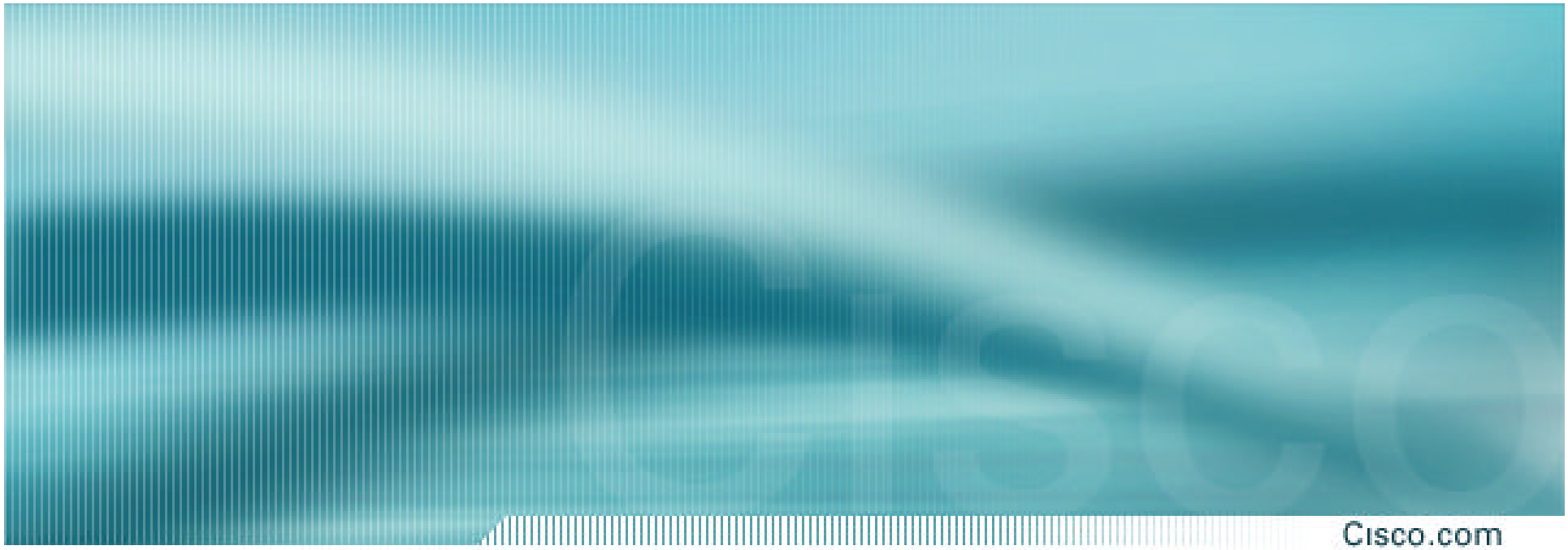


Troubleshooting BGP

Before We Begin…

- **My assumptions**
  - **Operational experience with BGP**
  - **Intermediate to advanced knowledge of the protocol**
- **What can you expect to get from this presentation?**
  - **Learn how to use show commands and debugs to troubleshoot BGP problems**
  - **Go through various real world examples**

Agenda

- **Peer Establishment**
- **Missing Routes**
- **Inconsistent Route Selection**
- **Loops and Convergence Issues**

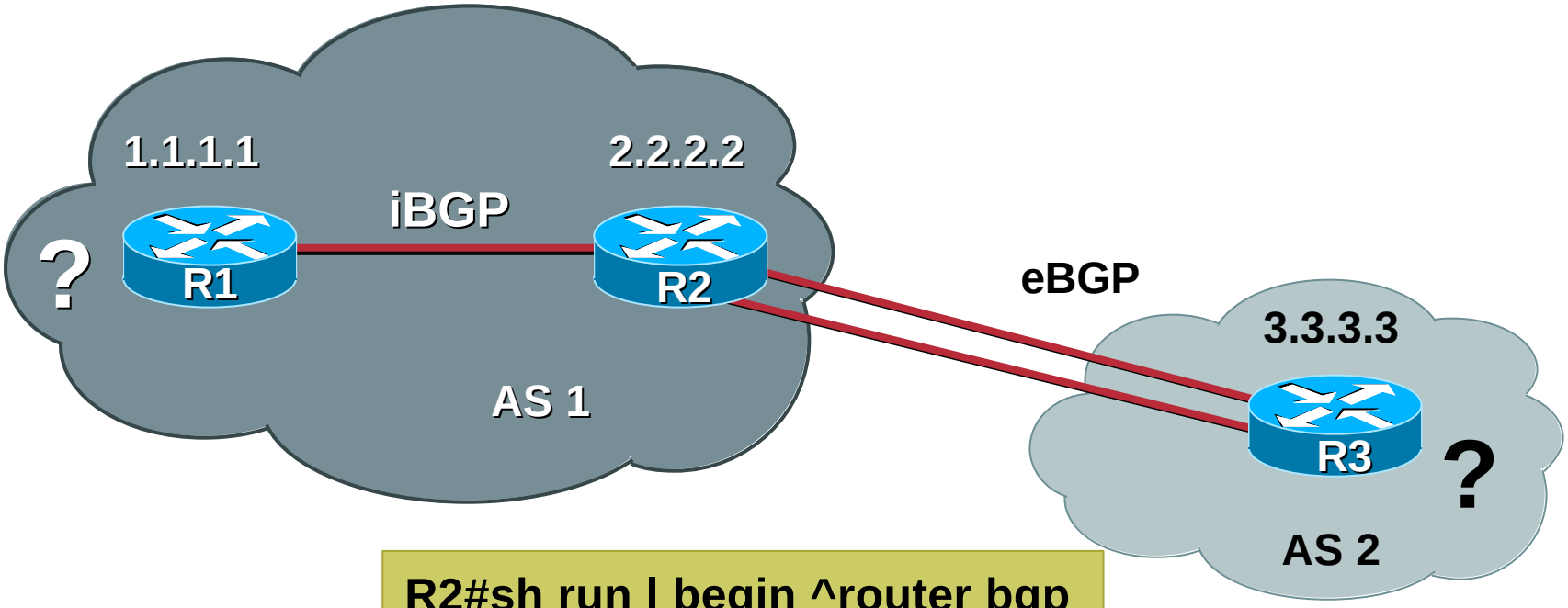
Peer Establishment

- **Routers establish a TCP session**
  - **Port 179—Permit in ACLs**
  - **IP connectivity (route from IGP)**
- **OPEN messages are exchanged**
  - **Peering addresses must match the TCP session**
  - **Local AS configuration parameters**

Common Problems

- **Sessions are not established**
  - **No IP reachability**
  - **Incorrect configuration**
- **Peers are flapping**
  - **Layer 2 problems**

Peer Establishment—Diagram




```
R2#sh run | begin ^router bgp
router bgp 1
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 1
 neighbor 3.3.3.3 remote-as 2
```

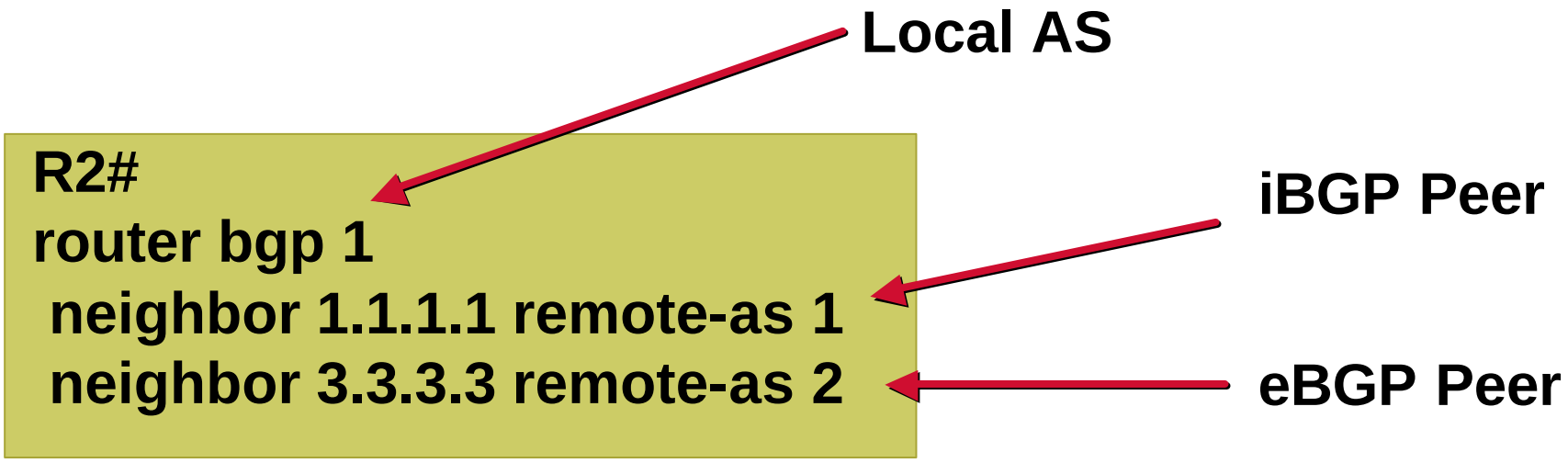
Peer Establishment—Symptoms


```
R2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 1
BGP table version is 1, main routing table version 1
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State
1.1.1.1         4     1       0       0        0    0    0 never    Active
3.3.3.3         4     2       0       0        0    0    0 never    Idle
```

- **Both peers are having problems**
  - **State may change between Active, Idle and Connect**

Peer Establishment

- **Is the Local AS configured correctly?**
- **Is the remote-as assigned correctly?**
- **Verify with your diagram or other documentation!**



Peer Establishment—iBGP

- Assume that IP connectivity has been checked
- Check TCP to find out what connections we are accepting

```
R2#show tcp brief all
TCB          Local Address          Foreign Address        (state)
005F2934   *.179                  3.3.3.3.*               LISTEN
0063F3D4  *.179                  1.1.1.1.*               LISTEN
```

We Are Listening for TCP Connections for Port 179 for the Configured Peering Addresses Only!

```
R2#debug ip tcp transactions
TCP special event debugging is on
R2#
TCP: sending RST, seq 0, ack 2500483296
TCP: sent RST to 4.4.4.4:26385 from 2.2.2.2:179
```

Remote Is Trying to Open the Session from 4.4.4.4 Address…

Peer Establishment—iBGP

What about Us?

```
R2#debug ip bgp
BGP debugging is on
R2#
BGP: 1.1.1.1 open active, local address 4.4.4.5
BGP: 1.1.1.1 open failed: Connection refused by remote host
```

