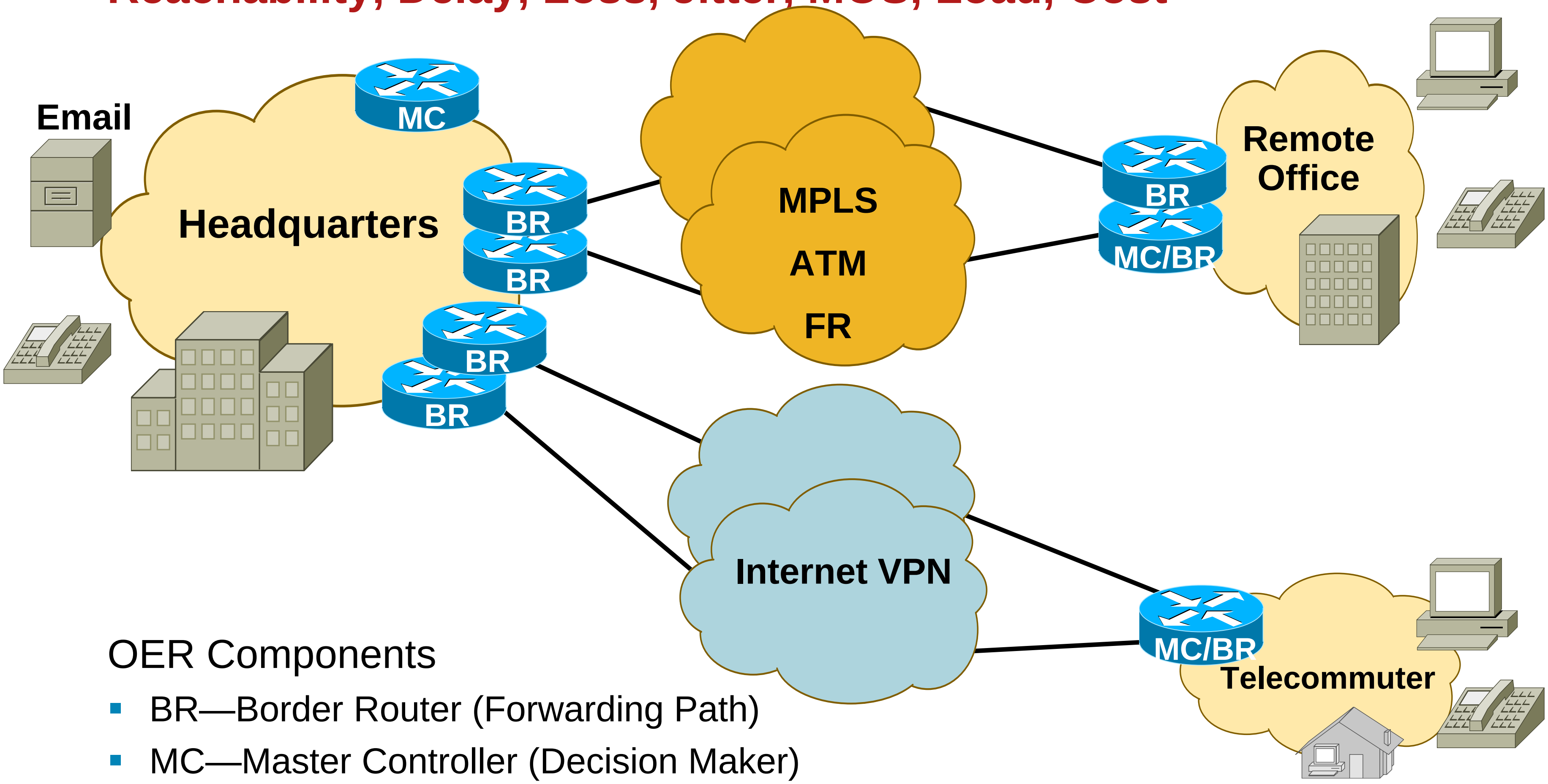
- BR—Border Router (Forwarding Path)
- MC—Master Controller (Decision Maker)

Optimized Edge Routing (OER)

Performance Based Routing for Enterprise WAN Edge

Exit Selection Criteria

Reachability, Delay, Loss, Jitter, MOS, Load, Cost

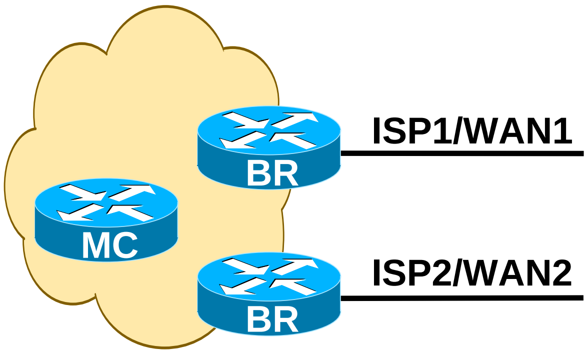


OER Components

- BR—Border Router (Forwarding Path)
- MC—Master Controller (Decision Maker)

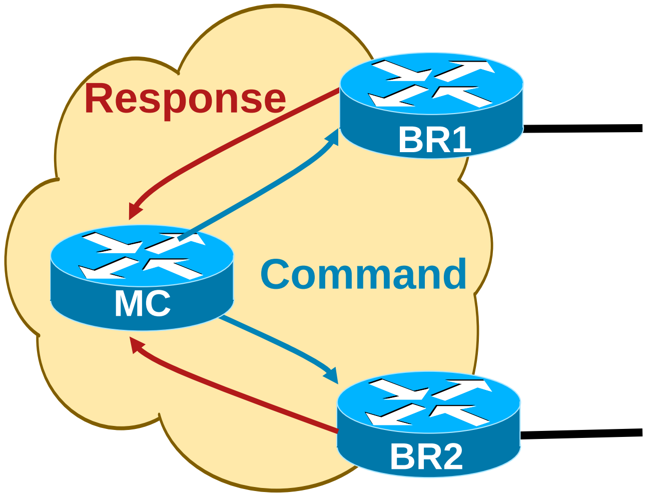
Component Description

- Master Controller (MC)
  - Cisco IOS® software feature
  - Apply Policy, Verification, Reporting
  - Standalone or collocated with BR
  - No routing protocol required
  - No packet forwarding/inspection required
- Border Router (BR)
  - Cisco IOS software feature in forwarding router
  - Learn, Measure, Enforcement
  - Netflow Collector
  - Probe Source (IP SLA Client)



Information Flow

- MC controls all operation
  - Issues commands to BRs
  - Contains traffic class/link data
  - Reports events
  - Reports measurements
  - Makes Policy decisions
- BR responds to MC commands
  - Sends responses to MC
  - Uses Netflow, IP SLA, BGP, static, RIB, …
  - Measures traffic class performance
  - Measures link performance
  - Enforces performance based routing



Which Applications to Manage?

