



Monitoring & Troubleshooting BGP Peering



Common BGP Peering Issues

- + Most common mistakes are misconfigurations
 - + Forgot to configure BGP on peer
 - + Wrong ASN configured
 - + Wrong peer IP address
 - + Interface to peer is disabled
 - + IGP or static route reachability to peer missing
 - + Missing "ebgp-multihop" command (for eBGP peers)
 - + Missing "update-source" command for Loopback peering
- + Less common issues
 - + Intermediary device blocking TCP 179
 - + Extreme network congestion dropping packets
 - + Layer-2 loops (Spanning-Tree bugs)
 - + Layer-2 WAN reachability issues
 - + Excessively high CPU on peer (DoS attack?)



Initial Steps in Troubleshooting

- + Can you ping your peer?
 - + If yes, then fundamental problem is not a Layer-1 or Layer-2 connectivity issue
 - + If no, check routing table for reachability (on both sides)
- + Is BGP correctly configured on both routers?
 - + If yes, (and peers can ping each other) then check for devices or features which could be blocking TCP 179 session establishment.
 - + Check for high CPU utilization on either device
 - + Check output of "show interface" counters for indications of link congestion and packet drops



Show IP BGP Neighbor

- + For BGP peer verification and troubleshooting, the key elements you need to look for in this command are;
 - + Neighbor state (should be "Established")
 - + Neighbor type ("external link" or "internal link")
 - + Neighbor's ASN
 - + Neighbor capability

```
R2#sho ip bgp neighbor
BGP neighbor is 1.2.1.1, remote AS 1, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Active
    Last read 00:00:25, last write 00:00:50, hold time is 180, keepalive interval is
    60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
```



- One of the most common reasons a neighbor would show as "Idle" is because you've lost your route to that neighbor.
- Secondly, a neighbor that WAS Established but suddenly disappears, causing the local BGP Hold Timer to expire will cause your router go to into the "Idle" state for this peer and then flip-flop between "Idle" and "Active"

Cisco Resources

- + <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/22166-bgp-trouble-main.html>



Final Thoughts

- + A BGP peer in the “Established” state doesn’t guarantee propagation of BGP routes
- + Several things could prevent a BGP router from propagating routes to another BGP peer:
 - + iBGP forwarding rules
 - + Next-Hop inaccessibility
 - + BGP outbound filtering
 - + BGP Well-Known Communities







BGP Basic Prefix Advertisement



Injecting Routes into BGP

- + There are two primary ways to inject routes into BGP:
 - + By using the BGP **network** command
 - + By using redistribution
- + Routes injected by either of these methods have some advantages and requirements.



Route Injection Using Network Command

- + The BGP “Network” command is used to inject routes into the BGP process
 - + Looks for a matching IGP route
 - + Must be an exact match
 - + Typically requires a “mask” keyword

```
R1#show ip route | i C
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
C       10.20.1.0/24 is directly connected, Loopback1
C       111.111.111.0/24 is directly connected, Loopback0
C       130.130.0.0/16 is directly connected, Loopback2
```

```
router bgp 1
  bgp log-neighbor-changes
  network 10.20.1.0 mask 255.255.255.0
  network 111.111.111.0 mask 255.255.255.0
  network 130.130.0.0
  neighbor 1.2.1.2 remote-as 2
```



- IF the same prefix is matched by BOTH a “network” and “redistribute” statement...the “network” statement wins.

Injecting Routes Using Redistribution

- + Redistribution can be an easy way to inject many IGP routes into BGP table.
- + This can be accomplished by using the **redistribute** command in BGP configuration mode.
- + To verify BGP routes, use the following command:
 - + **show ip bgp**

```
router bgp 3
  bgp log-neighbor-changes
  redistribute connected
  redistribute ospf 1
  neighbor 2.3.2.2 remote-as 2
```



Downsides of Redistribution

- + Redistribution can have some pitfalls:
 - + Will also advertise private IGP networks if not paired with filtering
 - + If redistributing static routes, actual reachability is not tracked.
 - + Redistributed routes are less preferred than routes advertised with “network” command (same behavior as EIGRP and OSPF).



- Redistributing default routes (0.0.0.0) will not work.