We Are Trying to Open the Session from 4.4.4.5 Address…

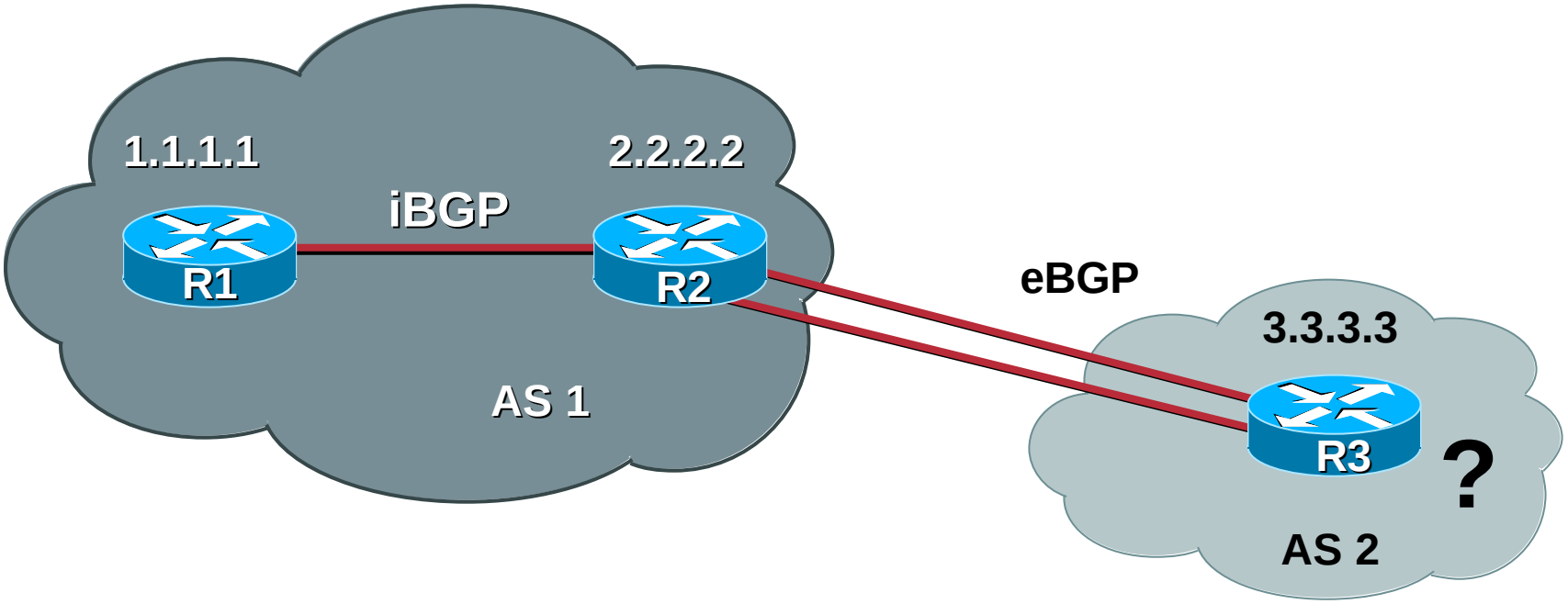
```
R2#sh ip route 1.1.1.1
Routing entry for 1.1.1.1/32
  Known via "static", distance 1, metric 0 (connected)
  * directly connected, via Serial1
      Route metric is 0, traffic share count is 1

R2#show ip interface brief | include Serial1
Serial1                4.4.4.5        YES manual   up     up
```

Peer Establishment—iBGP

- Source address is the outgoing interface towards the destination but peering in this case is using loopback interfaces!
- Force both routers to source from the correct interface
- Use “update-source” to specify the loopback when loopback peering

```
R2#
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 3.3.3.3 remote-as 2
 neighbor 3.3.3.3 update-source Loopback0
```

Peer Establishment—Diagram



- **R1 is established now**
- **The eBGP session is still having trouble!**

Peer Establishment—eBGP

- Trying to load-balance over multiple links to the eBGP peer
- Verify IP connectivity
  - Check the routing table
  - Use ping/trace to verify two way reachability

```
R2#ping 3.3.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

- Routing towards destination correct, but…

Peer Establishment—eBGP


```
R2#ping ip
Target IP address: 3.3.3.3
Extended commands [n]: y
Source address or interface: 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

- Use extended pings to test loopback to loopback connectivity
- R3 does not have a route to our loopback, 2.2.2.2

Peer Establishment—eBGP

- Assume R3 added a route to 2.2.2.2
- Still having problems…

```
R2#sh ip bgp neigh 3.3.3.3
BGP neighbor is 3.3.3.3,  remote AS 2, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:04, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0
  Index 2, Offset 0, Mask 0x4
  0 accepted prefixes consume 0 bytes
  Prefix advertised 0, suppressed 0, withdrawn 0
  Connections established 0; dropped 0
  Last reset never
  External BGP neighbor not directly connected.
  No active TCP connection
```

Peer Establishment—eBGP


```
R2#
router bgp 1
 neighbor 3.3.3.3 remote-as 2
 neighbor 3.3.3.3 ebgp-multihop 255
 neighbor 3.3.3.3 update-source Loopback0
```

- eBGP peers are normally directly connected
  - By default, TTL is set to 1 for eBGP peers
  - If not directly connected, specify ebgp-multihop
- At this point, the session should come up

Peer Establishment—eBGP


```
R2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 1

Neighbor   V  AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3    4   2      10      26       0    0    0 never    Active
```

- **Still having trouble!**

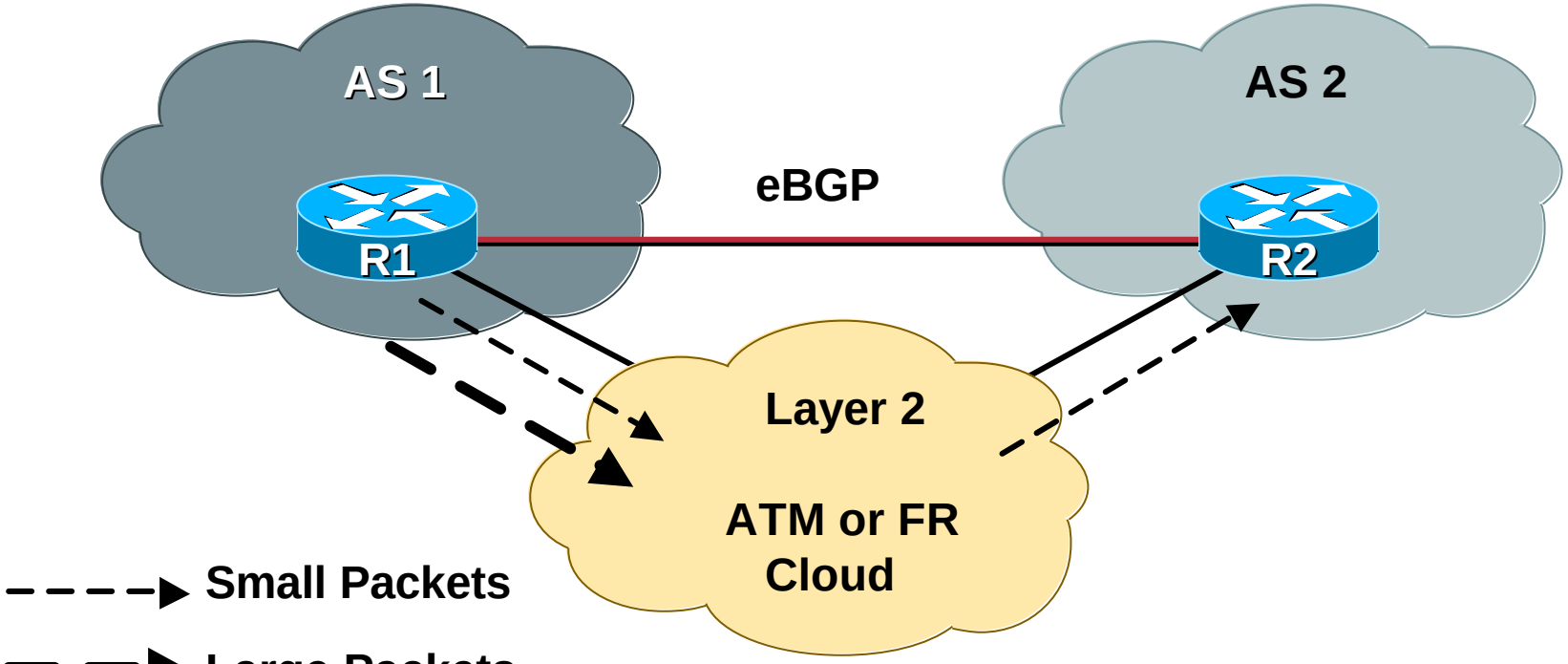
  Connectivity issues have already been checked and corrected

Peer Establishment—eBGP


```
R2#debug ip bgp events
14:06:37: BGP: 3.3.3.3 open active, local address 2.2.2.2
14:06:37: BGP: 3.3.3.3 went from Active to OpenSent
14:06:37: BGP: 3.3.3.3 sending OPEN, version 4
14:06:37: BGP: 3.3.3.3 received NOTIFICATION 2/2
        (peer in wrong AS) 2 bytes 0001
14:06:37: BGP: 3.3.3.3 remote close, state CLOSEWAIT
14:06:37: BGP: service reset requests
14:06:37: BGP: 3.3.3.3 went from OpenSent to Idle
14:06:37: BGP: 3.3.3.3 closing
```

- If an error is detected, a notification is sent and the session is closed
- R3 is configured incorrectly
  - Has “neighbor 2.2.2.2 remote-as 10”
  - Should have “neighbor 2.2.2.2 remote-as 1”
- After R3 makes this correction the session comes up

Flapping Peer—Diagram



- Small packets are ok
- Large packets are lost in the cloud
- BGP session flaps

Flapping Peer

- Enable “bgp log-neighbor-changes” so you get a log message when a peer flaps
- R1 and R2 are peering over ATM cloud

```
R2#

%BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down BGP
  Notification sent

%BGP-3-NOTIFICATION: sent to neighbor 1.1.1.1 4/0
  (hold time expired) 0 bytes

R2#show ip bgp neighbor 1.1.1.1 | include Last reset

 Last reset 00:01:02, due to BGP Notification sent,
  hold time expired
```

  - We are not receiving keepalives from the other side!