- OER manages traffic classes
- Applications are translated to traffic classes
- Traffic class contains two objects

Destination Prefix (Required)
12.4(5)

- **w.x.y.z/n**
- **Default—0.0.0.0/0**
- **Host—w.x.y.z/32**

Application Identifier (Optional)
12.4T

- **DSCP**
- **Source Prefix**
- **Src Port Range**
- **Dst Port Range**

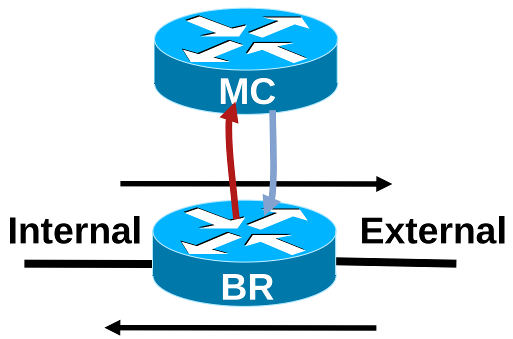
Entering Traffic Classes in the MC
Traffic Class Learning

- MC commands BRs to learn traffic classes
- BRs inspect all flows
- BRs ignore non-interesting flows
- BRs aggregate flows to prefix boundaries

BRs Know Traffic Classes

- BRs measure traffic class performance
- BRs sort traffic classes
- BRs send sorted traffic class lists to MC
    - BRs send host addresses used for probe targets
- MC combines and sorts to a single traffic class list
- MC enters top throughput and top delay into database

MC Knows Traffic Classes



Entering Traffic Classes in the MC

Traffic Class Learning

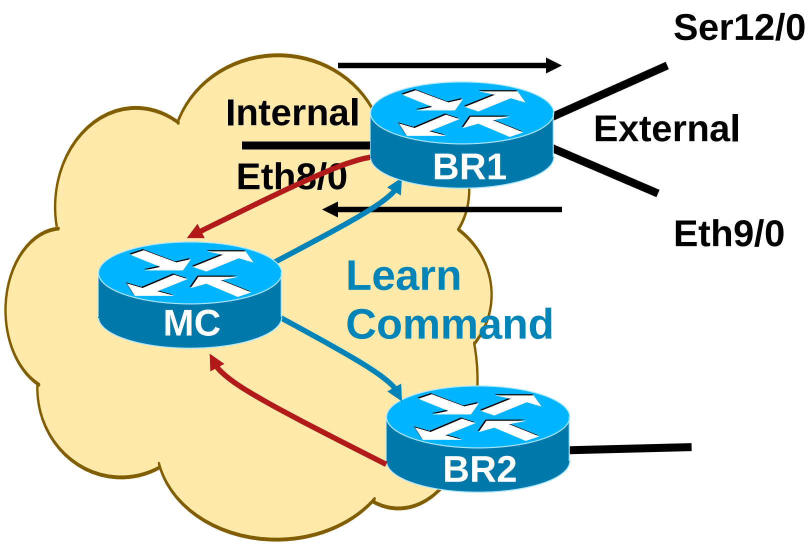
Prefix Learning—12.4(5)

MC Configuration

```
key chain key1
 key 1
 key-string oer
oer master
 learn
  throughput
  monitor 1
  periodic 0
 border 10.10.10.1 key-chain key1
  interface Ethernet8/0 internal
  interface Ethernet9/0 external
  interface Serial12/0 external
```

BR Configuration

```
key chain key1
 key 1
 key-string oer
oer border
 local ethernet 8/0
 master 10.10.10.2 key-chain key1
```



Entering Traffic Classes in the MC
Traffic Class Learning

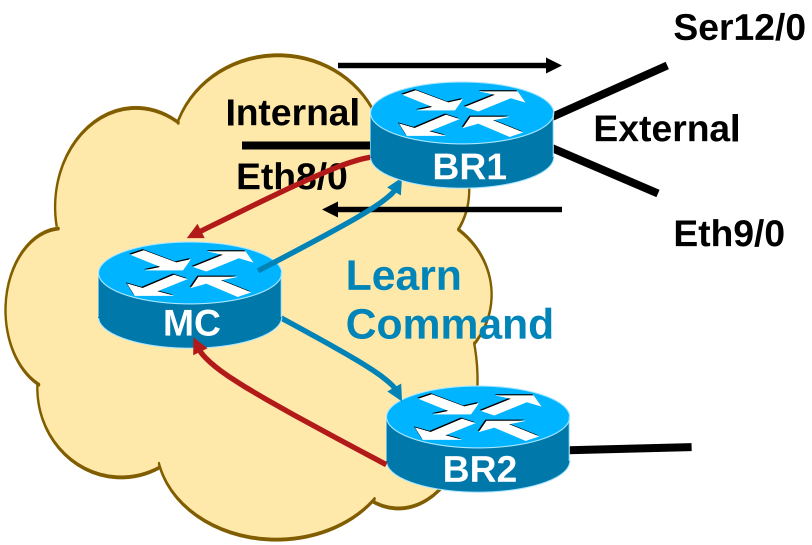
Application Learning—12.4T

MC Configuration

```
oer master
 learn
  throughput
  monitor 1
  periodic 0
  traffic-class keys dscp
 border 10.10.10.1 key-chain key1
  interface Ethernet8/0 internal
  interface Ethernet9/0 external
  interface Serial12/0 external
```

BR Configuration

```
oer border
 local ethernet 8/0
 master 10.10.10.2 key-chain key1
```



Entering Traffic Classes in the MC
Traffic Class Configuration

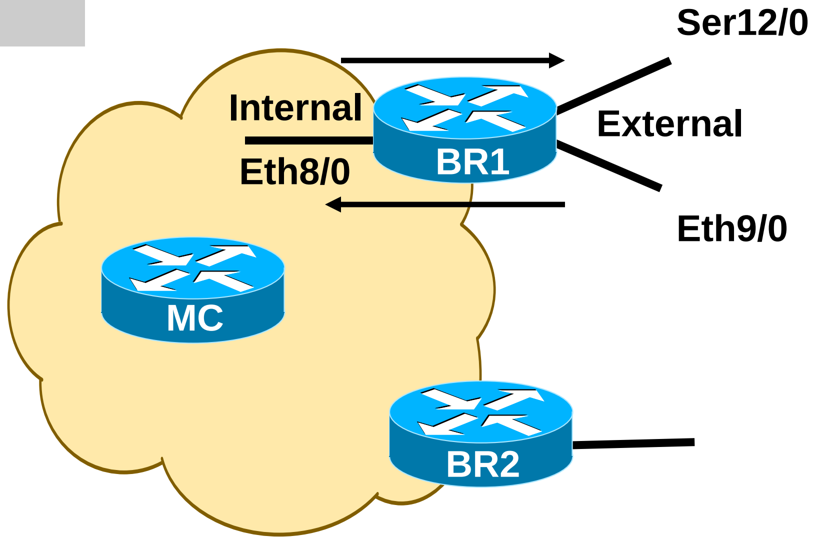
Prefix Configuration—12.4(5)

MC Configuration

```
oer master
 policy-rules MYMAP
 border 10.10.10.1 key-chain key1
  interface Ethernet8/0 internal
  interface Ethernet9/0 external
  interface Serial12/0 external
ip prefix-list PFX seq 5 permit 100.1.0.0/16
oer-map MYMAP 10
 match ip address prefix-list PFX
```

BR Configuration

```
oer border
 local ethernet 8/0
 master 10.10.10.2 key-chain key1
```