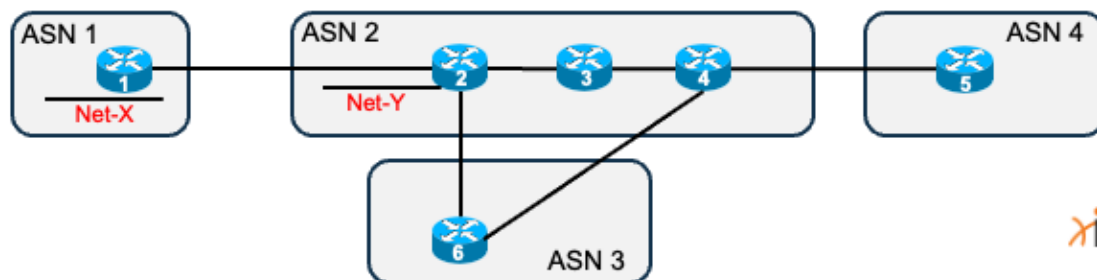
Prefix Injection...What Happens?

- + Matching route identified in Routing Table
- + Origin code set, based on method of injection
- + Local ASN added to AS-Path attribute, *if sending to eBGP peer*
- + Local Preference added *when sending to iBGP peer*
- + Next-Hop attribute set, value depends on peer type.



BGP Route Propagation Rules

- + BGP prefixes can be learned or originated
 - + Learned = Received BGP prefix from BGP peer within an "Update" message
 - + Originated = Router locally injected IGP route into BGP with "network" or "redistribute" commands
- + Regardless of route origination, send all prefixes to eBGP peers
- + iBGP-learned routes can only be sent to eBGP peers



Verifying Prefix Injection

```
R2#sho ip bgp neighbor 1.2.1.1 advertised-routes
BGP table version is 3, local router ID is 2.3.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>  22.22.22.0/24    0.0.0.0              0         32768 i
*>  33.33.33.0/24    2.3.2.3              0          0 3 i

Total number of prefixes 2
R2#
```







Understanding the BGP Next-Hop



BGP Next-Hop

- + BGP prefixes sent to eBGP neighbors have the next-hop changed.



- + Prefixes sent to iBGP neighbors (initially learned via eBGP) do NOT change the next-hop



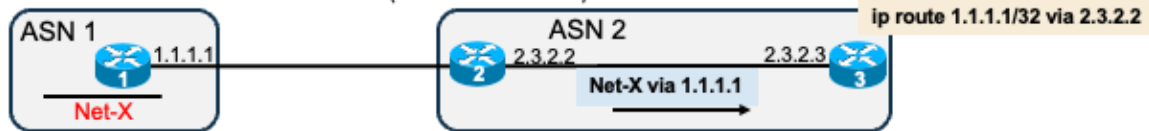
- + This can be a problem if iBGP peers can't reach the next-hop.

Solving The BGP Next-Hop Problem

- + There are several ways to fix the next-hop problem:
 - + Advertise the next-hop via an IGP



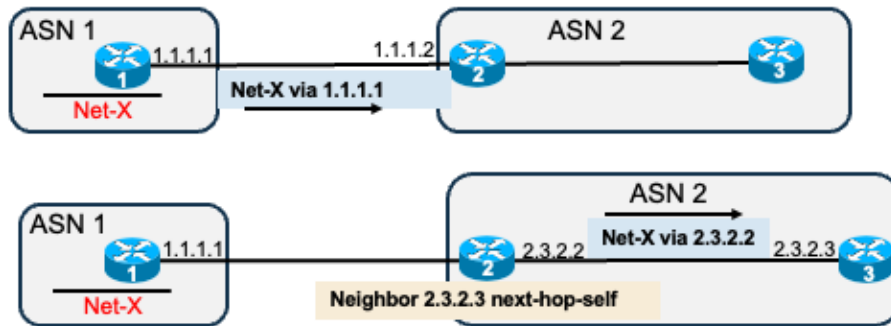
- + Utilize static routes (not scalable)



- + Is there another way?

BGP Next-Hop-Self

- + Adding “next-hop-self” against an iBGP peer can ensure next-hop reachability.







