Troubleshooting BGP

Philip Smith   <pfs@cisco.com>

NANOG 36

12-15 February 2006

Dallas

Presentation Slides

- ## Available on

  ftp://ftp-eng.cisco.com

  /pfs/seminars/NANOG36-BGP-Troubleshooting.pdf

  And on the NANOG 36 meeting pages at
  http://www.nanog.org/mtg-0602/pdf/smith.pdf

Assumptions

- **Presentation assumes working knowledge of BGP**

  Beginner and Intermediate experience of protocol

- **Knowledge of Cisco CLI**

  Hopefully you can translate concepts into your own router CLI

- **If in any doubt, please ask!**

Agenda

- **Fundamentals of Troubleshooting**

- **Local Configuration Problems**

- **Internet Reachability Problems**

Fundamentals:
Problem Areas

- ## First step is to recognise what usually causes problems

- ## Possible Problem Areas:

  ### Misconfiguration

  Configuration errors caused by bad documentation, misunderstanding of concepts, poor communication between colleagues or departments

  ### Human error

  Typos, using wrong commands, accidents, poorly planned maintenance activities

Fundamentals:
Problem Areas

- ## More Possible Problem Areas:

  ### "feature behaviour"

  #### Or – "it used to do this with Release X.Y(a) but Release X.Y(b) does that"

  ### Interoperability issues

  #### Differences in interpretation of RFC1771 and its developments

  ### Those beyond your control

  #### Upstream ISP or peers make a change which has an unforeseen impact on your network

Fundamentals:
Working on Solutions

- **Next step is to try and fix the problem**

  And this is not about diving into network and trying random commands on random routers, just to "see what difference this makes"

- **The best procedure for "unfamiliar problems" is to**

  Start at one place,

  Deal with one symptom, and learn more about it

Fundamentals:
Working on Solutions

- **Remember! Troubleshooting is about:**

    Not panicking

    Creating a checklist

    Working to that checklist

    Starting at the bottom and working up

Fundamentals: Checklists

- **This presentation will have references in the later stages to checklists**

  They are the best way to work to a solution

  They are what many NOC staff follow when diagnosing and solving network problems

  It may seem daft to start with simple tests when the problem looks complex

  But quite often the apparently complex can be solved quite easily

Fundamentals:
Tools

- **Use system and network logs as an aid**

- **Record keeping:**

  Good and detailed system logs

  Last known good configuration

  History trail of working configurations and all intermediate changes

  Record of commands entered on routers and other network devices

Fundamentals:
Tools

- **Familiarise yourself with the router's tools:**

    Is logging of the BGP process enabled?

    (And is it captured/recorded off the router?)

    Are you familiar with the BGP debug process and commands (if available)

    Check vendor documentation before switching on full BGP debugging – you might get fewer surprises