Flapping Peer

- Let's take a look at our peer!

```
R1#show ip bgp sum
BGP router identifier 172.16.175.53, local AS number 1
BGP table version is 10167, main routing table version 10167
10166 network entries and 10166 paths using 1352078 bytes of memory
1 BGP path attribute entries using 60 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 10166/300 prefixes, 10166/0 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2         4     2      53     284    10167    0    97 00:02:15        0
```

```
R1#show ip bgp summary | begin Neighbor
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2         4     2      53     284    10167    0    98 00:03:04        0
```

- Hellos are stuck in OutQ behind update packets!
- Notice that the MsgSent counter has not moved

Flapping Peer


```
R1#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/21/24 m
```

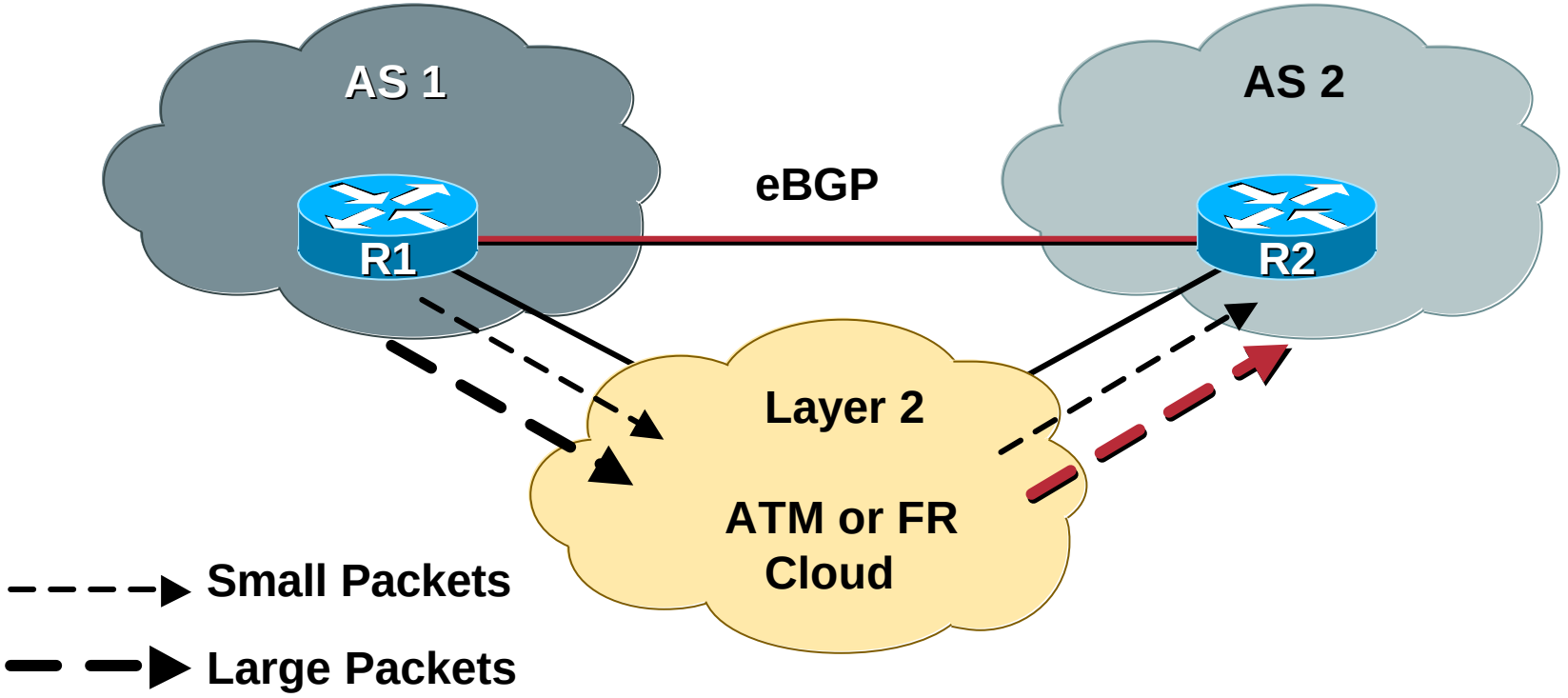
```
R1#ping ip
Target IP address: 2.2.2.2
Repeat count [5]:
Datagram size [100]: 1500
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

- **Normal pings work but a ping of 1500 fails?**

Flapping Peer

- **Things to check**
  - **MTU values**
  - **Traffic shaping**
  - **Rate-limiting parameters**
- **Looks like a Layer 2 problem**
- **At this point we have verified that BGP is not at fault**
- **Next step is to troubleshoot layer 2…**

Flapping Peer—Diagram



- Large packets are ok now
- BGP session is stable!

Troubleshooting Tips

- **Extended ping/traceroute allow you to verify**
  - **Loopback to loopback IP connectivity**
  - **TTL issues**
- **“show ip bgp summary”**
  - **Displays the state of all peers**
- **“show ip bgp neighbor”**
  - **Gives a lot of information regarding the peer**

Troubleshooting Tips

- **“debug ip bgp”**

  Should give you a good hint as to why a peer will not establish

- **“debug ip bgp events”**

  Displays state transitions for peers

- **“show ip bgp neighbor | include Last reset”**

  Will show you the last reset reason for all peers

Agenda

- Peer Establishment
- Missing Routes
- Inconsistent Route Selection
- Loops and Convergence Issues

Quick Review

- **Once the session has been established, UPDATEs are exchanged**
  - **All the locally known routes**
  - **Only the bestpath is advertised**
- **Incremental UPDATE messages are exchanged afterwards**

Quick Review

- **Bestpath received from eBGP peer**
  - **Advertise to all peers**
- **Bestpath received from iBGP peer**
  - **Advertise only to eBGP peers**
  - **A full iBGP mesh must exist**

Missing Routes—Agenda

- Route Origination
- UPDATE Exchange
- Filtering

Route Origination—Example I

- *All examples are with “auto-summary” enabled
- Basic network statement

```
R1# show run | begin bgp
 network 6.0.0.0
```

- BGP is not originating the route???

```
R1# show ip bgp | include 6.0.0.0
R1#
```

- Do we have a component route?

```
R1# show ip route 6.0.0.0 255.0.0.0 longer
R1#
```

Route Origination—Example I

- As soon as the RIB has a component route

```
R1# show ip route 6.0.0.0 255.0.0.0 longer
        6.0.0.0/32 is subnetted, 1 subnets
S       6.6.6.6 [1/0] via 20.100.1.6
```

- Bingo, BGP originates the route!!

```
R1# show ip bgp | include 6.0.0.0
*> 6.0.0.0  0.0.0.0              0     32768 i
```

Route Origination—Example II

- Network statement with mask

```
R1# show run | include 200.200.0.0
 network 200.200.0.0 mask 255.255.252.0
```

- BGP is not originating the route???

```
R1# show ip bgp | include 200.200.0.0
R1#
```

- Do we have the exact route?

```
R1# show ip route 200.200.0.0 255.255.252.0
% Network not in table
```

Route Origination—Example II

- Nail down routes you want to originate

```
ip route 200.200.0.0 255.255.252.0 Null 0  254
```

- Check the RIB

```
R1# show ip route 200.200.0.0 255.255.252.0
       200.200.0.0/22 is subnetted, 1 subnets
S      200.200.0.0 [1/0] via Null 0
```

- BGP originates the route!!

```
R1# show ip bgp | include 200.200.0.0
*> 200.200.0.0/220.0.0.0          0    32768
```

Route Origination—Example III

- **Trying to originate an aggregate route**

```
aggregate-address 7.7.0.0 255.255.0.0 summary-only
```

- **The RIB has a component but BGP does not create the aggregate???**

```
R1# show ip route 7.7.0.0 255.255.0.0 longer
      7.0.0.0/32 is subnetted, 1 subnets
C      7.7.7.7 [1/0] is directly connected, Loopback 0
```

```
R1# show ip bgp | i 7.7.0.0
R1#
```

Route Origination—Example III

- Remember, to have a BGP aggregate you need a BGP component, not a RIB (Routing Information Base, a.k.a. the routing table) component

```
R1# show ip bgp 7.7.0.0 255.255.0.0 longer
R1#
```

- Once BGP has a component route we originate the aggregate

```
network 7.7.7.7 mask 255.255.255.255


R1# show ip bgp 7.7.0.0 255.255.0.0 longer
*> 7.7.0.0/16  0.0.0.0                  32768 i
s> 7.7.7.7/32  0.0.0.0       0          32768 i
```

- s means this component is suppressed due to the "summary-only" argument

Troubleshooting Tips

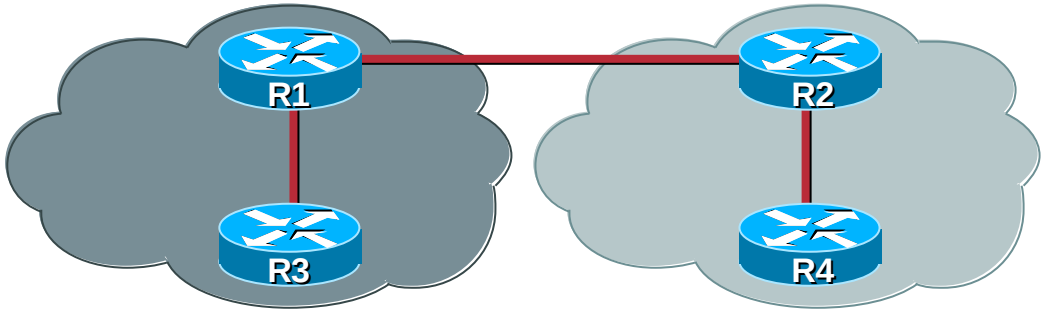
- "auto-summary" rules [default]
  - Network statement—must have component route (RIB)
  - Network/Mask statement—must have exact route (RIB)
- "no auto-summary" rules
  - Always need an exact route (RIB)
- aggregate-address looks in the BGP table, not the RIB
- "show ip route x.x.x.x y.y.y.y longer"
  - Great for finding RIB component routes
- "show ip bgp x.x.x.x y.y.y.y longer"
  - Great for finding BGP component routes

Missing Routes

- **Route Origination**
- **UPDATE Exchange**
- **Filtering**

Missing Routes—Example I

- Two RR clusters
- R1 is a RR for R3
- R2 is a RR for R4
- R4 is advertising 7.0.0.0/8
- R2 has the route but R1 and R3 do not?



Missing Routes—Example I

- **First, did R2 advertise the route to R1?**

```
R2# show ip bgp neighbors 1.1.1.1 advertised-routes
BGP table version is 2, local router ID is 2.2.2.2
   Network          Next Hop            Metric LocPrf Weight Path
*>i7.0.0.0          4.4.4.4                  0    100      0 I
```

- **Did R1 receive it?**

```
R1# show ip bgp neighbors 2.2.2.2 routes
Total number of prefixes 0
```

Missing Routes—Example I

- Time to debug!!

```
access-list 100 permit ip host 7.0.0.0 host 255.0.0.0
R1# debug ip bgp update 100
```

- Tell R2 to resend his UPDATEs

```
R2# clear ip bgp 1.1.1.1 soft out
```

- R1 shows us something interesting

```
*Mar  1 21:50:12.410: BGP(0): 2.2.2.2 rcv UPDATE w/ attr:
nexthop 4.4.4.4, origin i, localpref 100, metric 0,
originator 100.1.1.1, clusterlist 2.2.2.2, path , community
, extended community
*Mar  1 21:50:12.410: BGP(0): 2.2.2.2 rcv UPDATE about
7.0.0.0/8 -- DENIED due to: ORIGINATOR is us;
```

- Cannot accept an update with our Router-ID as the ORIGINATOR_ID. Another means of loop detection in BGP

Missing Routes—Example I

- R1 and R4 have the same Router-ID

```
R1# show ip bgp summary | include identifier.
BGP router identifier 100.1.1.1, local AS number 100.
```

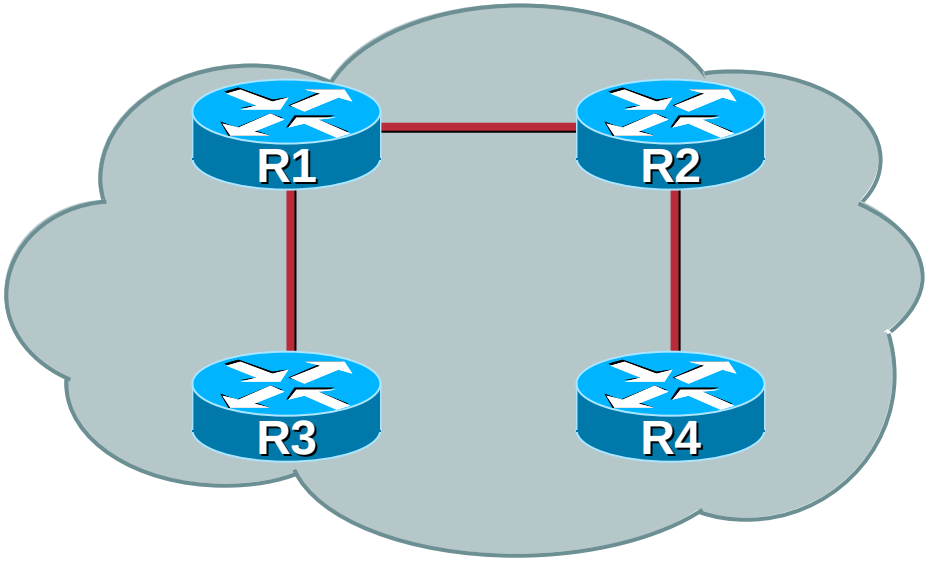
```
R4# show ip bgp summary | include identifier.
BGP router identifier 100.1.1.1, local AS number 100.
```

- Can be a problem in multicast networks; for RP (Rendezvous Point) purposes the same address may be assigned to multiple routers
- Specify a unique Router-ID

```
R1#show run | include router-id
 bgp router-id 1.1.1.1
R4# show run | include router-id
 bgp router-id 4.4.4.4
```

Missing Routes—Example II

- One RR cluster
- R1 and R2 are RRs
- R3 and R4 are RRCs
- R4 is advertising 7.0.0.0/8
  - R2 has it
  - R1 and R3 do not




```
R1#show run | include cluster
 bgp cluster-id 10
R2#show run | include cluster
 bgp cluster-id 10
```

Missing Routes—Example II

- Same steps as last time!
- Did R2 advertise it to R1?