Entering Traffic Classes in the MC
Traffic Class Configuration

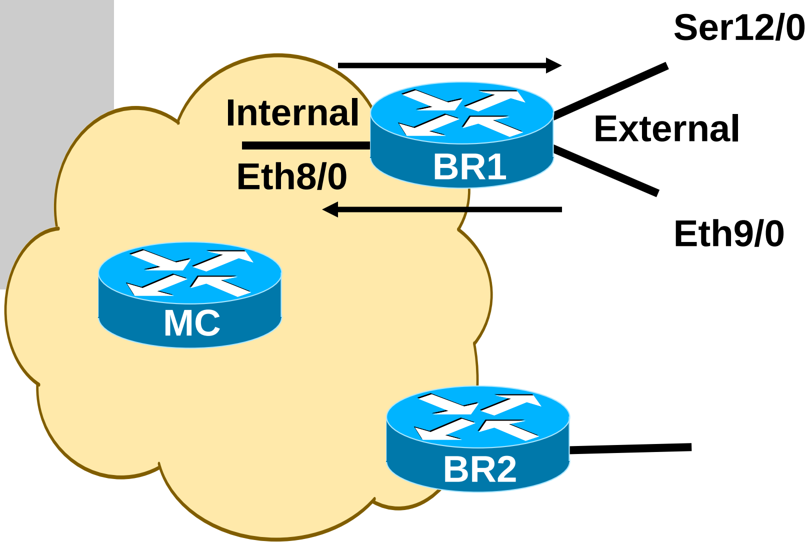
Application Configuration—12.4(6)T

MC Configuration

```
oer master
 policy-rules MYMAP
 border 10.10.10.1 key-chain key1
  interface Ethernet8/0 internal
  interface Ethernet9/0 external
  interface Serial12/0 external
ip prefix-list PFX seq 5 permit 100.1.0.0/16
ip access-list extended DSCP
 permit ip any 100.1.0.0 0.0.255.255 dscp cs7
oer-map MYMAP 10
 match ip address prefix-list PFX
 set delay threshold 300
oer-map MYMAP 30
 match ip address access-list DSCP
 set delay threshold 50
```

BR Configuration

```
oer border
 local ethernet 8/0
 master 10.10.10.2 key-chain key1
```



Measuring Traffic Class Performance

- Active
  - OER enables IP SLA feature
  - Probes sourced from BRs
  - icmp probes learned or configured
  - tcp, udp, jitter need `ip sla responder`

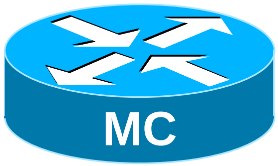
Delay
Reachability
Jitter 12.4(6)T
MOS 12.4(6)T

```
oer master
 active-probe echo 70.1.1.1
oer-map MYMAP 30
 set active-probe jitter 30.1.1.1 target-port 1000
      codec g729a
 set probe frequency 4
```

- Passive
  - OER Netflow monitoring of traffic classes
- Monitor modes

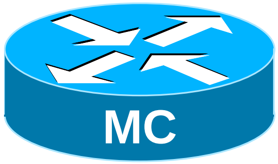
```
oer master
 mode monitor { both | active | passive}
```

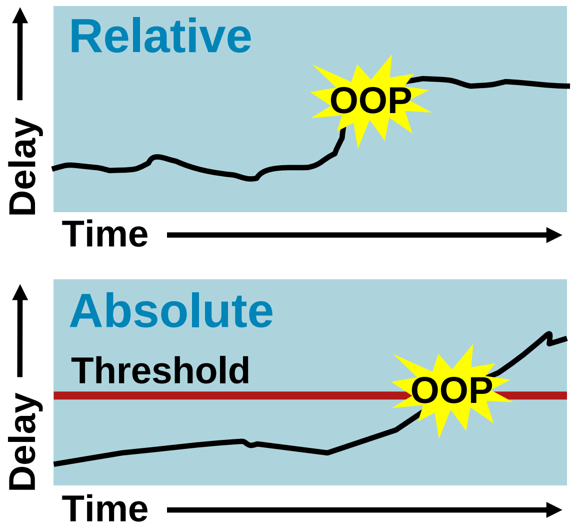
Delay
Loss
Reachability
Egress BW
Ingress BW



Applying Policy
Traffic Classes and Link

- Unreachable always applied
- Default policy
  - Traffic class relative delay
    - If delay increases, re-route traffic class
  - External link utilization
    - If utilization >75%, re-route traffic classes
- Policy type
  - Relative (default)
    - If metric rises sharply, then OOP
  - Absolute
    - If metric exceeds threshold, then OOP





Applying Policy
Traffic Classes and Link

- Global

  `show oer master`

```
oer master
 delay relative 20
 loss threshold 10000
 max-range-utilization percent 30
```

- Per traffic class

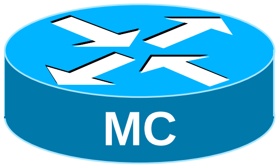
  `show oer master prefix 100.1.0.0/16 policy`

```
oer-map MYMAP 10
 set delay relative 20
 set loss threshold 10000
```

- External link

  `show oer master border detail`

```
oer master
 border 10.10.10.1
 interface Serial12/0 external
  max-xmit-utilization percentage 80
  maximum utilization receive percentage 80
```



Selecting “Best” Traffic Class Exit

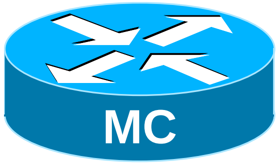
OER selects a policy conforming exit

1. Gather traffic class measurements for all exits
2. Gather link utilization for all external interfaces
3. Exits with no measurements ignored
4. Measurements applied using priority with variance
5. Exits within variance are candidates

After All Priorities Examined:

1. If a single candidate
   - Use single candidate
2. If multiple candidates includes current exit
   - Choose current exit
3. Else, randomly choose a candidate

```
oer master
 resolve delay priority 4 variance 20
 resolve loss priority 6 variance 20
 resolve util priority 8 variance 20
```

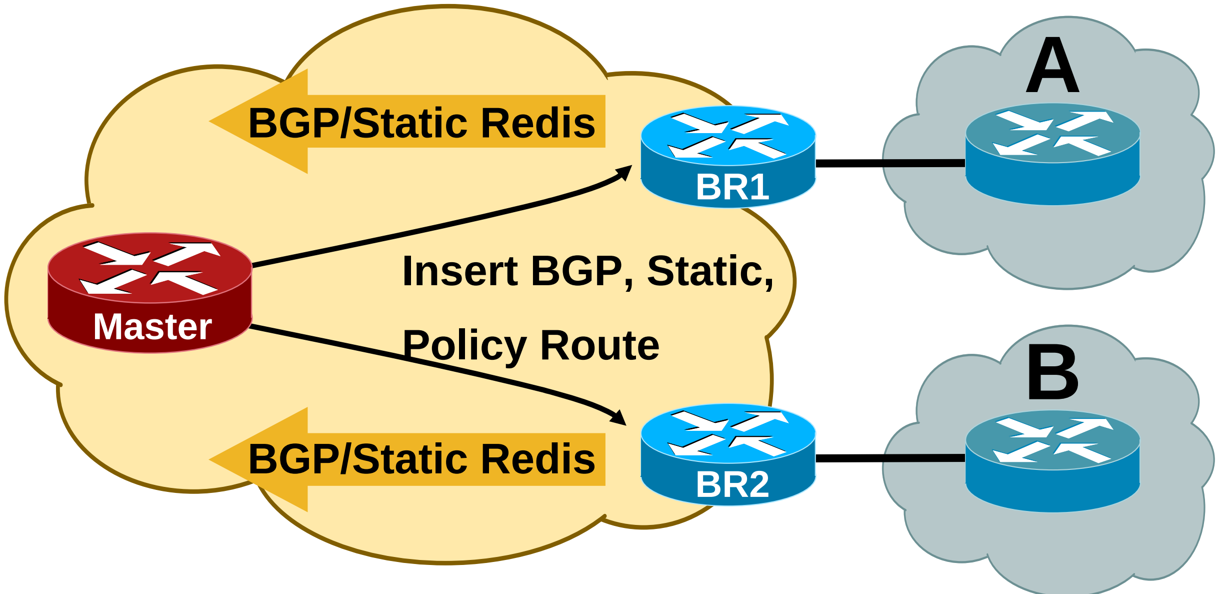


Enforcing Traffic Class Exit

- MC tells BR to insert prefix in BGP or static table
- MC tells BR to insert Traffic Class in policy route