Fundamentals:
Tools

- **Traffic and traffic flow measurement in the network**

  Unexplained change in traffic levels on an interface, a connection, a peering,…

  Correlation of customer feedback on network or connectivity issues…

Agenda

- **Fundamentals**

- **Local Configuration Problems**

- **Internet Reachability Problems**

Local Configuration Problems

- **Peer Establishment**

- **Missing Routes**

- **Inconsistent Route Selection**

- **Loops and Convergence Issues**

Peer Establishment

- **Routers establish a TCP session**

   Port 179 – Permit in interface filters

   IP connectivity (route from IGP)
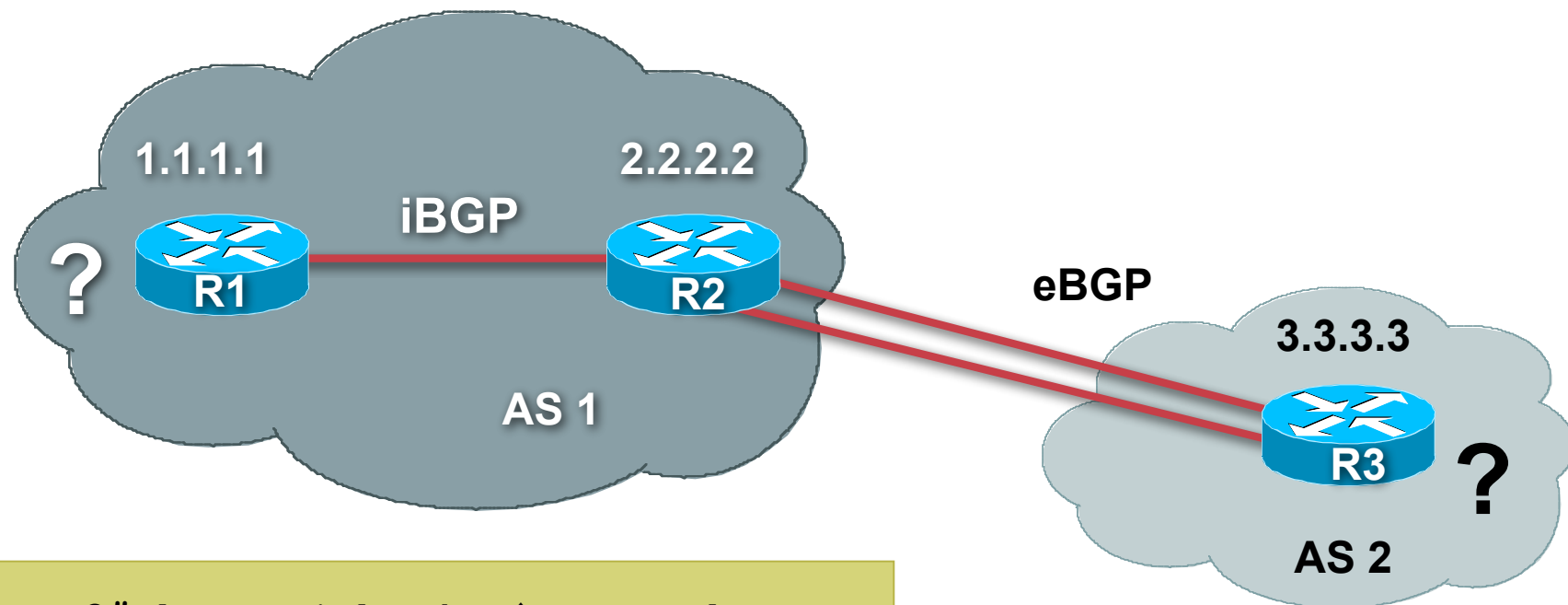
- **OPEN messages are exchanged**