```
R2# show ip bgp neighbors 1.1.1.1 advertised-routes
BGP table version is 2, local router ID is 2.2.2.2
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*>i7.0.0.0          4.4.4.4                  0    100      0 i
```

- Did R1 receive it?

```
R1# show ip bgp neighbor 2.2.2.2 routes
Total number of prefixes 0
```

Missing Routes—Example II

- Time to debug!!

```
access-list 100 permit ip host 7.0.0.0 host 255.0.0.0

R1# debug ip bgp update 100
```

- Tell R2 to resend his UPDATEs

```
R2# clear ip bgp 1.1.1.1 soft out
```

- R1 shows us something interesting

```
*Mar  3 14:28:57.208: BGP(0): 2.2.2.2 rcv UPDATE w/ attr: nexthop
4.4.4.4, origin i, localpref 100, metric 0, originator 4.4.4.4,
clusterlist 0.0.0.10, path , community , extended community

*Mar  3 14:28:57.208: BGP(0): 2.2.2.2 rcv UPDATE about 7.0.0.0/8 --
DENIED due to: reflected from the same cluster;
```

- Remember, all RRCs must peer with all RRs in a cluster; allows R4 to send the update directly to R1

Troubleshooting Tips

- **"show ip bgp neighbor x.x.x.x advertised-routes"**

  Lets you see a list of NLRI that you sent a peer

  Note: The attribute values shown are taken from the BGP table; attribute modifications by outbound route-maps will not be shown

- **"show ip bgp neighbor x.x.x.x routes"**

  Displays routes x.x.x.x sent to us that made it through our inbound filters

- **"show ip bgp neighbor x.x.x.x received-routes"**

  Can only use if "soft-reconfig inbound" is configured

  Displays all routes received from a peer, even those that were denied

Troubleshooting Tips

- **“clear ip bgp x.x.x.x soft in”**
  - **Ask x.x.x.x to resend his UPDATEs to us**
- **“clear ip bgp x.x.x.x soft out”**
  - **Tells BGP to resend UPDATEs to x.x.x.x**
- **“debug ip bgp update”**
  - **Always use an ACL to limit output**
  - **Great for troubleshooting “Automatic Denies”**
- **“debug ip bgp x.x.x.x update”**
  - **Allows you to debug updates to/from a specific peer**
  - **Handy if multiple peers are sending you the same prefix**

Missing Routes

- Route Origination
- UPDATE Exchange
- Filtering

Update Filtering

- **Type of filters**
  - **Prefix filters**
  - **AS_PATH filters**
  - **Community filters**
  - **Route-maps**
- **Applied incoming and/or outgoing**

Missing Routes—Update Filters

- **Determine which filters are applied to the BGP session**

  show ip bgp neighbors x.x.x.x

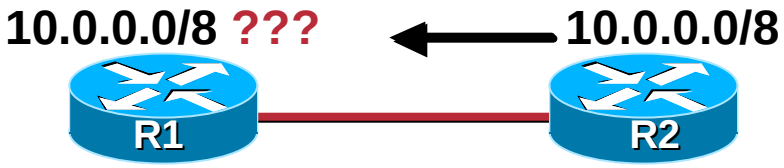
  show run | include neighbor x.x.x.x

- **Examine the route and pick out the relevant attributes**

  show ip bgp x.x.x.x

- **Compare the attributes against the filters**

Missing Routes—Update Filters



- Missing 10.0.0.0/8 in R1 (1.1.1.1)
- Not received from R2 (2.2.2.2)

```
R1#show ip bgp neigh 2.2.2.2 routes

Total number of prefixes 0
```

Missing Routes—Update Filters

- R2 originates the route
- Does not advertise it to R1

```
R2#show ip bgp neigh 1.1.1.1 advertised-routes
     Network          Next Hop            Metric LocPrf Weight Path
```

```
R2#show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 1660
Paths: (1 available, best #1)
  Not advertised to any peer
  Local
    0.0.0.0 from 0.0.0.0 (2.2.2.2)
    Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local, best
```

Missing Routes—Update Filters

- Time to check filters!
- ^ matches the beginning of a line
- $ matches the end of a line
- ^$ means match any empty AS_PATH
- Filter “looks” correct

```
R2#show run | include neighbor 1.1.1.1
 neighbor 1.1.1.1 remote-as 3
 neighbor 1.1.1.1 filter-list 1 out

R2#sh ip as-path 1
  AS path access list 1
    permit ^$
```

Missing Routes—Update Filters

Cisco.com

```
R2#show ip bgp filter-list 1

R2#show ip bgp regexp ^$
BGP table version is 1661, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.0         0.0.0.0                  0         32768 i
```

- Nothing matches the filter-list???
- Re-typing the regexp gives the expected output

Missing Routes—Update Filters

- **Copy and paste the entire regexp line from the configuration**

```
R2#show ip bgp regexp ^$

Nothing matches again! Let's use the up arrow key to see where the
cursor stops

R2#show ip bgp regexp ^$ 
```

End of Line Is at the Cursor

- **There is a trailing white space at the end**
- **It is considered part of the regular expression**

Missing Routes—Update Filters

- Force R2 to resend the update after the filter-list correction
- Then check R1 to see if he has the route

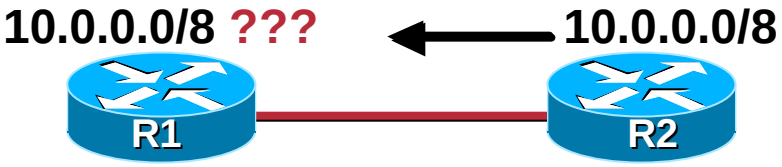
```
R2#clear ip bgp 1.1.1.1 soft out

R1#show ip bgp 10.0.0.0
% Network not in table
```

- R1 still does not have the route
- Time to check R1's inbound policy for R2

Missing Routes—Update Filters


```
R1#show run | include neighbor 2.2.2.2
 neighbor 2.2.2.2 remote-as 12
 neighbor 2.2.2.2 route-map POLICY in
R1#show route-map POLICY
route-map POLICY, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100 101
    as-path (as-path filter): 1
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
R1#show access-list 100
Extended IP access list 100
    permit ip host 10.0.0.0 host 255.255.0.0
R1#show access-list 101
Extended IP access list 101
    permit ip 200.1.0 0.0.0.255 host 255.255.255.0
R1#show ip as-path 1
AS path access list 1
    permit ^12$
```

Missing Routes—Update Filters



- **Confused? Let's run some debugs**

```
R1#show access-list 99
Standard IP access list 99
    permit 10.0.0.0

R1#debug ip bgp 2.2.2.2 update 99
BGP updates debugging is on for access list 99 for neighbor 2.2.2.2

R1#
4d00h: BGP(0): 2.2.2.2 rcvd UPDATE w/ attr: nexthop 2.2.2.2, origin i,
 metric 0, path 12
4d00h: BGP(0): 2.2.2.2 rcvd 10.0.0.0/8 -- DENIED due to: route-map;
```

Missing Routes—Update Filters