Monitoring BGP Prefixes



Displaying BGP Routes

```
R2#show ip route bgp | begin (Gateway.*)
Gateway of last resort is not set

    33.0.0.0/24 is subnetted, 1 subnets
B       33.33.33.0 [20/0] via 2.3.2.3, 00:40:06
    111.0.0.0/24 is subnetted, 1 subnets
B       111.222.111.0 [200/0] via 1.2.1.1, 00:00:27
R2#
```



Understanding The BGP Table

```
R2#show ip bgp
BGP table version is 9, local router ID is 22.7.22.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	0.0.0.0	0.0.0.0	0		32768	1
r>	22.6.22.0/24	22.6.22.6	0			6 ?
*>	55.55.55.0/24	22.6.22.6	0			6 ?
*>	77.77.77.0/24	22.7.22.7	0			7 6688 123 4458 1001 3002 78009 65 i
*>i		5.8.5.8	0	100		8 4056 702 899 6711 65 i
*>	77.77.78.0/24	22.7.22.7	0			7 1
*>	166.66.66.0/24	22.6.22.6	0			6 ?



- When a BGP router has learned of a route via an IGP (or a static route) and then injects that route into BGP with a “network” or “redistribute” command, its OWN local BGP table will display the next-hop of the IGP router from which it originally learned the route.
- The next-hop is only 0.0.0.0 when the route being advertised by BGP is directly-connected to the local router.

Digging Deeping Into A Prefix

```
R2#show ip bgp 77.77.77.0/24
BGP routing table entry for 77.77.77.0/24, version 5
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
    1          2
  Refresh Epoch 2
  7 6688 123 4458 1001 3002 78009 65
  22.7.22.7 from *22.7.22.7 (22.7.22.7)
    Origin IGP, metric 0, localpref 100, valid, external
    rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  8 4056 702 899 6711 65 (Received from a RR-client)
  5.8.5.8 from *2.5.2.5 (5.8.5.5)
    Origin IGP, metric 0, localpref 100, valid, internal, best
    rx pathid: 0, tx pathid: 0x0
```

BGP Peer who
advertised the prefix

BGP Next-Hop



- Version is the “BGP Table Version”. This indicates how often this prefix has been updated, learned or changed since the BGP table was first initialized.
- BGP will dynamically group peers that are meant to receive the same BGP update into “update-groups”. Which peers are in each update group can be seen from the output of “show ip bgp update-group”

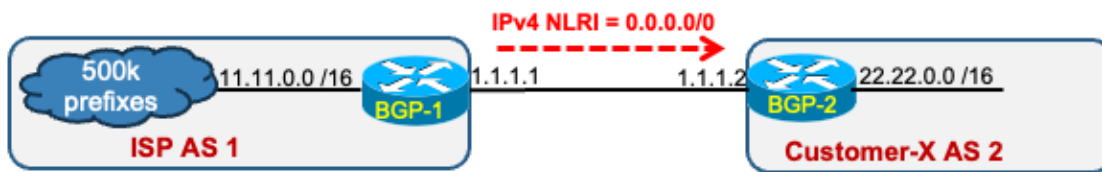




Injecting a BGP Default Route



BGP Default Routes



- + Two available methods to generate a default route:
 - + **Network 0.0.0.0**
 - ✓ Requires presence of default IGP route
 - + **Neighbor x.x.x.x default-originate**
- + Redistribution of IGP (or Static) default route into BGP does not work.
- + Using "aggregate-address" doesn't work either

```
R2(config-router)#aggregate-address 0.0.0.0 0.0.0.0 as-set summary-only
% Aggregating to create default makes no sense,
use a network statement instead.
```

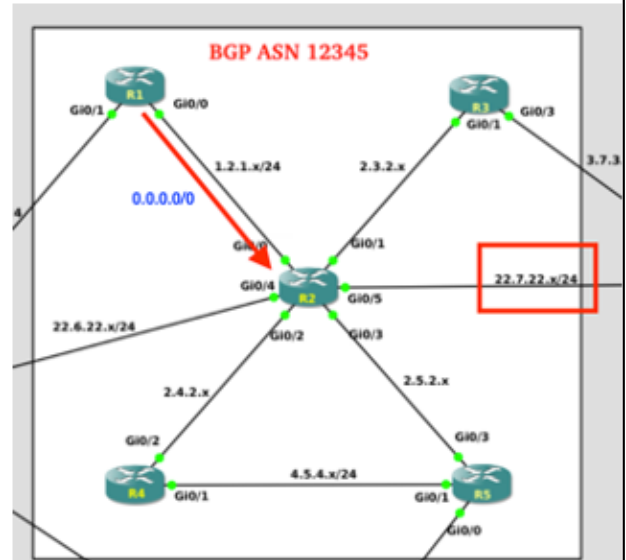


- Injection using the "neighbor default-originate" command allows you to conditionally advertise the default and only to certain neighbors...see next section.
- "network 0.0.0.0" subject to normal outbound BGP filtering rules.
- "neighbor default" NOT subject to normal outbound filtering mechanisms.
- A default route created by BGP can NOT be used to provide next-hop reachability (for either peer reachability or prefix next-hop reachability).

BGP Conditional Default Routes

- + Default routes can be configured to be conditional upon the presence of other routes
- + Requires matching on a route-map which matches the conditional route