Enable enforcement

```
oer master
 mode route control
```

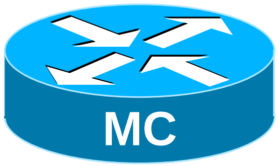


- **Modify BGP local preference**
  Local preference must be highest
- **Install static route at the exit**
  Redistribute static should be configured
- **Report any route changes**

Influencing Prefix Entrance

- Passive measurements gathered for all entrances
- Measurements applied in priority order
  - Priority with variance applied
- Identify entrances to downgrade
- Downgrade entrance using BGP advertisement
  - AS path prepend
  - Append downgrade BGP community aa:nn
    - ISP specific community
    - ISP AS prepend community
    - ISP local prefix community

```
oer master
 border 10.10.10.1
 interface Serial12/0 external
  downgrade bgp community aa:nn (optional)
```



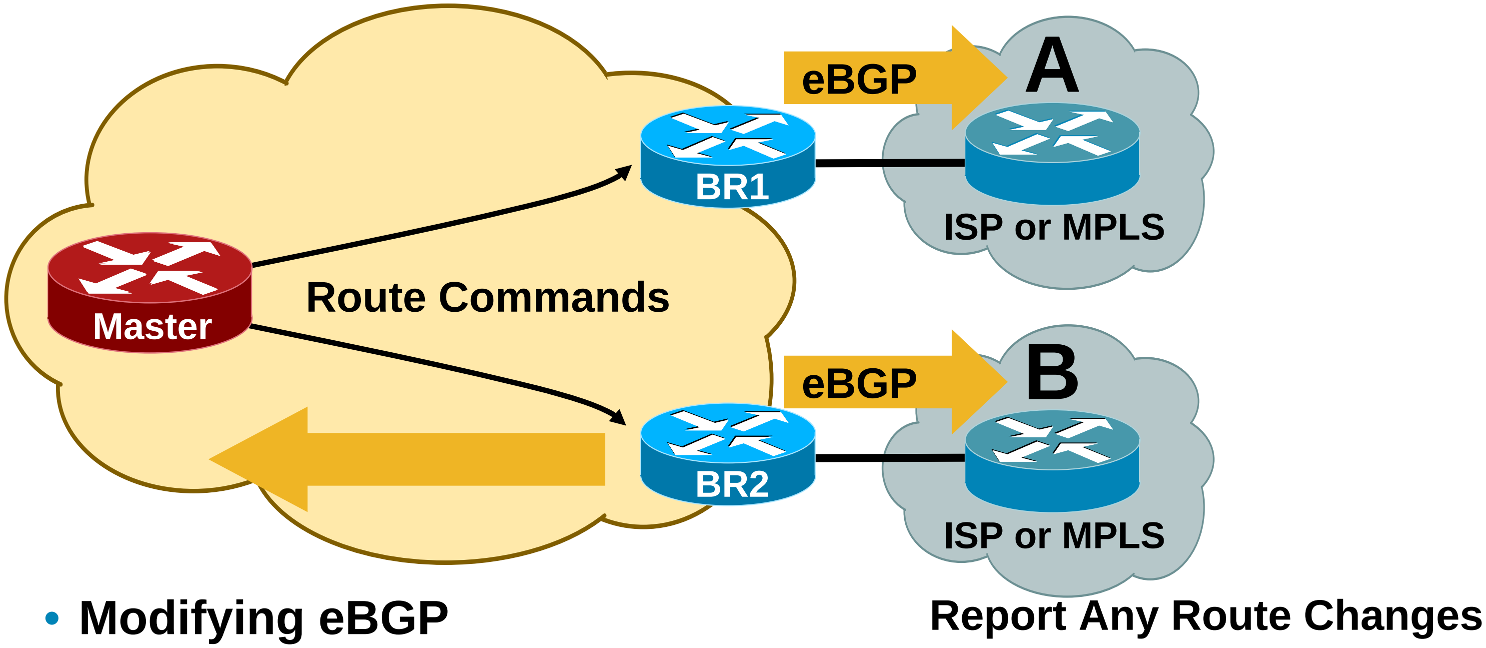
Influencing Prefix Entrance
12.4T

- MC tells BR to modify eBGP advertisement

    Learn which prefixes are advertised

    `show oer master prefix inside`

```
oer master
 learn
 inside bgp
```



- **Modifying eBGP**

    Prepend AS Hop(s) (default)

    Append BGP downgrade community aa:nn

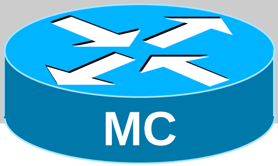
Reporting Results

- Syslog

```
sh log | i 100.1.1.0
*Apr 26 22:58:20.919: %OER_MC-5-NOTICE: Discovered Exit for prefix 100.1.1.0/24, BR
     10.10.10.1, i/f Et9/0
*Apr 26 23:03:14.987: %OER_MC-5-NOTICE: Route changed 100.1.1.0/24, BR 10.10.10.1, i/f
     Se12/0, Reason Delay, OOP Reason Timer Expired
*Apr 26 23:09:18.911: %OER_MC-5-NOTICE: Passive REL Loss OOP 100.1.1.0/24, loss 133, BR
     10.10.10.1, i/f Se12/0, relative loss 23, prev BR Unknown i/f Unknown
*Apr 26 23:10:51.123: %OER_MC-5-NOTICE: Route changed 100.1.1.0/24, BR 10.10.10.1, i/f Et9/0,
     Reason Delay, OOP Reason Loss
```

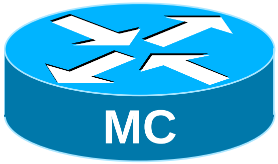
- Show commands

```
sh oer master prefix
Prefix         State    Time Curr BR      CurrI/F      Protocol
           PasSDly PasLDly  PasSUn  PasLUn PasSLos PasLLos
           ActSDly ActLDly  ActSUn  ActLUn   EBw   IBw
--------------------------------------------------------------------------------
100.1.1.0/24   HOLDDOWN  42 10.10.10.1   Et9/0        STATIC
                16      16       0       0       0       0
                 U       U       0       0      55       2
```