```
R1#sh run | include neighbor 2.2.2.2
 neighbor 2.2.2.2 remote-as 12
 neighbor 2.2.2.2 route-map POLICY in
R1#sh route-map POLICY
route-map POLICY, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100 101
    as-path (as-path filter): 1
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
R1#sh access-list 100
Extended IP access list 100
    permit ip host 10.0.0.0 host 255.255.0.0
R1#sh access-list 101
Extended IP access list 101
    permit ip 200.1.1.0 0.0.0.255 host 255.255.255.0
R1#sh ip as-path 1
AS path access list 1
    permit ^12$
```

Missing Routes—Update Filters

Cisco.com

- **Wrong mask! Needs to be /8 and the ACL allows a /16 only!**

  Extended IP access list 100

  permit ip host 10.0.0.0 host 255.255.0.0

- **Should be**

  Extended IP access list 100

  permit ip host 10.0.0.0 host 255.0.0.0

- **Use prefix-list instead, more difficult to make a mistake**

  ip prefix-list my_filter permit 10.0.0.0/8

- **What about ACL 101?**

  Multiple matches on the same line are ORed

  Multiple matches on different lines are ANDed

- **ACL 101 does not matter because ACL 100 matches which satisfies the OR condition**

Troubleshooting Tips

- **“show ip as-path-access-list”**

  Displays the filter

- **“show ip bgp filter-list”**

  Displays BGP paths that match the filter

- **“show ip bgp regexp”**

  Displays BGP paths that match the as-path regular expression; handy for troubleshooting filter-list issues

Troubleshooting Tips

- **“show ip community-list”**

  Displays the filter

- **“show ip bgp community-list”**

  Displays BGP paths that match the filter

- **“show ip prefix-list”**

  Displays the filter

  Prefix-list are generally easier to use than ACLs

- **“show ip bgp prefix-list”**

  Displays BGP paths that match the filter

Troubleshooting Tips

- **“show route-map”**
  - **Displays the filter**
- **“show ip bgp route-map”**
  - **Displays BGP paths that match the filter**
- **“show access-list”**
  - **Displays the filter**
- **debug ip bgp update ACL**
  - **After going through the config, debug!**
  - **Don’t forget the ACL**

Agenda

- Peer Establishment
- Missing Routes
- Inconsistent Route Selection
- Loops and Convergence Issues

Inconsistent Route Selection

- Two common problems with route selection
  - Inconsistency
  - Appearance of an incorrect decision
- RFC 1771 defines the decision algorithm
- Every vendor has tweaked the algorithm

  http://www.cisco.com/warp/public/459/25.shtml
- Route selection problems can result from oversights by RFC 1771

Inconsistent—Example I

- RFC says that MED is not always compared
- As a result, the ordering of the paths can effect the decision process
- By default, the prefixes are compared in order of arrival (most recent to oldest)

  Use bgp deterministic-med to order paths consistently

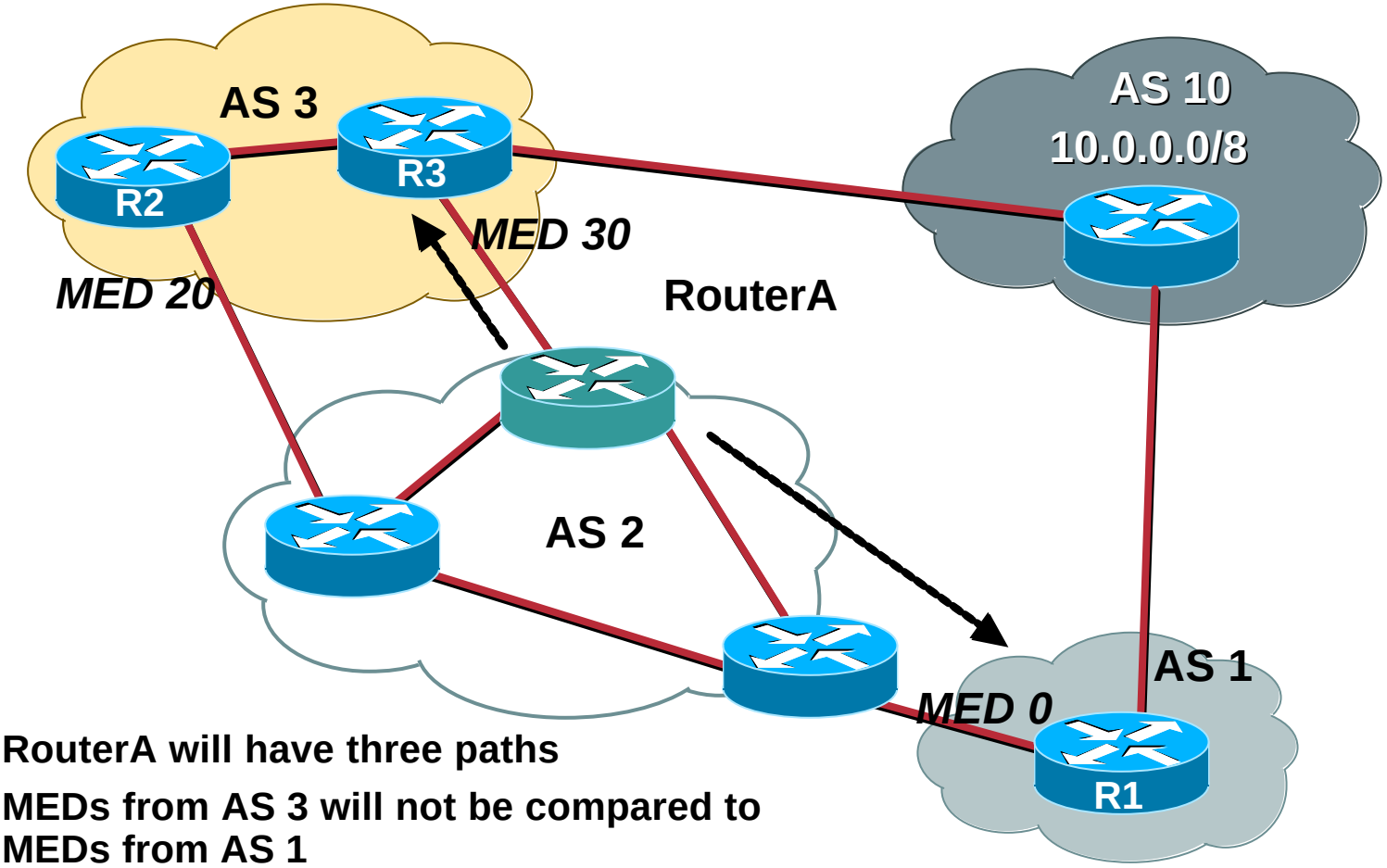
  The bestpath is recalculated as soon as the command is entered

  Enable in all the routers in the AS

Inconsistent—Example I

- **Inconsistent route selection may cause problems**
  - **Routing loops**
  - **Convergence loops—i.e. the protocol continuously sends updates in an attempt to converge**
  - **Changes in traffic patterns**
- **Difficult to catch and troubleshoot**
- **It is best to avoid the problem in the first place**
  - **bgp deterministic-med**

Symptom I—Diagram



- RouterA will have three paths
- MEDs from AS 3 will not be compared to MEDs from AS 1
- RouterA will sometimes select the path from R1 as best and but may also select the path from R3 as best

Inconsistent—Example I


```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #3, advertised over iBGP, eBGP)
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

- **Initial State**

  Path 1 beats Path 2—Lower MED

  Path 3 beats Path 1—Lower Router-ID

Inconsistent—Example I


```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #3, advertised over iBGP, eBGP)
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external, best
```

- **1.1.1.1 bounced so the paths are re-ordered**
  - **Path 1 beats Path 2—Lower Router-ID**
  - **Path 3 beats Path 1—External vs Internal**

Deterministic MED—Operation

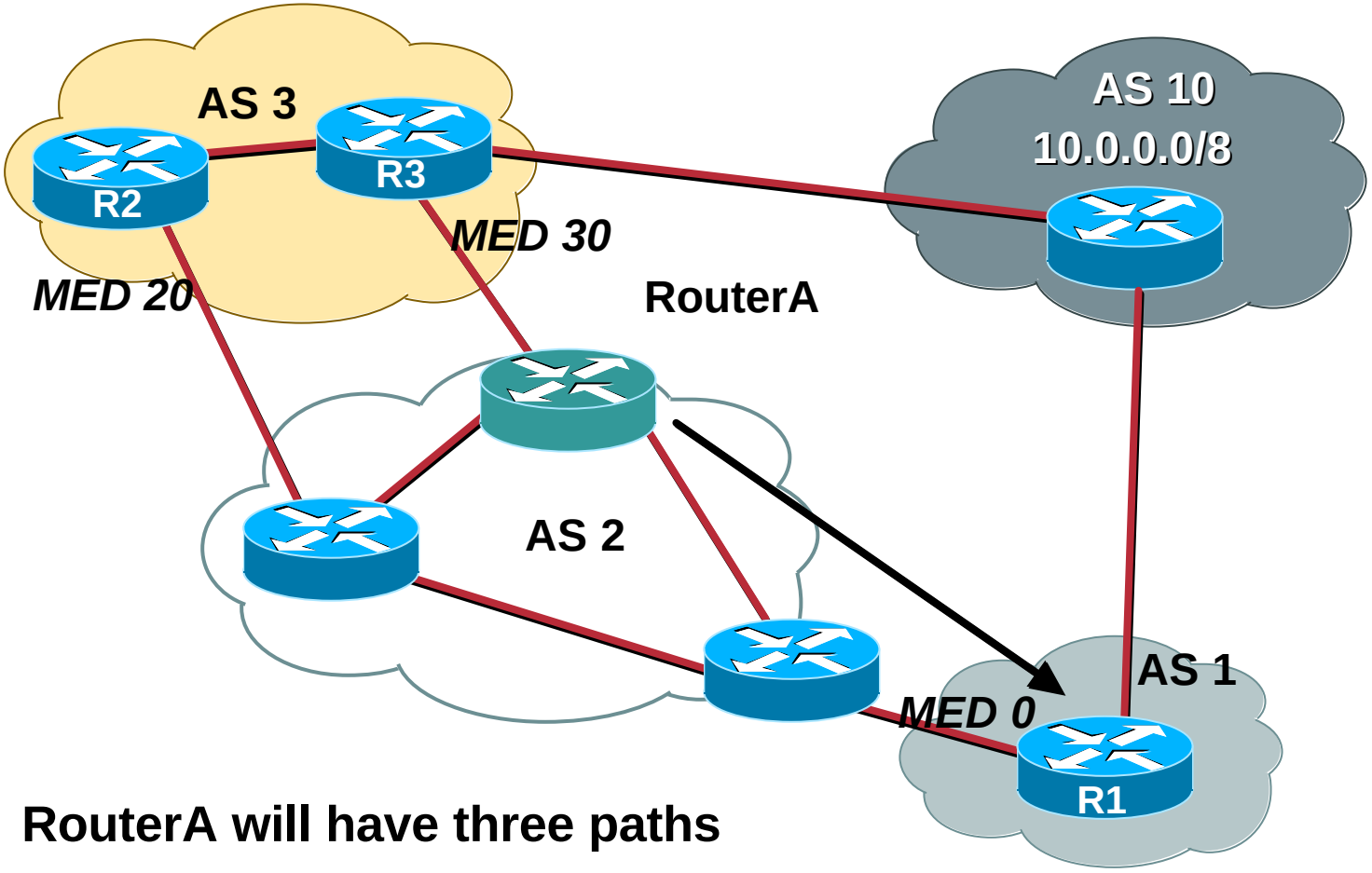
- The paths are ordered by Neighbor AS
- The bestpath for each Neighbor AS group is selected
- The overall bestpath results from comparing the winners from each group
- The bestpath will be consistent because paths will be placed in a deterministic order

Deterministic MED—Result


```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #1, advertised over iBGP, eBGP)
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external
```

- Path 1 is best for AS 1
- Path 2 beats Path 3 for AS 3—Lower MED
- Path 1 beats Path 2—Lower Router-ID

Solution—Diagram



- RouterA will have three paths
- RouterA will consistently select the path from R1 as best!

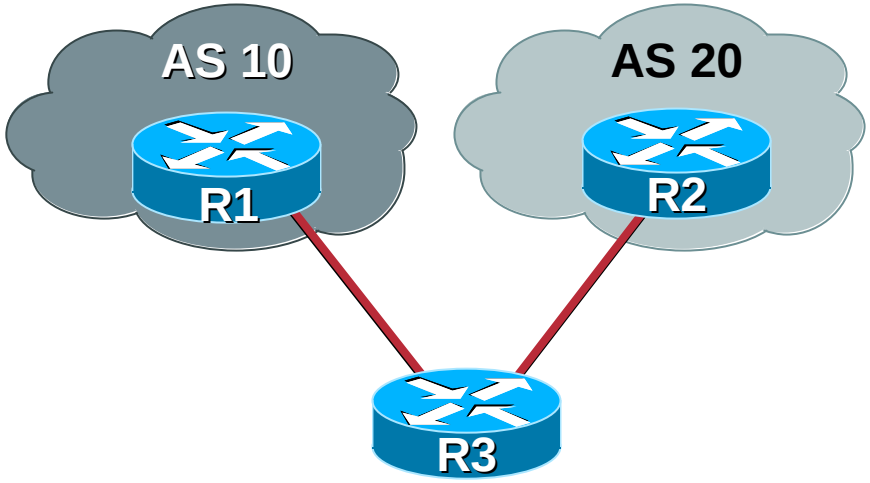
Deterministic MED—Summary

Cisco.com

- Always use “bgp deterministic-med”
- Need to enable throughout entire network at roughly the same time
- If only enabled on a portion of the network routing loops and/or convergence problems may become more severe
- As a result, default behavior cannot be changed so the knob must be configured by the user

Inconsistent—Example II

- The bestpath changes every time the peering is reset




```
R3#show ip bgp 7.0.0.0
BGP routing table entry for 7.0.0.0/8, version 15
  10 100
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, external
  20 100
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Inconsistent—Example II