```
access-list 1 permit 22.7.22.0 0.0.0.255
!
route-map Conditional permit 10
match ip address 1
!
router bgp 12345
neighbor 1.2.1.2 remote-as 12345
neighbor 1.2.1.2 default-originate route-map Conditional
```



- In this example, router R1 is originating a default route and sending it only to its peer, R2
- R1 is monitoring its routing table for the presence of the 22.7.22.0/24 network. The default route is ONLY sent if this other route exists.





iBGP Scaling with Route Reflectors

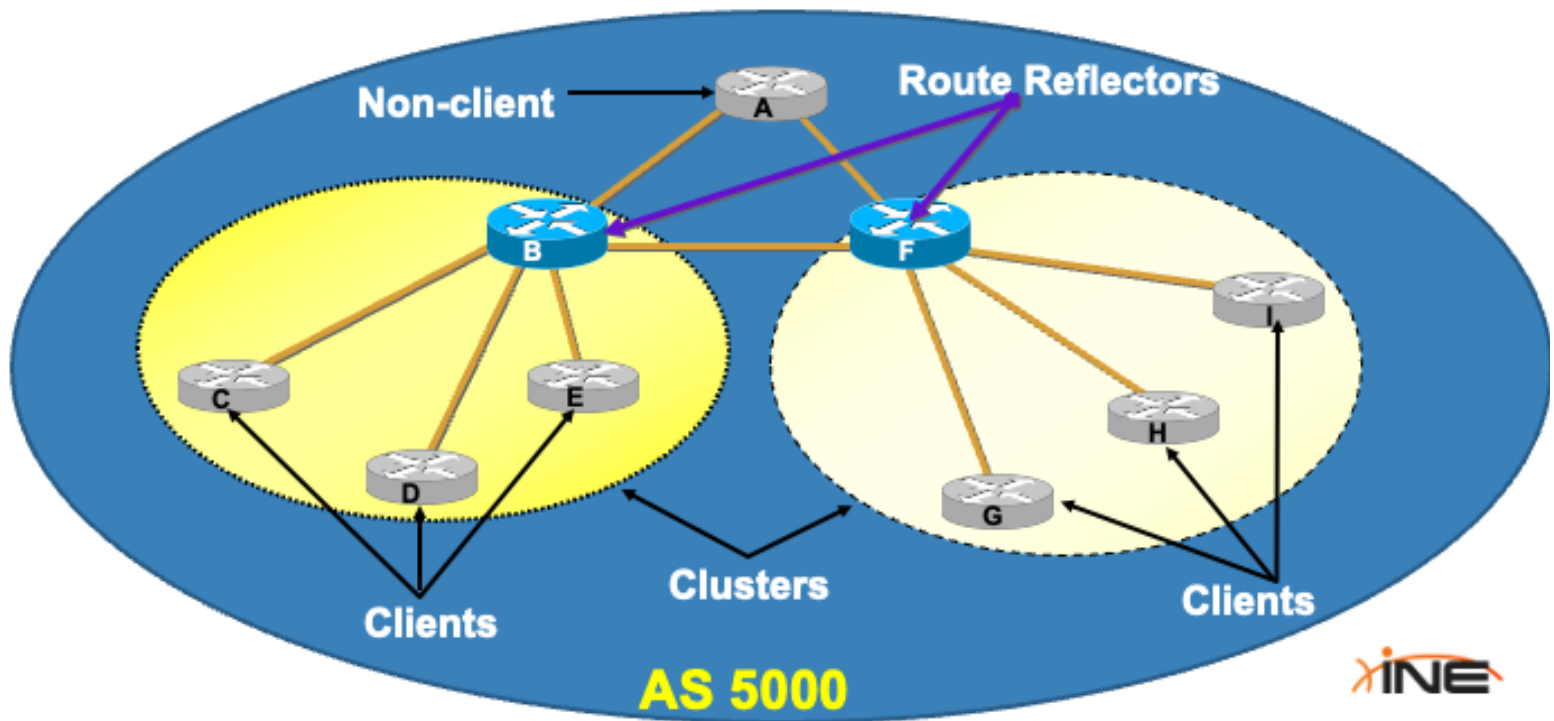


What Problem Is Solved?

- + In order to prevent iBGP Loops:
 - + iBGP does not modify AS-Path when receiving/transmitting iBGP updates.
 - + iBGP updates received from iBGP peers *are NOT propagated to other iBGP peers*.
- + Full-Mesh typically required of all iBGP peers to ensure convergence.
- + Full-Mesh not scalable in large iBGP deployments.