   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

1.1.1.1
2.2.2.2
iBGP
?
R1
R2
eBGP
3.3.3.3
R3
?
AS 1
AS 2

```
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!**

Local AS

iBGP Peer

eBGP Peer

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

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: iBGP – Summary

- **Assume that IP connectivity has been checked**

  Including IGP reachability between peers

- **Check TCP to find out what connections we are accepting**

  Check the ports and source/destination addresses

  Do they match the configuration?

- **Common problem:**

  iBGP is run between loopback interfaces on router (for stability), but the configuration is missing from the router ⇒ iBGP fails to establish

  Remember that source address is the IP address of the outgoing interface unless otherwise specified

Peer Establishment: Diagram
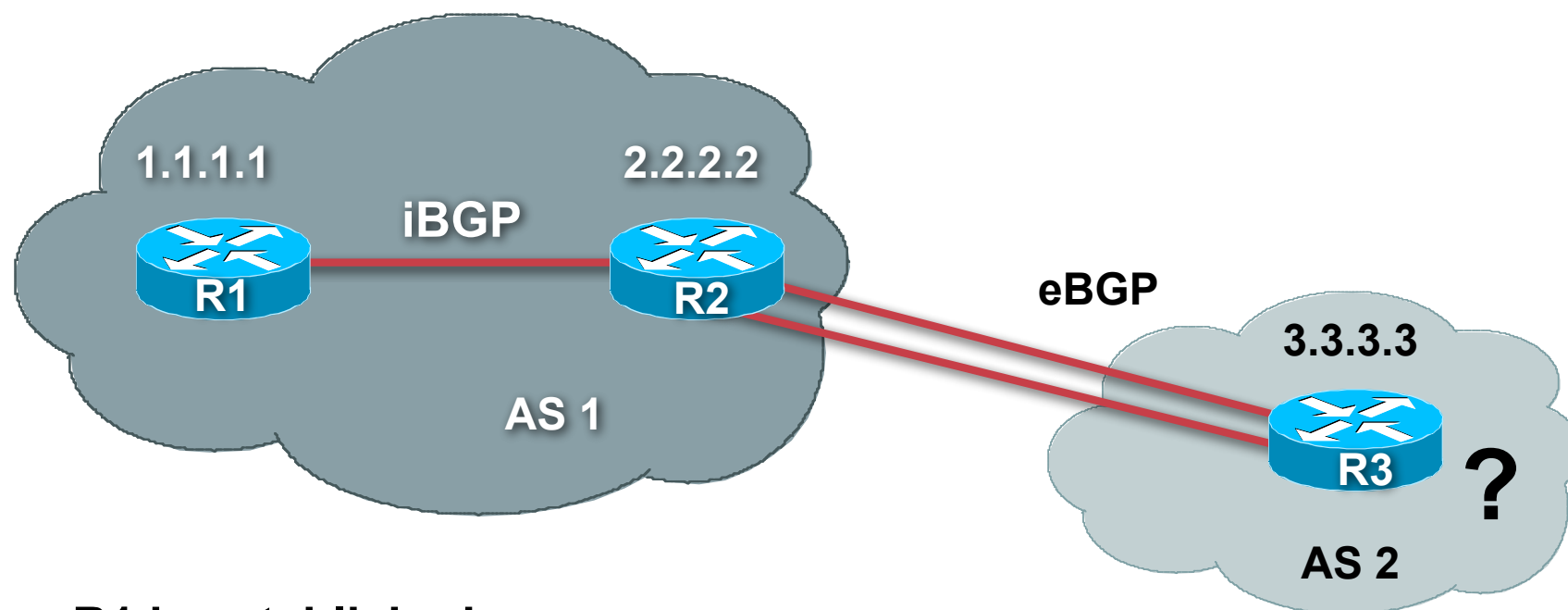
- **R1 is established now**

- **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 is 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 2
 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
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 should come up**

Peer Establishment: eBGP – Summary

- **Remember to allow TCP/179 through edge filters**

```
access-list 100 permit tcp host 3.3.3.3 eq 179 host 2.2.2.2
access-list 100 permit tcp host 3.3.3.3 host 2.2.2.2 eq 179
```

- **Be very careful with multihop eBGP**

    Check IP connectivity (local and remote routing tables)

    Remember to source updates from loopback

    Watch for filters anywhere in the path

    TTL must be at least 2 for ebgp-multihop between directly connected neighbours

    Use TTL value carefully

Peer Establishment: Passwords

- **Using passwords on iBGP and eBGP sessions**

  Link won't come up

  Been through all the previous troubleshooting steps

```
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
```

Peer Establishment: Passwords

```
R2#
router bgp 1
 neighbor 3.3.3.3 remote-as 2
 neighbor 3.3.3.3 ebgp-multihop 2
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 3.3.3.3 password 7 05080F1C221C
```

- **Configuration on R2 looks fine!**

- **Check the log messages – enable "log-neighbor-changes"**

```
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179
to 2.2.2.2:11272
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179
to 2.2.2.2:11272
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179
to 2.2.2.2:11272
```

Peer Establishment: Passwords

```
R3#
router bgp 2
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 ebgp-multihop 2
 neighbor 2.2.2.2 update-source Loopback0
```

- **Check configuration on R3**

  Password is missing from the eBGP configuration

- **Fix the R3 configuration**

  Peering should now come up!

  But it does not

Peer Establishment: Passwords

- **Let's look at the log messages again for clues**

```
R2#

%TCP-6-BADAUTH: Invalid MD5 digest from 3.3.3.3:11024 to 2.2.2.2:179

%TCP-6-BADAUTH: Invalid MD5 digest from 3.3.3.3:11024 to 2.2.2.2:179

%TCP-6-BADAUTH: Invalid MD5 digest from 3.3.3.3:11024 to 2.2.2.2:179
```

- **We are getting invalid MD5 digest messages – password mismatch!**

Peer Establishment: Passwords

- ## We must have typo'ed the password on one of the peering routers

  ### Fix the password – best to re-enter password on both routers

  ### eBGP session now comes up