```
R3#show ip bgp 7.0.0.0
BGP routing table entry for 7.0.0.0/8, version 17
Paths: (2 available, best #2)
  Not advertised to any peer
  20 100
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 0, localpref 100, valid, external
  10 100
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, external, best
```

- **The “oldest” external is the bestpath**
  - **All other attributes are the same**
  - **Stability enhancement!!—CSCdk12061—Integrated in 12.0(1)**
- **“bgp bestpath compare-router-id” will disable this enhancement—CSCdr47086—Integrated in 12.0(11)S and 12.1(3)**

Inconsistent—Example III


```
R1#sh ip bgp 11.0.0.0
BGP routing table entry for 11.0.0.0/8, version 10
  100
    1.1.1.1 from 1.1.1.1
      Origin IGP, localpref 120, valid, internal
  100
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 0, localpref 100, valid, external, best
```

- Path 1 has higher localpref but path 2 is better???
- This appears to be incorrect…

Inconsistent—Example III

- Path is from an internal peer which means the path must be synchronized by default
- Check to see if sync is on or off

```
R1# show run | include sync
R1#
```

- Sync is still enabled, check for IGP path:

```
R1# show ip route 11.0.0.0
% Network not in table
```

- CSCdr90728 “BGP: Paths are not marked as not synchronized”—Fixed in 12.1(4)
- Path 1 is not synchronized
- Router made the correct choice

Troubleshooting Tips

- **“show run | include sync”**

  Quick way to see if synchronization is enabled

- **“show run | include bgp”**

  Will show you what bestpath knobs you have enabled (bgp deterministic-med, bgp always-compare-med, etc.)

- **“show ip bgp x.x.x.x”**

  Go through the decision algorithm step-by-step

  Understand why the bestpath is the best

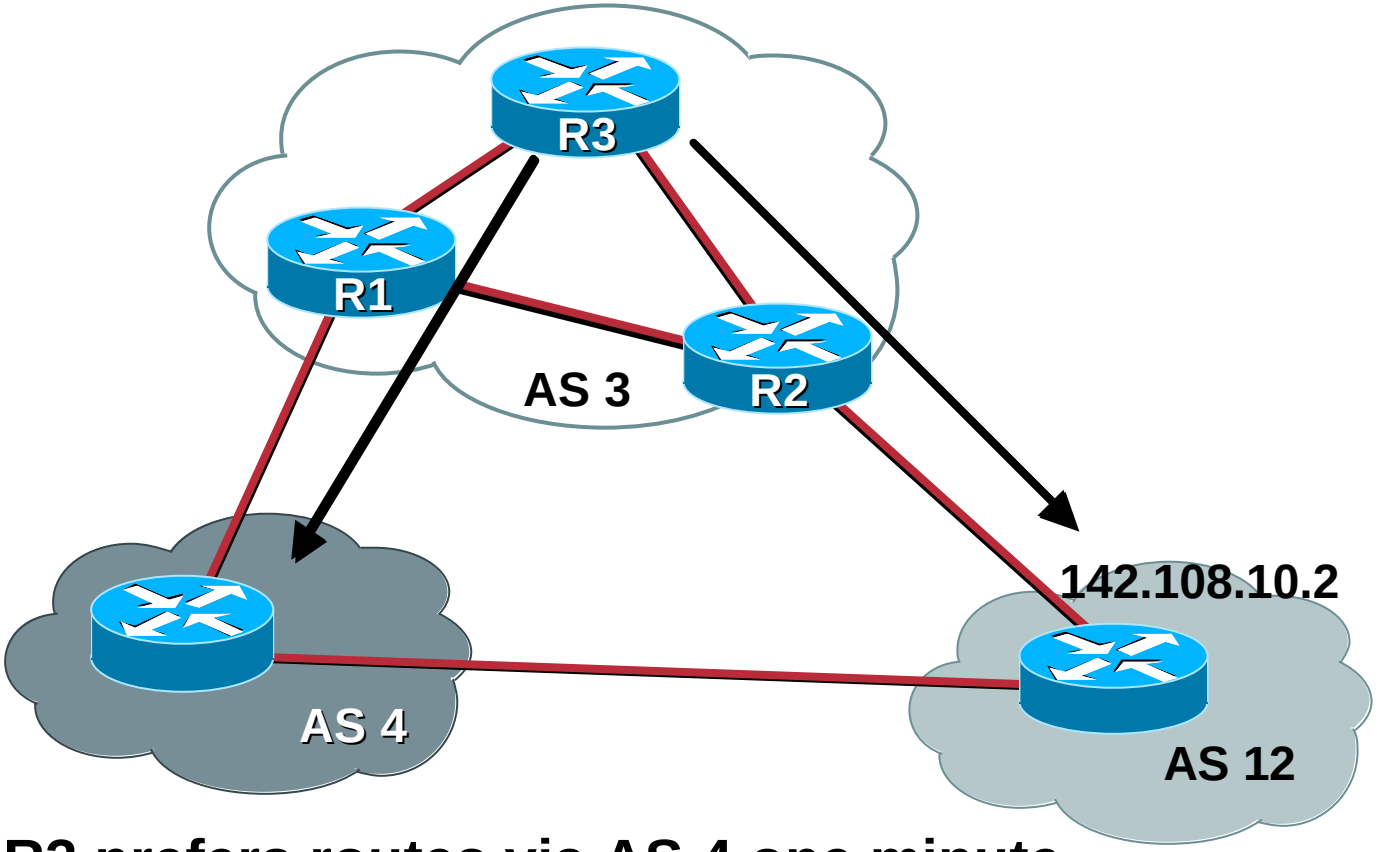
Agenda

- Peer Establishment
- Missing Routes
- Inconsistent Route Selection
- Loops and Convergence Issues

Route Oscillation

- One of the most common problems!
- Every minute routes flap in the routing table from one nexthop to another
- With full routes the most obvious symptom is high CPU in “BGP Router” process

Route Oscillation—Diagram



- R3 prefers routes via AS 4 one minute
- BGP scanner runs then R3 prefers routes via AS 12
- The entire table oscillates every 60 seconds

Route Oscillation—Symptom


```
R3#show ip bgp summary
BGP router identifier 3.3.3.3, local AS number 3
BGP table version is 502, main routing table version 502
267 network entries and 272 paths using 34623 bytes of memory

R3#sh ip route summary | begin bgp
bgp 3          4          6            520        1400
  External: 0 Internal: 10 Local: 0
internal       5                                  5800
Total          10         263        13936       43320
```

- **Watch for:**
  - **Table version number incrementing rapidly**
  - **Number of networks/paths or external/internal routes changing**

Route Oscillation—Troubleshooting

Cisco.com

- Pick a route from the RIB that has changed within the last minute
- Monitor that route to see if it changes every minute

```
R3#show ip route 156.1.0.0
Routing entry for 156.1.0.0/16
  Known via "bgp 3", distance 200, metric 0
 Routing Descriptor Blocks:
  * 1.1.1.1, from 1.1.1.1, 00:00:53 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2, BGP network version 474
```

```
R3#show ip bgp 156.1.0.0
BGP routing table entry for 156.1.0.0/16, version 474
Paths: (2 available, best #1)
  Advertised to non peer-group peers:
    2.2.2.2
  4 12
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, internal, best
  12
    142.108.10.2 (inaccessible) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal
```

Route Oscillation—Troubleshooting

- Check again after bgp_scanner runs
- bgp_scanner runs every 60 seconds and validates reachability to all nexthops

```
R3#sh ip route 156.1.0.0
Routing entry for 156.1.0.0/16
  Known via "bgp 3", distance 200, metric 0
    Routing Descriptor Blocks:
  * 142.108.10.2, from 2.2.2.2, 00:00:27 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 478

R3#sh ip bgp 156.1.0.0
BGP routing table entry for 156.1.0.0/16, version 478
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    1.1.1.1
  4 12
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, internal
  12
    142.108.10.2 from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation—Troubleshooting

- **Lets take a closer look at the nexthop**

```
R3#show ip route 142.108.10.2
Routing entry for 142.108.0.0/16
  Known via "bgp 3", distance 200, metric 0
 Routing Descriptor Blocks:
  * 142.108.10.2, from 2.2.2.2, 00:00:50 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 476

R3#show ip bgp 142.108.10.2
BGP routing table entry for 142.108.0.0/16, version 476
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    1.1.1.1
  4 12
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, internal
  12
    142.108.10.2 from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation—Troubleshooting

- BGP nexthop is known via BGP
- Illegal recursive lookup
- Scanner will notice and install the other path in the RIB

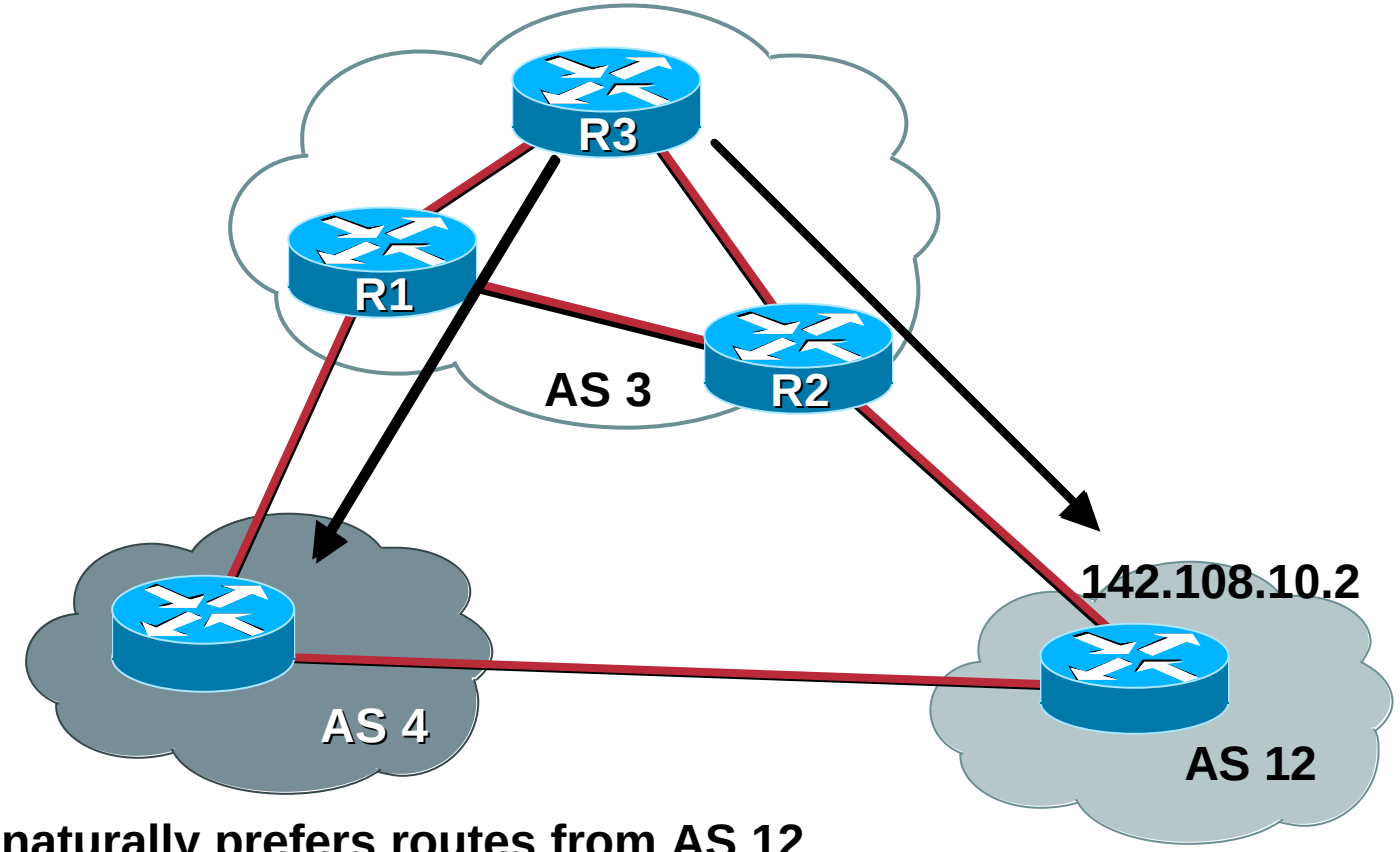
```
R3#sh debug
  BGP events debugging is on
  BGP updates debugging is on
  IP routing debugging is on
R3#
BGP: scanning routing tables
BGP: nettable_walker 142.108.0.0/16 calling revise_route
RT: del 142.108.0.0 via 142.108.10.2, bgp metric [200/0]
BGP: revise route installing 142.108.0.0/16 -> 1.1.1.1
RT: add 142.108.0.0/16 via 1.1.1.1, bgp metric [200/0]
RT: del 156.1.0.0 via 142.108.10.2, bgp metric [200/0]
BGP: revise route installing 156.1.0.0/16 -> 1.1.1.1
RT: add 156.1.0.0/16 via 1.1.1.1, bgp metric [200/0]
```

Route Oscillation—Troubleshooting

- Route to the nexthop is now valid
- Scanner will detect this and re-install the other path
- Routes will oscillate forever