- + Route-Reflectors can “reflect” received iBGP updates to other iBGP peers...avoiding the need for a full-mesh.



Route Reflectors—Terminology



- Yellow lines represent iBGP peering sessions.
- -
- Only the Route-Reflectors require any special BGP configuration.

Route Reflectors – Loop Avoidance

+Originator_ID attribute

- + Carries the RID of the originator of the route in the local AS (created by the RR)

+Cluster_list attribute

- + The local cluster-id is added when the update is sent to clients (added by the RR).
- + Cluster-ID used by RR to detect loops
- + Default is to use the Router-id
- + *bgp cluster-id x.x.x.x*

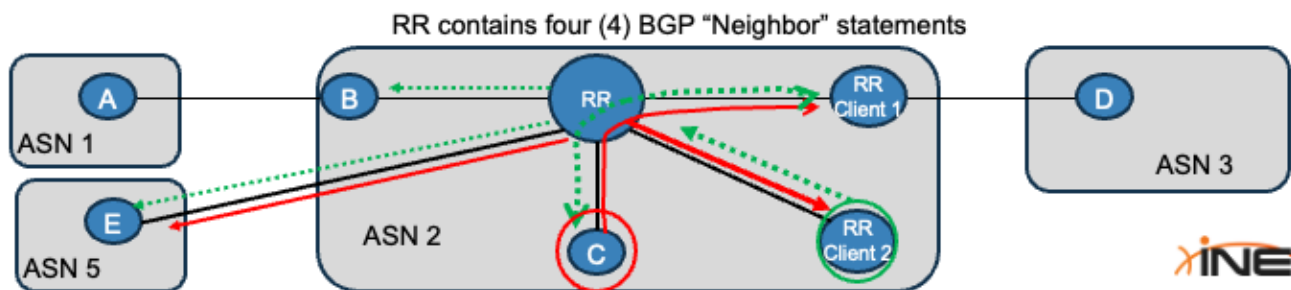


- Many documents state that when two-or-more RRs are servicing the same set of clients you should place them all into the same cluster (instead of having overlapping clusters). You do so by manually configuring the cluster-ID of both RRs to be the same...so they won't accept routes from each other (if peered together).