```
%TCP-6-BADAUTH: Invalid MD5 digest from 3.3.3.3:11027
to 2.2.2.2:179
%BGP-5-ADJCHANGE: neighbor 3.3.3.3 Up
```

Peer Establishment:
Passwords – Summary

- **Common problems:**

    Missing password – needs to be on both ends

    Cut and paste errors – don't!

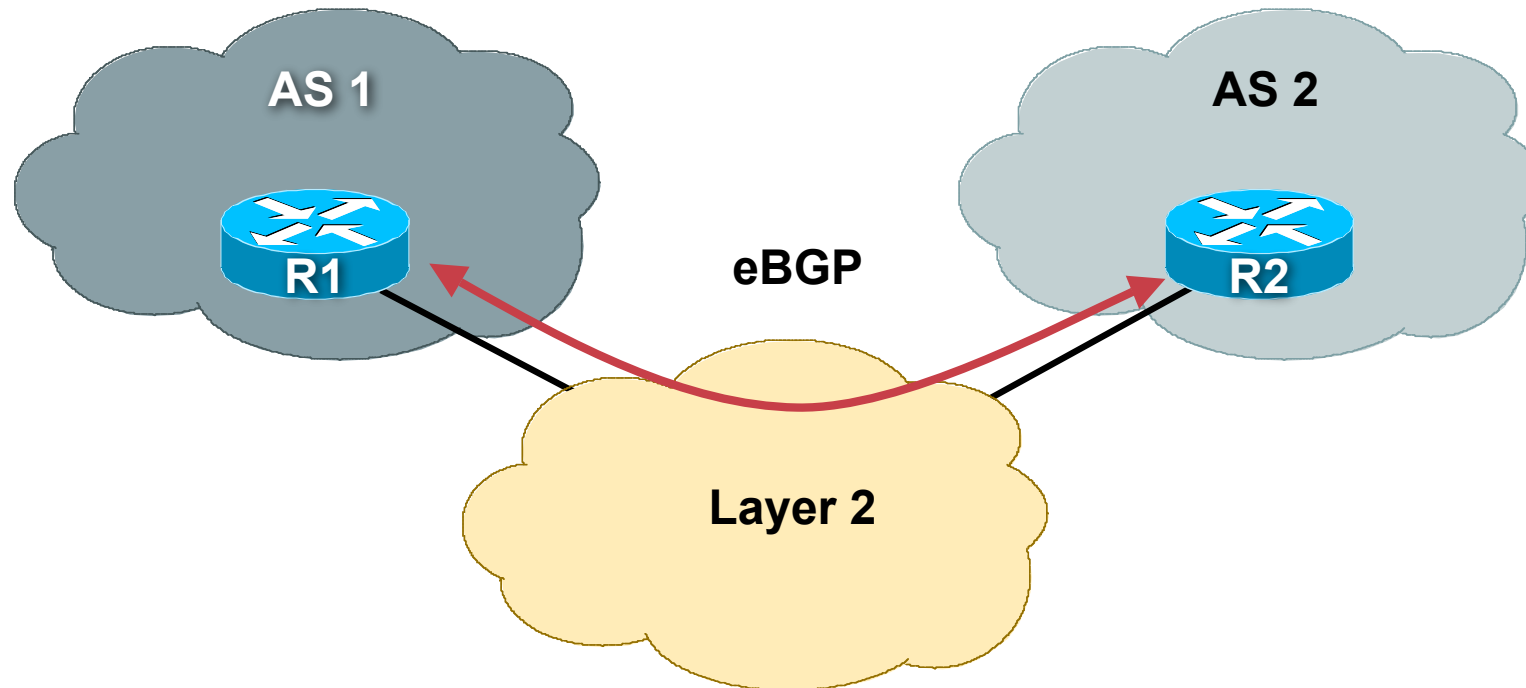   **Typographical & transcription errors**

    Capitalisation, extra characters, white space…

- **Common solutions:**

    Check for symptoms/messages in the logs

    Re-enter passwords using keyboard, from scratch – don't cut&paste

Flapping Peer:
Common Symptoms

AS 1

AS 2

eBGP

R1

R2

Layer 2

- Symptoms – the eBGP session flaps

- eBGP peering establishes, then drops, re-establishes, then drops,…

Flapping Peer

- **Ensure BGP neighbour logging is enabled**

    no logs → no clue what is going on

- **R1 and R2 are peering over some 3rd party L2 network**

```
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 summary
BGP router identifier 172.16.175.53, local AS number 1
BGP table version is 10167, main routing table version 10167

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 ms
```

```
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 1500byte ping fails?**

Flapping Peer:
Diagnosis and Solution

- **Diagnosis**

    Keepalives get lost because they get stuck in the router's queue behind BGP update packets.