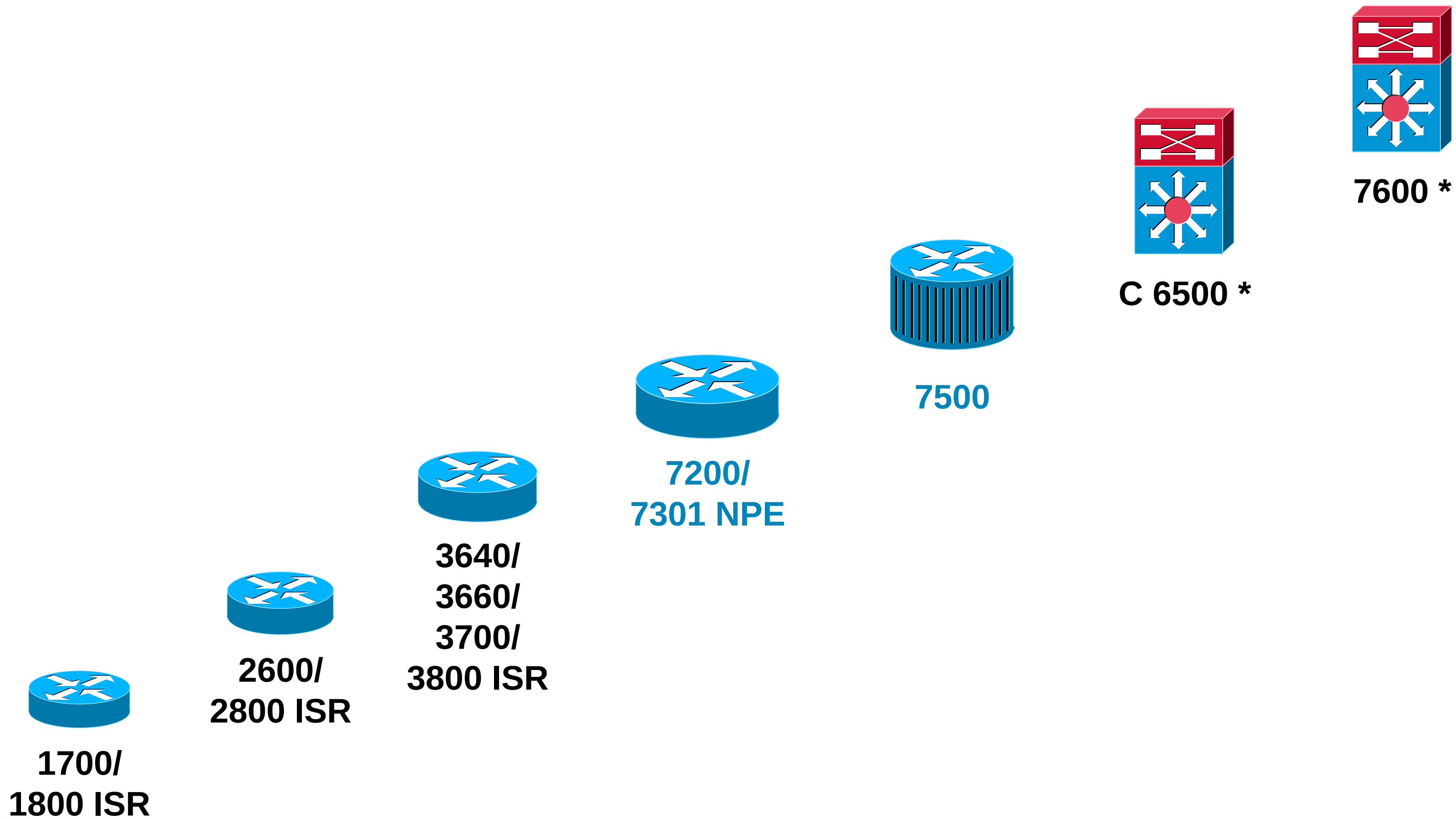


OER CISCO IOS Support

- Released in 12.3(8)T (May 17, 2004)
- 12.4
  - Prefix optimization
- 12.4T
  - Traffic class optimization
  - Voice optimization
- 12.2S on the C6500 and 7600 platforms
  - Prefix and traffic class management
  - 1Q07



OER Platform Support



- •C6500/7600 Supported in 12.2S
- •1Q07

Agenda

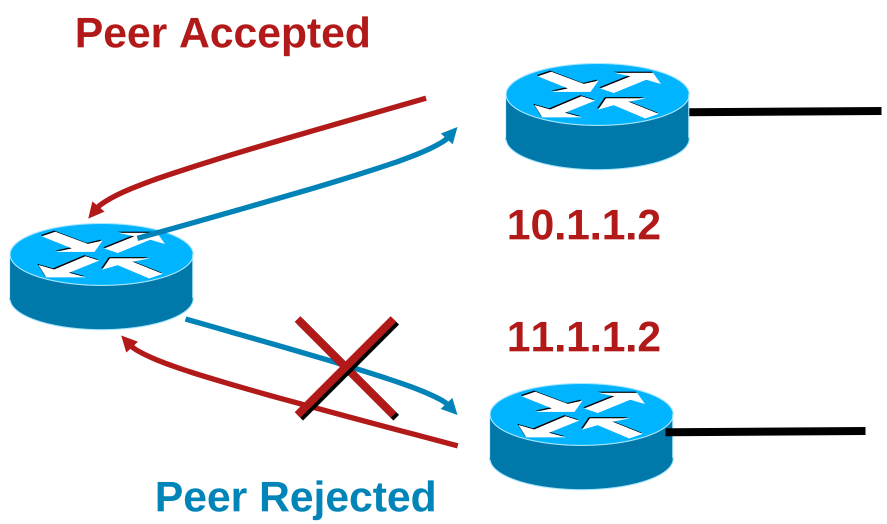
- Faster Convergence
- BGP → TCP Enhancements
- 4-byte AS
- NSR – Non Stop Routing
- OER - Optimized Edge Routing
- Whiteboard Features
    - Dynamic Peering
    - Admin Down Cease
    - MD5 Static Key Rollover
    - Others

Whiteboard Features - Caveat

Caveat: These features are being considered for IOS. However, this list changes based on customer and business needs. Therefore, these features may change in priority, schedule, and even implementation details. If the business needs change, features may be added or removed from this list.

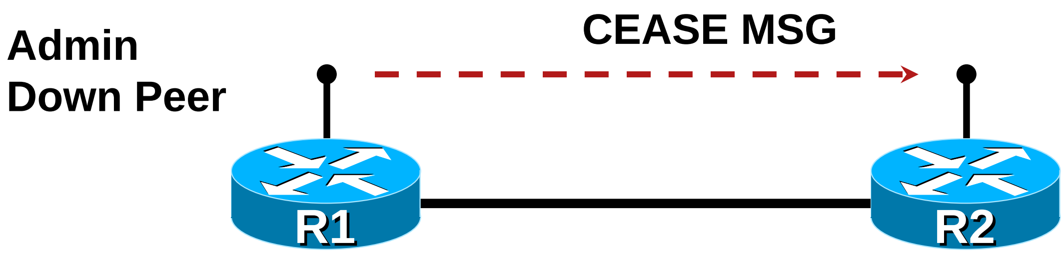
Whiteboard Features - Dynamic Peering

bgp listen range 10/8 peer-group X



- Allows significant reduction in configuration overhead
- Permits peers to be formed when sourced from specified prefix range
- Potential security issues - be forewarned about deployment scenarios
- There will be initial scope limitations for scaling reasons

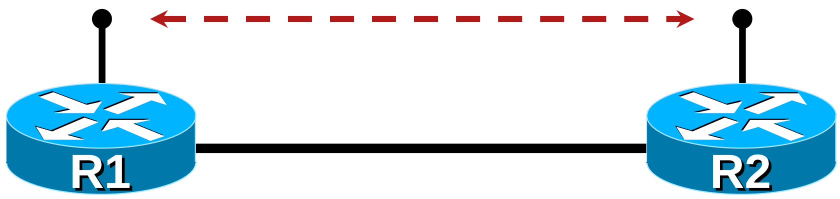
Whiteboard Features - CEASE



- Graceful Shutdown Mechanism
- Faster convergence, allowing R2 to run BESTPATH without having to wait for the session to timeout.
- When peer is administratively shutdown, CEASE Notification message will be sent.