- -
- The above assumes that all clients within the cluster have iBGP peering to BOTH RRs.

Route Reflectors – Reflection Decisions

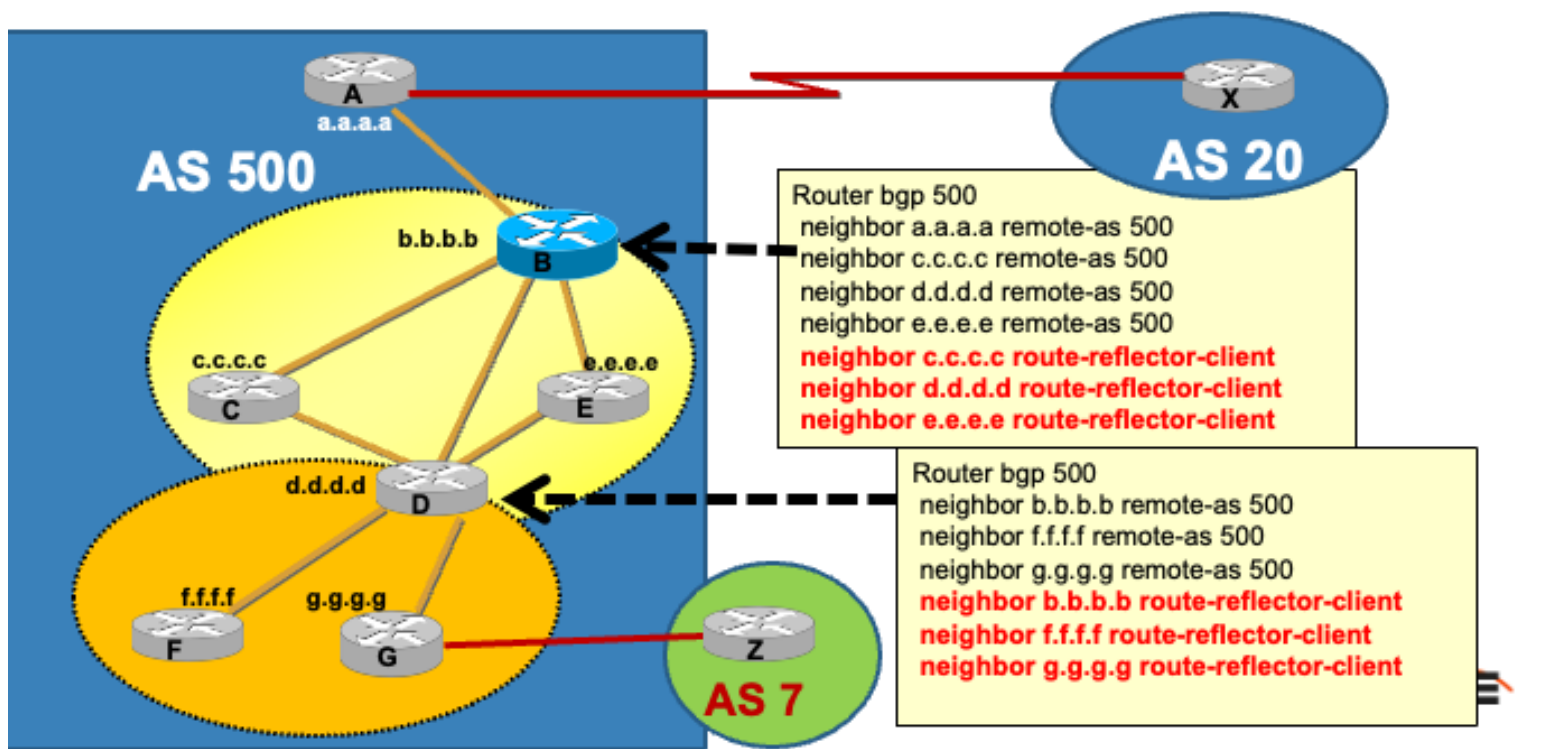
+ Once the best path is selected:

- + From non-client iBGP peer → reflect to all clients + normal eBGP propagation
- + From client iBGP peer → reflect to all non-clients & RR clients + normal eBGP propagation
- + From eBGP peer → normal iBGP and eBGP propagation



- Route reflectors do not add any Originator-ID or Cluster_ID when propagating a prefix they learned from an eBGP peer.

Route Reflectors—Configuration



Route Reflector Caveats

- +The “set” clause for outbound route-maps does not affect routes reflected to iBGP peers
- +The *nexthop-self* command will only affect the next-hop of eBGP learned routes (the next-hop of reflected routes should not be changed)



Confirming Reflection