```
R3#
BGP: scanning routing tables
BGP:  ip nettable_walker 142.108.0.0/16 calling revise_route
RT: del 142.108.0.0 via 1.1.1.1, bgp metric [200/0]
BGP: revise route installing 142.108.0.0/16 -> 142.108.10.2
RT: add 142.108.0.0/16 via 142.108.10.2, bgp metric [200/0]
BGP: nettable_walker 156.1.0.0/16 calling revise_route
RT: del 156.1.0.0 via 1.1.1.1, bgp metric [200/0]
BGP: revise route installing 156.1.0.0/16 -> 142.108.10.2
RT: add 156.1.0.0/16 via 142.108.10.2, bgp metric [200/0]
```

Route Oscillation—Step by Step



- R3 naturally prefers routes from AS 12
- R3 does not have an IGP route to 142.108.10.2 which is the next-hop for routes learned via AS 12
- R3 learns 142.108.0.0/16 via AS 4 so 142.108.10.2 becomes reachable

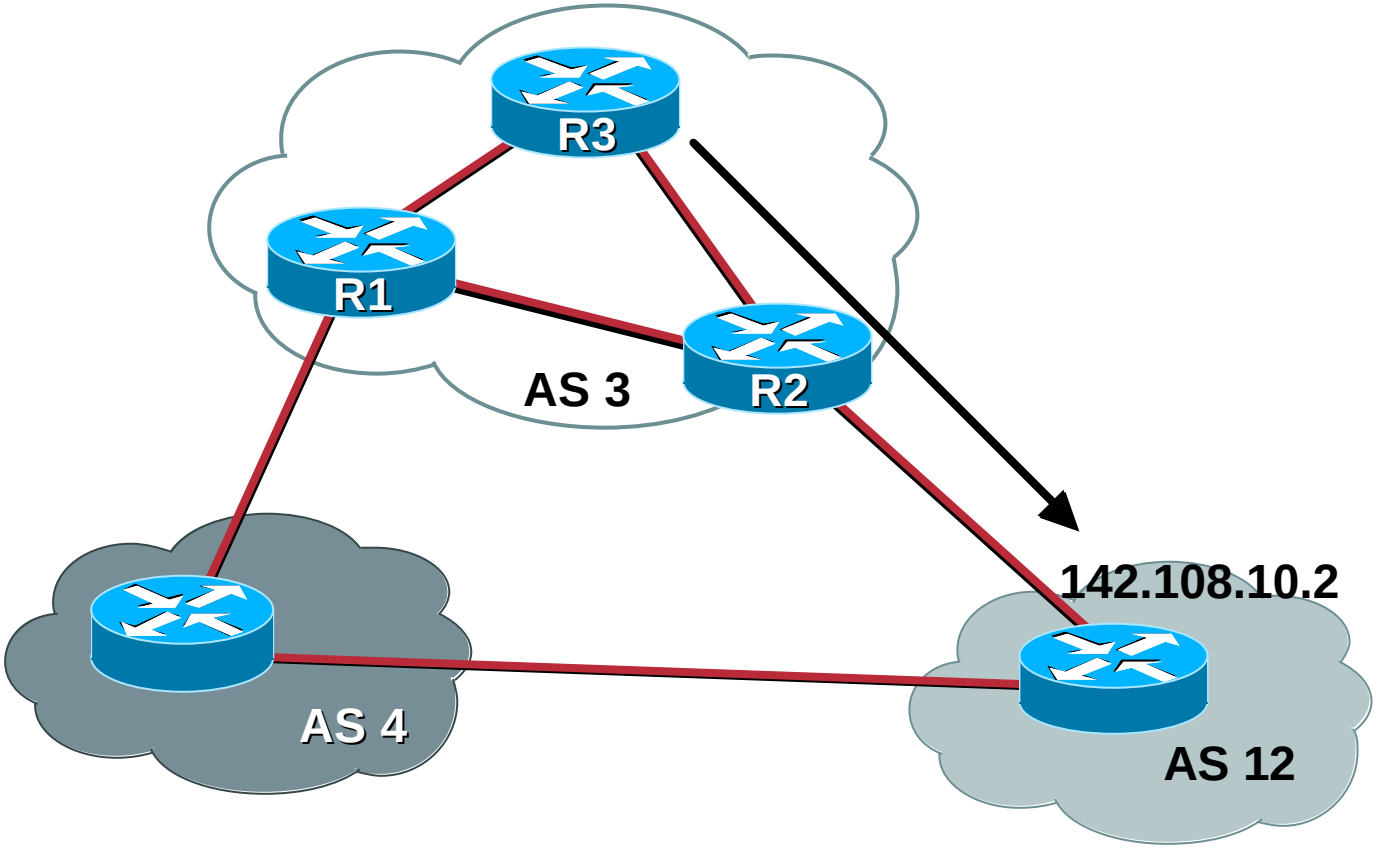
Route Oscillation—Step by Step

- R3 then prefers the AS 12 route for 142.108.0.0/16 whose next-hop is 142.108.10.2
- This is an illegal recursive lookup
- BGP detects the problem when scanner runs and flags 142.108.10.2 as inaccessible
- Routes through AS 4 are now preferred
- The cycle continues forever…

Route Oscillation—Solution

- iBGP preserves the next-hop information from eBGP
- To avoid problems
  - Use “next-hop-self” for iBGP peering
  - Make sure you advertise the next-hop prefix via the IGP

Route Oscillation—Solution

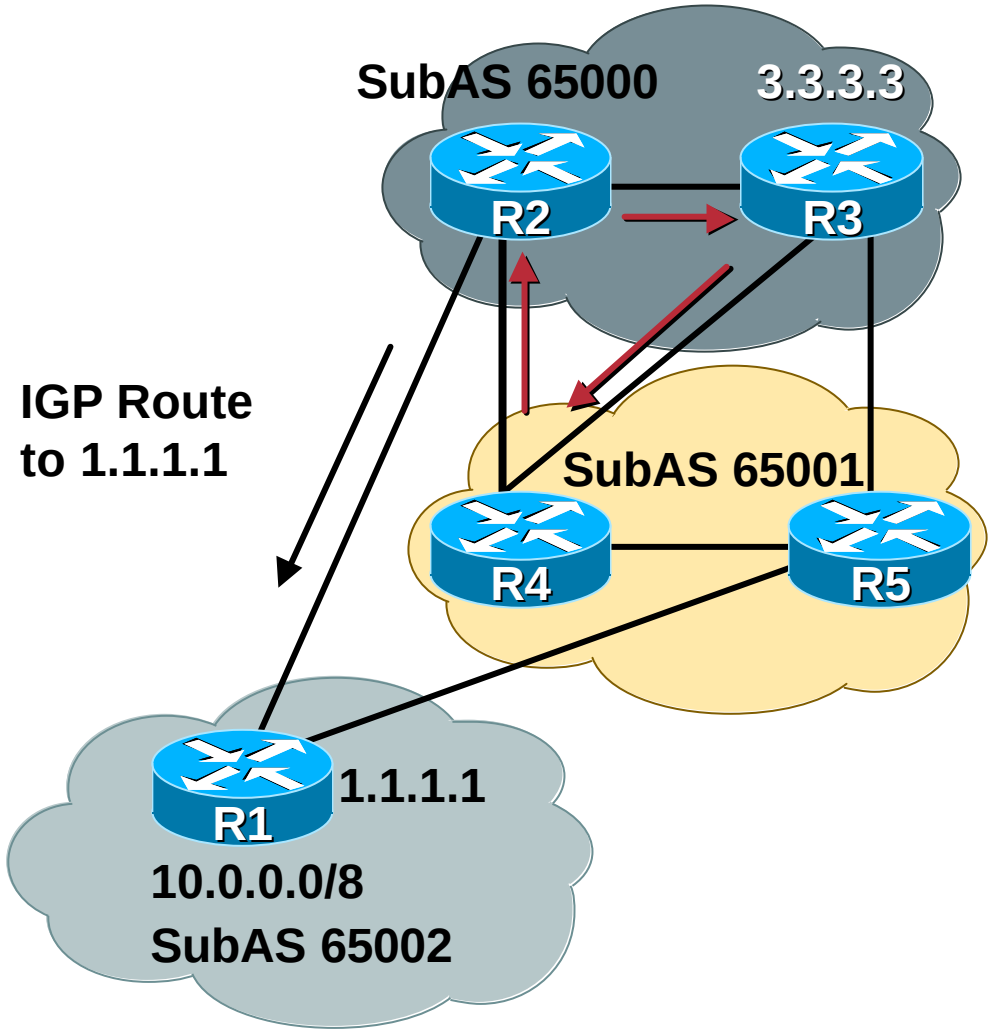


- R3 now has IGP route to AS 12 next-hop or R2 is using next-hop-self
- R3 now prefers routes via AS 12 all the time
- No more oscillation!!

Routing Loop


```
R5# traceroute 10.1.1.1

 1 30.100.1.1
 2 20.20.20.4  - R3
 3 30.1.1.26   - R4
 4 30.1.1.17   - R2
 5 20.20.20.4  - R3
 6 30.1.1.26   - R4
 7 30.1.1.17   - R2
 8 20.20.20.4
 9 30.1.1.26
10 30.1.1.17
```

- Traffic loops between R3, R4, and R2



Routing Loop

- First capture a “show ip route” from the three problem routers
- R3 is forwarding traffic to 1.1.1.1 (R1)