Whiteboard Features - MD5 Static Key Rollover

R1 changes MD5 Key with R2 on established session



- R1 and R2 Peer using MD5 static Key
- In the past, an update of the key would cause the session to transition
- This feature allows a dead interval period where the MD5 keys do NOT match and the session will not reset.
- Allows both sides to be updated within a close proximity of time, while the session is maintained thus greatly reducing churn and unnecessary re-convergence.
- Basically, Keychain functionality

Whiteboard Features - Others

- BGP Event Log

  Track BGP issues and history with event-log reporting.

- Session/Neighbor Dampening

  Utilizing Dampening Logic on Sessions/Neighbors rather than prefixes

- Enforce First AS per Neighbor

  Making the Enforce First AS a per peer knob rather than global

  Be aware of security issues here for spoofing!

IOS-XR BGP Design Goals

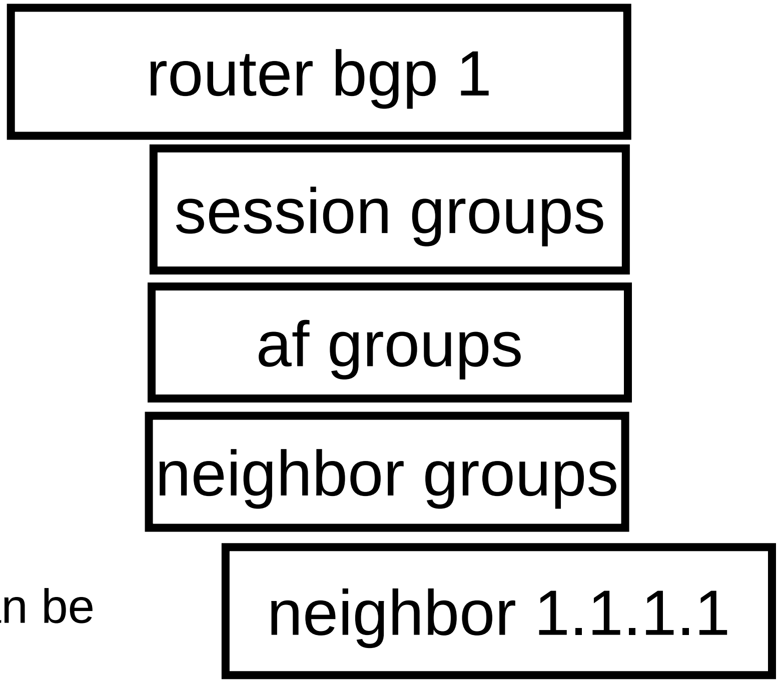
- Scalability
  - Thousands of peers
  - Millions of prefixes & paths
- Reliability
  - NSF, Graceful Restart, Process restart capabilities
- Performance
  - Faster convergence with large number of peers/prefixes

CLI Partitioning

- No address-family enabled by default
- Explicitly enable address-family support
- Neighbor based config
  - Global AF-Independent configuration
  - Global AF-Dependent configuration
  - Neighbor AF-Independent configuration
  - Neighbor AF-Dependent configuration

Configuration and Update Grouping

- Configuration grouping
  - Support hierarchical configuration
  - Clearer semantics for inheritance
  - Supports three types of groups
    - session-group
    - af-group
    - neighbor-group
- Update grouping
  - Dynamically group peers whose updates can be replicated
  - Peer to update-group assignment is based on configuration of outbound policies



Example Configuration

```
router bgp 1
    router-id 10.0.0.1
    address-family ipv4 unicast
        scan-time 20
        network 10.1.0.0 mask 255.255.0.0
        redistribute static
    address-family ipv4 multicast
        redistribute isis 1
        network 224.1.0.0 mask 255.255.0.0
    neighbor 1.1.1.1
        remote-as 1
        timers 10 30
        address-family ipv4 unicast
            next-hop-self
            route-policy pass-all in
            route-policy pass-all out
        address-family ipv4 multicast
            next-hop-self
            route-policy pass-all in
            route-policy pass-all out
    neighbor 2.1.1.1
        remote-as 2
        ebgp-multihop 4
        address-family ipv4 unicast
            max-prefix 1000
        address-family ipv4 multicast
            route-policy pass-all in
            route-policy pass-all out
            route-reflector-client
```


Session Group

- Contains only address family independent config
- Can inherit from another session-group

```
router bgp 100
 session-group sg-generic
  password encrypted xyz
  version 4
 !
 session-group sg-internal
  remote-as 222
  update-source Loopback0
  use session-group sg-generic
 !
!
 neighbor 1.1.1.1
  use session-group sg-internal
```

AF Group

- Contains only AF specific configuration for a single AF
- Can inherit from another af-group

```
router bgp 100
 af-group af-pol address-family ipv4 unicast
  route-policy pass-all in
 !
 af-group af-nei address-family ipv4 unicast
  use af-group af-pol
  weight 600
 !
 neighbor 1.1.1.1
  remote-as 222
  address-family ipv4 unicast
   use af-group af-nei
  !
!
```


Neighbor Group

- Can contain address-family as well as session specific configurations
- Can inherit from session-group, af-group or another neighbor-group

```
router bgp 100
 !
 neighbor-group ng-internal
  use session-group sg-internal
  address-family ipv4 unicast
   use af-group af-nei
  !
 !
 neighbor 1.1.1.1
  use neighbor-group ng-internal
 !
!
```

Inheritance Rules

- Simple order of precedence for config
  - If item is configured specifically for a peer, use the peer's specific configuration
  - If peer belongs to a neighbor-group, session-group, or af-group, use configuration from defined group
  - Otherwise use default value
- Session-group and af-group have higher precedence over neighbor-group

Config Grouping Show Commands

- Combining session groups, af groups, and neighbor groups may result in a complex configuration
- New show command provides configuration which is actually being used by a peer