```
R5#show ip bgp 22.22.22.0/24
BGP routing table entry for 22.22.22.0/24, version 0
Paths: (1 available, no best path)
Flag: 0x4100
  Not advertised to any peer
  Refresh Epoch 1
  Local
    1.2.1.2 (inaccessible) from 4.5.4.4 (1.4.1.4)
      Origin IGP, metric 0, localpref 100, valid, internal
      Originator: 2.3.2.2, Cluster list: 1.4.1.4, 1.2.1.1
      rx pathid: 0, tx pathid: 0
R5#
```



- In this example, the route had to go through two Route Reflectors before R5 received it.
- In the Cluster-List, the first entry (left-to-right) is the route-reflector that reflected the route to your local router. An entry after that (1.2.1.1) is another RR that reflected the route to YOUR RR.

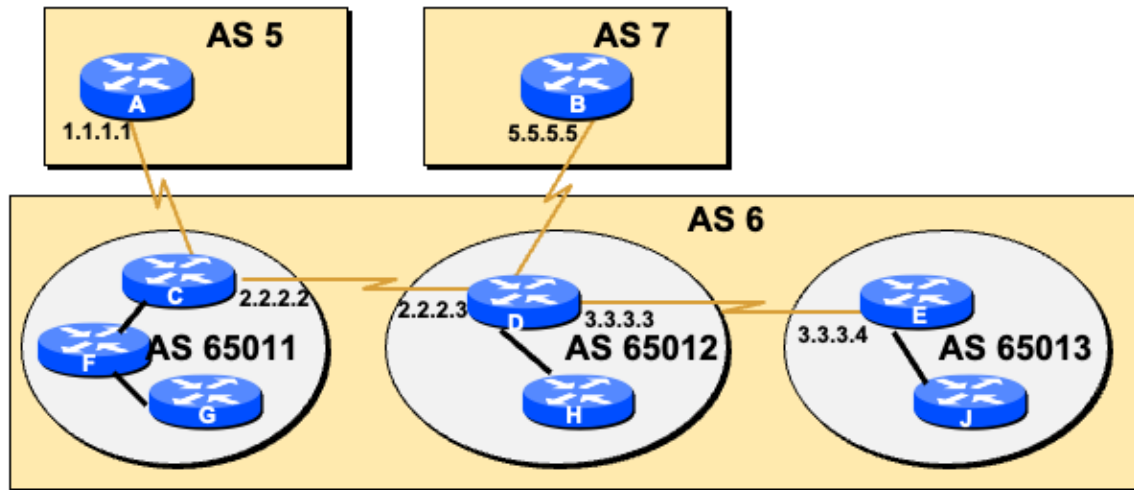




iBGP Scaling with Confederations



Confederations



- The idea is to break up the AS into mini sub-autonomous-systems

Confederations

- + Solves iBGP mesh problem
- + Divide the AS into sub-AS's
 - + Recommended to use private AS#s for Sub AS's
- + Visible to outside world as single AS
- + Preserve local preference, MED, and NEXT_HOP
- + iBGP speakers within a sub-AS are fully meshed
- + Route-reflectors can be used within a Sub AS



BGP within Confederations

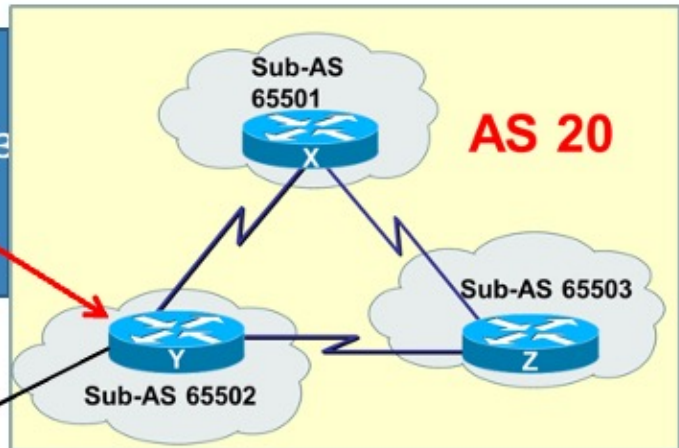
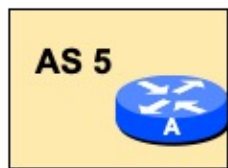
- +eBGP used **between** sub-AS's
- +iBGP used **within** sub-AS's
- +Next_hop from remote AS is preserved as update is passed between Sub-AS's
- +AS_Path attribute changed by adding:
 - +AS_Confederation_Set attribute (sub-AS listing in an unordered manner; used for aggregated routes).
 - +AS_Confederation_Sequence (sub-AS listing in sequential order; used for regular non-aggregated prefixes)
- +Both of the above will be removed from the AS-Path before sending an update to another "real" external AS.



- AS_Confed_Set and AS_Confed_Seq not counted AT ALL in bestpath algorithm!!
- -
- Confed-Seq denoted by parenthesis (xxx) whereas Confed-Set denoted by brackets [xxx]