```
R3# show ip route 10.1.1.1
Routing entry for 10.0.0.0/8
  Known via "bgp 65000", distance 200, metric 0
  Routing Descriptor Blocks:
    1.1.1.1, from 5.5.5.5, 01:46:43 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0, BGP network version 0
  * 1.1.1.1, from 4.4.4.4, 01:46:43 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0, BGP network version 0
```

Routing Loop

- **R4 is also forwarding to 1.1.1.1 (R1)**

```
R4# show ip route 10.1.1.1
Routing entry for 10.0.0.0/8
  Known via "bgp 65001", distance 200, metric 0
  Routing Descriptor Blocks:
  * 1.1.1.1, from 5.5.5.5, 01:47:02 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

Routing Loop

- **R2 is forwarding to 3.3.3.3? (R3)**

```
R2# show ip route 10.1.1.1
Routing entry for 10.0.0.0/8
  Known via "bgp 65000", distance 200, metric 0
Routing Descriptor Blocks:
  * 3.3.3.3, from 3.3.3.3, 01:47:00 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0, BGP network version 3
```

- **Very odd that the NEXT_HOP is in the middle of the network**

Routing Loop

- **Verify BGP paths on R2**

```
R2#show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 3
Paths: (4 available, best #1)
  Advertised to non peer-group peers:
    1.1.1.1 5.5.5.5 4.4.4.4
  (65001 65002)
    3.3.3.3 (metric 11) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, confed-internal,
best
 (65002)
    1.1.1.1 (metric 50) from 1.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, confed-external
```

- **R3 path is better than R1 path because of IGP cost to the NEXT_HOP**
- **R3 is advertising the path to us with a NEXT_HOP of 3.3.3.3 ???**

Routing Loop

- **What is R3 advertising?**

```
R3# show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 3
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  5.5.5.5 2.2.2.2
  (65001 65002)
    1.1.1.1 (metric 5031) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, confed-external, best, multipath
  (65001 65002)
    1.1.1.1 (metric 5031) from 5.5.5.5 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, valid, confed-external, multipath
```

- **Hmmm, R3 is using multipath to load-balance**

```
R3#show run | i maximum
 maximum-paths 6
```

Routing Loop

- “maximum-paths” tells the router to reset the NEXT_HOP to himself
  - R3 sets NEXT_HOP to 3.3.3.3
- Forces traffic to come to him so he can load-balance
- Is typically used for multiple eBGP sessions to an AS
  - Be careful when using in Confederations!!
- Need to make R2 prefer the path from R1 to prevent the routing loop
  - Make IGP metric to 1.1.1.1 better than IGP metric to 4.4.4.4

Troubleshooting Tips

- **High CPU in “Router BGP” is normally a sign of a convergence problem**
- **Find a prefix that changes every minute**

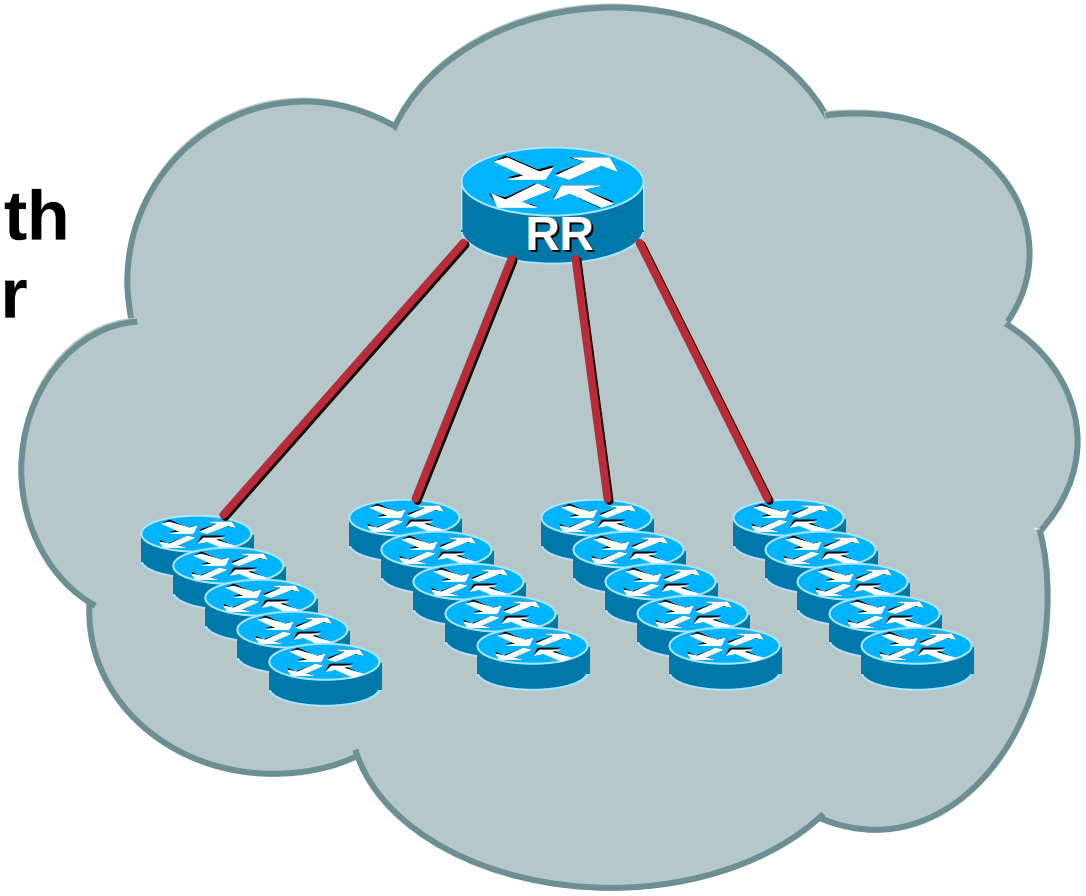
  show ip route | include , 00:00
- **Troubleshoot/debug that one prefix**

Troubleshooting Tips

- **BGP routing loop?**
  - **First, check for IGP routing loops to the BGP NEXT_HOPs**
- **BGP loops are normally caused by**
  - **Not following physical topology in RR environment**
  - **Multipath with confederations**
  - **Lack of a full iBGP mesh**
- **Get the following from each router in the loop path**
  - **show ip route x.x.x.x**
  - **show ip bgp x.x.x.x**
  - **show ip route NEXT_HOP**

Convergence Problems

- Route reflector with 250 route reflector clients
- 100k routes
- BGP will not converge



Convergence Problems

- Have been trying to converge for 10 minutes
- Peers keep dropping so we never converge?

```
RR# show ip bgp summary
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
20.3.1.160      4   100      10    5416     9419    0    0 00:00:12  Closing
20.3.1.161      4   100      11    4418     8055    0  335 00:10:34        0
20.3.1.162      4   100      12    4718     8759    0  128 00:10:34        0
20.3.1.163      4   100       9    3517        0    1    0 00:00:53  Connect
20.3.1.164      4   100      13    4789     8759    0  374 00:10:37        0
20.3.1.165      4   100      13    3126        0    0  161 00:10:37        0
20.3.1.166      4   100       9    5019     9645    0    0 00:00:13  Closing
20.3.1.167      4   100       9    6209     9218    0  350 00:10:38        0
```

- Check the log to find out why

```
RR#show log | i BGP
*May  3 15:27:16: %BGP-5-ADJCHANGE: neighbor 20.3.1.118 Down— BGP Notification sent
*May  3 15:27:16: %BGP-3-NOTIFICATION: sent to neighbor 20.3.1.118 4/0 (hold time expired) 0 bytes
*May  3 15:28:10: %BGP-5-ADJCHANGE: neighbor 20.3.1.52 Down— BGP Notification sent
*May  3 15:28:10: %BGP-3-NOTIFICATION: sent to neighbor 20.3.1.52 4/0 (hold time expired) 0 bytes
```

Convergence Problems

- We are either missing hellos or our peers are not sending them
- Check for interface input drops

```
RR# show interface gig 2/0 | include input drops
Output queue 0/40, 0 drops; input queue 0/75, 72390 drops
RR#
```

- 72k drops will definitely cause a few peers to go down
- We are missing hellos because the interface input queue is very small
- A rush of TCP Acks from 250 peers can fill 75 spots in a hurry
- Increase the size of the queue

```
RR# show run interface gig 2/0
interface GigabitEthernet 2/0
 ip address 7.7.7.156 255.255.255.0
 hold-queue 2000 in
```

Convergence Problems

- **Let's start over and give BGP another chance**

```
RR# clear ip bgp *
RR#
```

- **No more interface input drops**

```
RR# show interface gig 2/0 | include input drops
Output queue 0/40, 0 drops; input queue 0/2000, 0 drops
RR#
```

- **Our peers are stable!!**

```
RR# show log | include BGP
RR#
```

Convergence Problems

- BGP converged in 25 minutes
- Still seems like a long time
- What was TCP doing?

```
RR#show tcp stat | begin Sent:
Sent: 1666865 Total, 0 urgent packets
      763 control packets (including 5 retransmitted)
      1614856 data packets (818818410 bytes)
      39992 data packets (13532829 bytes) retransmitted
      6548 ack only packets (3245 delayed)
      1 window probe packets, 2641 window update packets
```

```
RR#show ip bgp neighbor | include max data segment
Datagrams (max data segment is 536 bytes):
```

Convergence Problems

- 1.6 Million packets is high
- 536 is the default MSS (max segment size) for a TCP connection
- Very small considering the amount of data we need to transfer

```
RR#show ip bgp neighbor | include max data segment
Datagrams (max data segment is 536 bytes):
Datagrams (max data segment is 536 bytes):
```

- Enable path mtu discovery
- Sets MSS to max possible value

```
RR#show run | include tcp
ip tcp path-mtu-discovery
RR#
```

Convergence Problems

- **Restart the test one more time**

```
RR# clear ip bgp *
RR#
```

- **MSS looks a lot better**

```
RR#show ip bgp neighbor | include max data segment
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
```

Convergence Problems

- TCP sent 1 million fewer packets
- Path MTU discovery helps reduce overhead by sending more data per packet

```
RR# show tcp stat | begin Sent:
Sent: 615415 Total, 0 urgent packets
      0 control packets (including 0 retransmitted)
      602587 data packets (818797102 bytes)
      9609 data packets (7053551 bytes) retransmitted
      2603 ack only packets (1757 delayed)
      0 window probe packets, 355 window update packets
```

- BGP converged in 15 minutes!
- A respectable time for 250 peers and 100k routes

Summary/Tips

- Use ACLs when enabling debug commands
- Enable bgp log-neighbor-changes
- Use bgp deterministic-med
- If the entire table is having problem pick one prefix and troubleshoot it

References

- **TAC BGP pages—Very nice**

  http://www.cisco.com/cgi-bin/Support/PSP/psp_view.pl?p=Internetworking:BGP

- **BGP Case Studies**

  http://www.cisco.com/warp/public/459/bgp-toc.html

- **Internet Routing Architectures**

  http://www.ciscopress.com/book.cfm?series=1&book=155

- **Standards**

  RFC 1771, 1997, etc…

  http://www.rfc-editor.org/rfcsearch.html

  http://search.ietf.org/search/brokers/internet-drafts/query.html