  show bgp neighbor <addr> configuration

- New show commands for various group types

  show bgp session-group <name>

  show bgp af-group <name>

  show bgp neighbor-group <name>

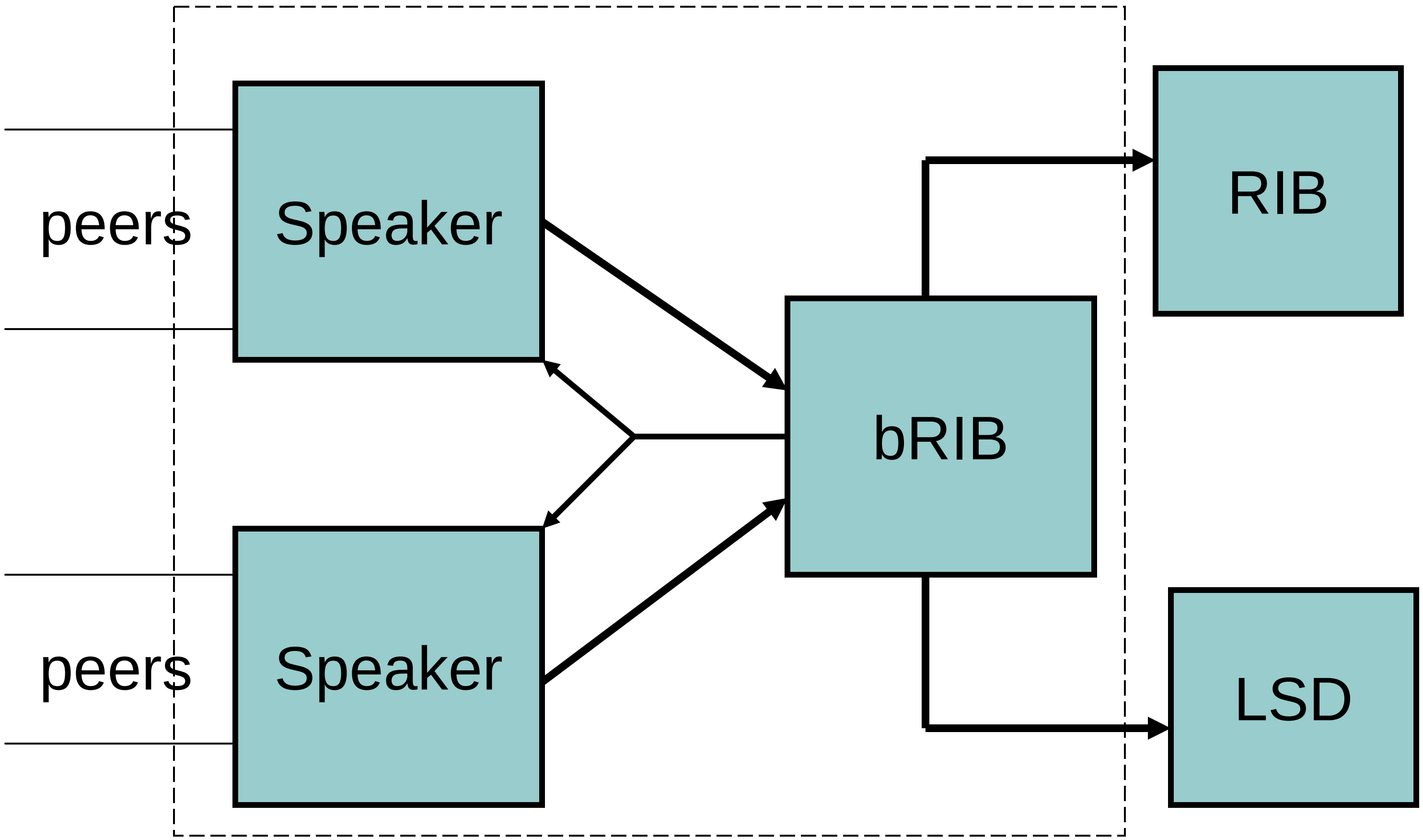
Stand-alone and Distribute modes

- IOS-XR BGP runs in one of two modes (configurable)
  - Stand-alone mode
  - Distributed mode
- Stand-alone is typical usage - single RP system
- Stand-alone spawns two processes
  - BGP Process Manager (BPM)
  - BGP Speaker
- Distributed mode
  - Distribute peers among multiple RPs
  - Distribute address-families among multiple RPs
- Three process are spawned
  - BGP Process Manager (BPM)
  - BGP RIB (bRIB)
  - BGP Speaker

BGP Process Manager (BPM)

- Configuration verification, Basic sanity checking
- Starts speaker and bRIB processes
- Mode transition (standalone ←→ distributed)
- Performs neighbor allocation to speakers
- Publishes AS/router-id etc. information
- Processes BGP config templates
- Publishes neighbor config to speakers

Distributed BGP



BGP Speaker

- Multiple BGP speaker processes (max 15)
- Receive updates from peers [1]
- Calculate partial bestpaths [2]
  - Only paths received from local neighbors considered
  - Speakers don’t have access to entire BGP-RIB
  - All steps up to MED comparison
- Send partial bestpaths to bRIB [3]

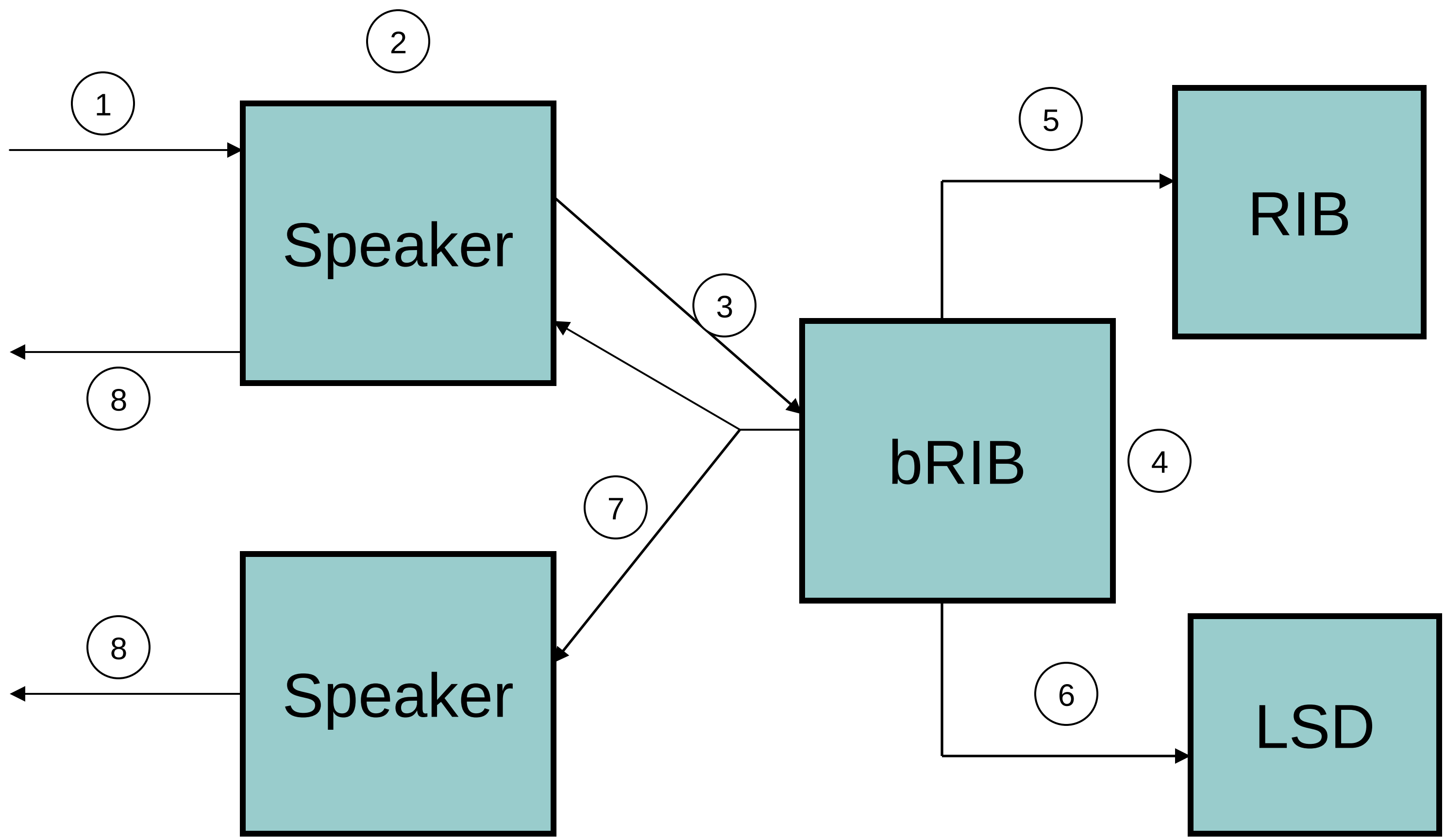
BGP Speaker

- Receive final bestpaths from bRIB [7]
- Send updates to neighbors [8]
- Speakers support all AFIs
- Can operate in standalone mode.
  - Full bestpath calculation
  - Directly updates global RIB and LSD.