    BGP update packets are packed to the size of the MTU – keepalives and BGP OPEN packets are not packed to the size of the MTU ⇒ Path MTU problems

    Use ping with different size packets to confirm the above – 100byte ping succeeds, 1500byte ping fails = MTU problem somewhere

- **Solution**

   **Pass the problem to the L2 folks – but be helpful, try and pinpoint using ping where the problem might be in the network**

Flapping Peer:
Other Common Problems

- **Remote router rebooting continually (typical with a 3-5 minute BGP peering cycle time)**

- **Remote router BGP process unstable, restarting**

- **Traffic Shaping & Rate Limiting parameters**

- **MTU incorrectly set on links, PMTU discovery disabled on router**

- **For non-ATM/FR links, instability in the L2 point-to-point circuits**

  Faulty MUXes, bad connectors, interoperability problems, PPP problems, satellite or radio problems, weather, etc. The list is endless – your L2 folks should know how to solve them

  For you, *ping* is the tool to use

Flapping Peer: Fixed!

AS 1

AS 2

R1

R2

eBGP

Layer 2

- - - → Small Packets
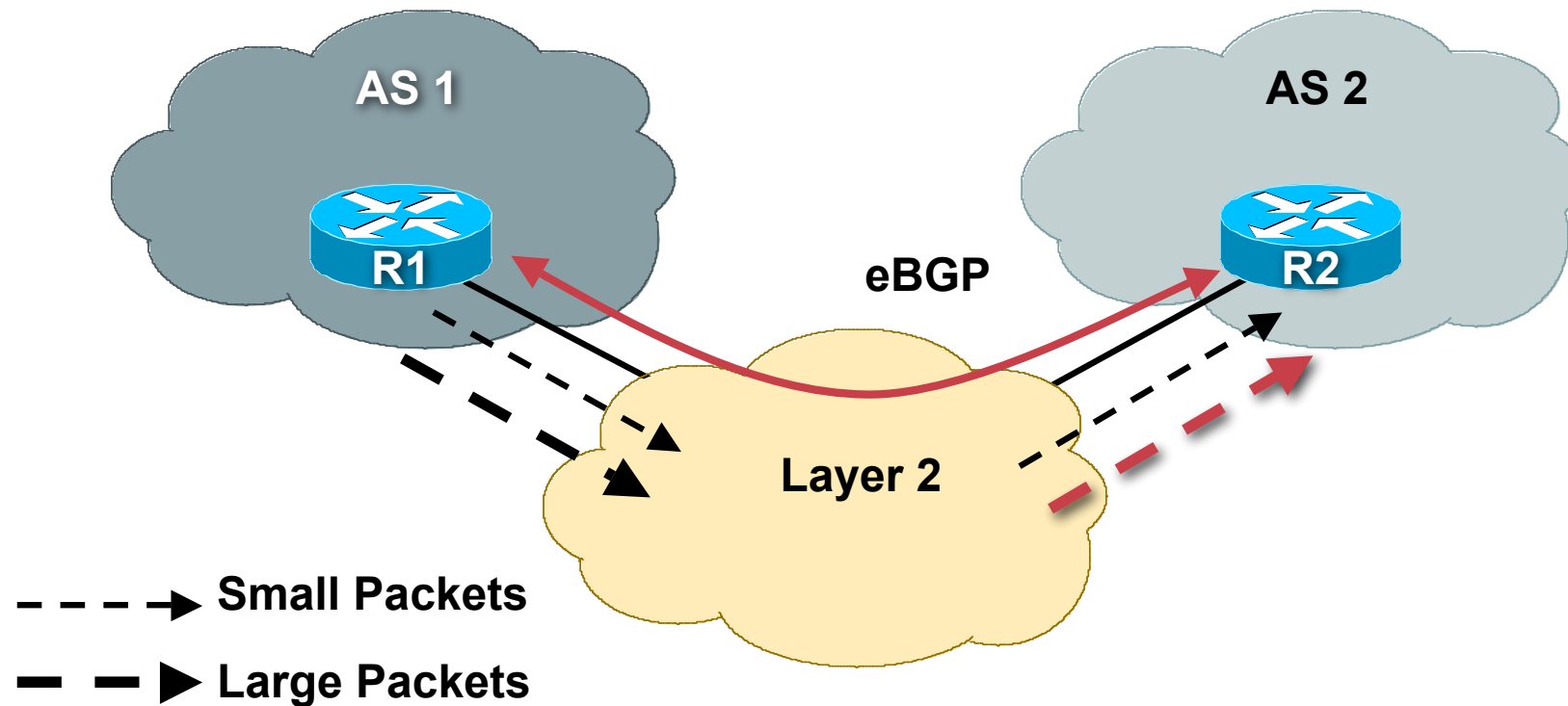
— — ▶ Large Packets