- All routes received from confederation peers (whether external or internal peers) are considered as iBGP-learned routes!!

Confederations - Configuration

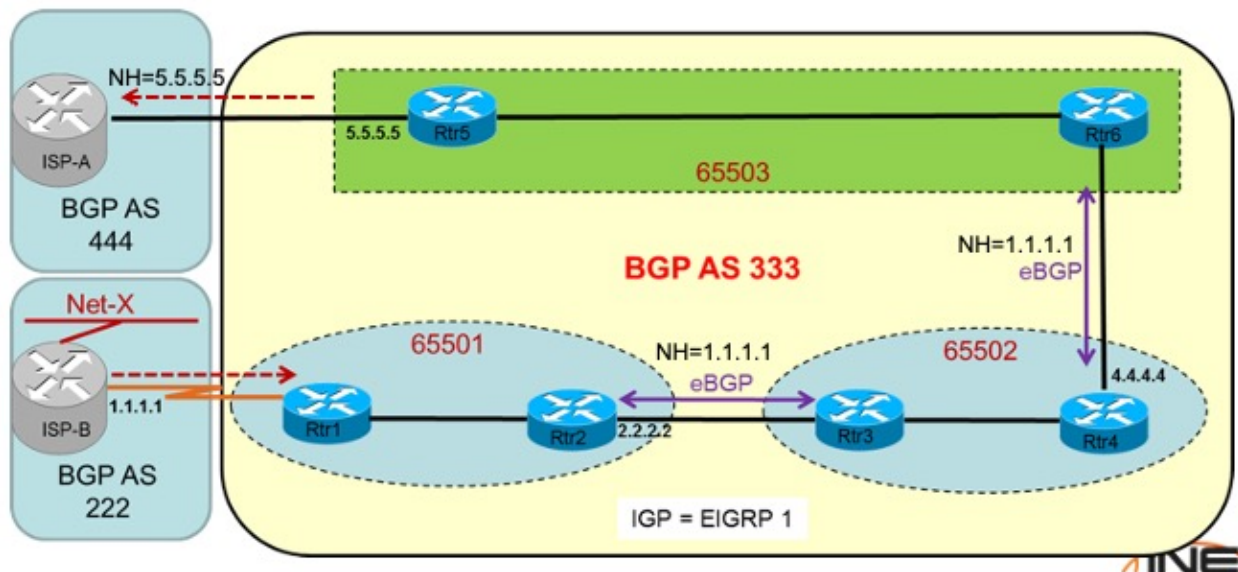
```
router bgp 65502
  bgp confederation identifier 20
  bgp confederation peers 65501 65503
  neighbor x.x.x.x remote-as 65501
  neighbor z.z.z.z remote-as 65503
  neighbor a.a.a.a remote-as 5
```



[illegible]

- Confederations add (by default) Confederation-AS-Seq to the FRONT of the AS-Path.
- -
- Similar to regular AS-Path...in (x y z) “Z” is the first confederation the route passed through and “X” is our neighbors Confederation-ID in a Confed-AS-Seq.
- -

Confederations and Next-Hop



- Confederations **do not change** the next-hop when transmitting to external Confederation peers.
- -
- Next-hop-self CAN be used to change the next-hop if desired.

Confederations in "show ip bgp"

```
Sw-1#sho ip bgp
BGP table version is 42, local router ID is 33.33.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop           Metric LocPrf Weight Path
r> 1.2.1.0/24      1.2.1.254             0      100      0 (65502 65501) 222 7 99 2033 ?
r i 1.2.1.0/24      1.2.1.254             0      100      0 (65501) 222 7 99 2033 ?
*> 1.2.2.0/24      1.2.1.254             0      100      0 (65502 65501) 222 7 99 2033 ?
* i 1.2.2.0/24      1.2.1.254             0      100      0 (65501) 222 7 99 2033 ?
r> 44.44.44.0/24   33.33.4.4              0      100      0 (65502) i
*> 192.168.1.4/30  1.2.1.254             0      100      0 (65502 65501) 222 7 99 2033 i
* i 192.168.1.4/30 1.2.1.254             0      100      0 (65501) 222 7 99 2033 i
```

AS_CONFED_SEQ



Route Propagation Decisions

+ Same as with “normal” BGP:

- + From peer in same sub-AS → only to external peers
- + From external peers → to all neighbors

+ “External peers” refers to

- + Peers outside the confederation
- + Peers in a different sub-AS
- + Preserve LOCAL_PREF, MED and NEXT_HOP