BGP RIB (bRIB)

- IPv4, IPv6, VPNv4 bRIBs
- Receives partial bestpaths from speakers [3]
- Computes final bestpaths [4]
- Performs Import processing
- Installs bestpaths into RIB/allocate labels from LSD. [5/6]
- Generates aggregates and locally sourced networks
- Sends bestpaths to speakers [7]

Distributed BGP Flow



Show Commands

- **Useful show commands**

```
show bgp <afi> <safi> process detail
show bgp process performance-statistics detail
show bgp <afi> <safi> summary
show bgp <afi> <safi>
show bgp update-group
show bgp neighbors
show bgp neighbor performance-statistics
show process bgp
```

Show commands

show bgp process

```
BGP is operating in DISTRIBUTED mode
Autonomous System: 1
Router ID: 10.0.0.1
Cluster ID: 10.0.0.1
Fast external fallover enabled
Neighbor logging is enabled
Enforce first AS enabled
Default local preference: 100
Default keepalive: 60
Update delay: 120

Address family: IPv4 Unicast
Dampening is not enabled
Client reflection is enabled
IGP synchronization not enabled
Auto-summary is not enabled
Main Table Version: 32218

Node     Process         Nbrs Estab Rst  Upd-Rcvd Upd-Sent Nfn-Rcvd Nfn-Sent
node2    bRIB    1          0     0   0         0        0        0        0
node3    bRIB    3          0     0   0         0        0        0        0
node2    Speaker 1          2     2   1    230347        9        0        0
node3    Speaker 2          2     2   1    159520       10        0        0
```


Show commands

```
show bgp ipv4 unicast summary

BGP router identifier 30.30.30.1, local AS number 1
BGP generic scan interval 60 secs
BGP table state: Active
BGP main routing table version 101068
BGP scan interval 60 secs
BGP is operating in DISTRIBUTED mode.

Process    Id    RecvTblVer    bRIB/RIB  LabelVer  ImportVer  SendTblVer
Speaker    1             34          34         0          0          34
Speaker    2              1           1         0          0           0
bRIB       1             61          61         0          0          61

Neighbor        Spk     AS MsgRcvd MsgSent    TblVer  InQ OutQ Up/Down  St/PfxRcd
11.0.1.2          1      2     999     996        31    0    0 01:09:04        5
11.0.2.2          1      1     999     996        31    0    0 01:09:04        5
```

Show commands

show bgp vpnv4 unicast summary

```
DRP/0/2/CPU1:rgr2-q1#show bgp vpnv4 u sum
BGP router identifier 10.0.0.1, local AS number 1
BGP generic scan interval 60 secs
BGP table state: Active
BGP main routing table version 68541
BGP scan interval 60 secs
BGP is operating in DISTRIBUTED mode.

Process    Id     RecvTblVer    bRIB/RIB  LabelVer  ImportVer  SendTblVer
Speaker     1              1           1         0          0           0
Speaker     2           2115        2115         0          0        1595
bRIB        3          68541       68541     68541      68541       68541

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  St/PfxRcd
11.0.4.2          2     1    1019    1180     1595    0    0 01:09:10        5
```

Show commands

show bgp vrf <name> summary

```
DRP/0/2/CPU1:rgr2-q1#show bgp vrf t1 sum
BGP VRF t1, state: Active
BGP Route Distinguisher: 10.0.0.1:0
BGP router identifier 10.1.0.1, local AS number 1
BGP generic scan interval 60 secs
BGP table state: Active
BGP main routing table version 68541
BGP scan interval 60 secs
BGP is operating in DISTRIBUTED mode.

Process    Id     RecvTblVer    bRIB/RIB  LabelVer  ImportVer  SendTblVer
Speaker    1               1           1         0          0           0
Speaker    2            2115        2115         0          0        1595

Neighbor        Spk    AS   MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  St/PfxRcd
11.0.3.2          2  1001      1025    1023     1595    0    0 01:10:48         5
```

Debug commands

- Debugs may be enabled concurrently for the same neighbor with and without a filter, or with multiple filters

```
debug bgp update 10.0.101.1 in
debug bgp update 10.0.101.1 in <route-policy>
```

- You can enable debugs for multiple neighbors at the same time

BGP – show/debug

- Received updates from peers

```
show bgp neighbor z.z.z.z routes
debug bgp update  z.z.z.z  in <rpl>
```

- Calculate Partial best-paths in speaker

```
show bgp x.x.x.x/y speaker <speaker-id>
debug bgp update z.z.z.z in <rpl>
```

- Speaker sent partial bestpath to bRIB

```
show bgp x.x.x.x/y speaker a
show bgp x.x.x.x/y brib
debug bgp brib-update <rpl> in speaker a
debug bgp brib-update <rpl> in brib 1
```

BGP – show/debug

- Computes final bestpaths

```
show bgp x.x.x.x/y [brib <id> | speaker <id>]
show bgp x.x.x.x/y
show bgp x.x.x.x/y bestpath-compare
```

- Installs bestpaths into RIB

```
show bgp x.x.x.x/y
show route x.x.x.x/y
debug bgp rib <rpl>
debug routing ipv4 <acl>
```

- Allocate label from LSD – need to load mpls pie

```
show bgp vrf <vrf_name> x.x.x.x/y
show bgp vrf <vrf_name> labels
show mpls lsd forwarding label <label>
```


BGP – show/debug

- Sends bestpaths to speakers

```
show bgp x.x.x.x/y brib
show bgp x.x.x.x/y speaker <spkr-id>
debug bgp brib-update <rpl> out speaker <spkr-id>
debug bgp brib-update <rpl> out brib <brib-id>
```

- Send updates to neighbors

```
show bgp neighbor w.w.w.w advertise
debug bgp update w.w.w.w out <rpl>
```

- Session bring up

```
show bgp neighbor x.x.x.x
debug bgp io
```


Meet the Experts
IP and MPLS Infrastructure Evolution

- Andy Kessler
  Technical Leader

- Beau Williamson
  Consulting Engineer

- Benoit Lourdelet
  IP services Product manager

- Bertrand Duvivier
  Consulting Systems Engineer

- Bruce Davie
  Cisco Fellow

- Bruce Pinsky
  Distinguished Support Engineer

Meet the Experts
IP and MPLS Infrastructure Evolution

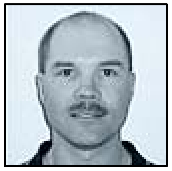
- Gunter Van de Velde
  Technical Leader

- John Evans
  Distinguished Systems Engineer

- Oliver Boehmer
  Network Consulting Engineer

- Patrice Bellagamba
  Consulting Engineer

- Shannon McFarland
  Technical Leader



Meet the Experts
IP and MPLS Infrastructure Evolution

- Andres Gasson
  Consulting Systems Engineer

- Steve Simlo
  Consulting Engineer

- Toerless Eckert
  Technical Leader

- Dino Farinacci
  Cisco Fellow & Senior Software Engineer

Recommended Reading

BRKIPM -3005

- Practical BGP
- Optimal Routing Design

Available in the Cisco Company Store

Any Questions ?

BGP Sessions

- BGP plays a role in OER:
- IPM-2015 – Deploying Optimized Edge Routing

  Friday at 8:30
- IPM-3004 - IGP, BGP, and PIM Fast Convergence

  Wednesday at 15:30
- Various IP& MPLS Sessions