- Large packets are ok now

- BGP session is stable!

Local Configuration Problems

- **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

    (Unless we are using Route Reflectors)

Missing Routes

- **Route Origination**

- **UPDATE Exchange**

- **Filtering**

- **iBGP mesh problems**

Missing Routes:
Route Origination

- **Common problem occurs when putting prefixes into the BGP table**

- **BGP table is NOT the RIB**

  (RIB = Routing Information Base – a.k.a the Routing Table)

  BGP table, as with OSPF table, ISIS table, static routes, etc, is used to feed the RIB, and hence the FIB

  Each routing protocol has a different priority or "distance"

Missing Routes:
Route Origination

- **To get a prefix into BGP, it must exist in another routing process too, typically:**

    Static route pointing to customer (for customer routes into your iBGP)

    Static route pointing to Null (for aggregates you want to put into your eBGP)

Route Origination: Example I

- **Network statement**

  ```
  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 I

- **Nail down routes you want to originate**

  ```
  ip route 200.200.0.0 255.255.252.0 Null0 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

  *> 200.200.0.0/22    0.0.0.0                0     32768
  ```

Route Origination: Example II

- **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 II

- **Remember, to have a BGP aggregate you need a BGP component, not a RIB 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     0.0.0.00        32768 i
```

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

Troubleshooting Tips

- **BGP Network statement 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**

- **iBGP mesh problems**

Missing Routes:
Update Exchange

- ## Ah, Route Reflectors…

  Such a nice solution to help scale iBGP

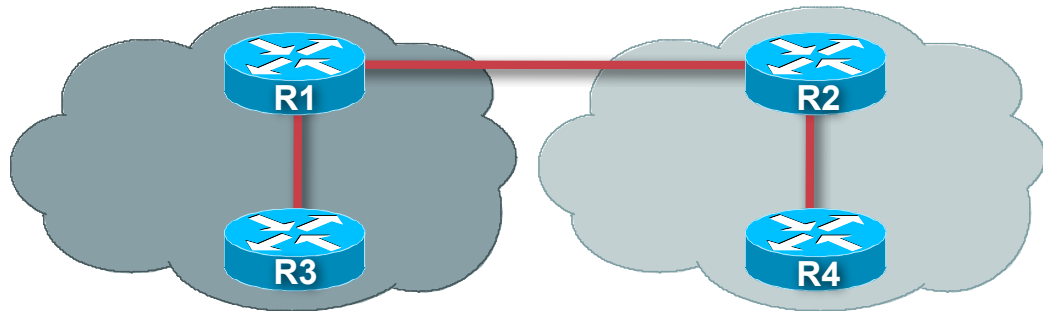 **But why do people insist in breaking the rules all the time?!**

- ## Common issues

  Clashing router IDs

  Clashing cluster IDs

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 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 – Summary

- ## R1 is not accepting the route when R2 sends it on from its client, R4

  R1 and R4 have the same router ID!

  If R1 sees its own router ID in the originator attribute in any received prefix, it will reject that prefix

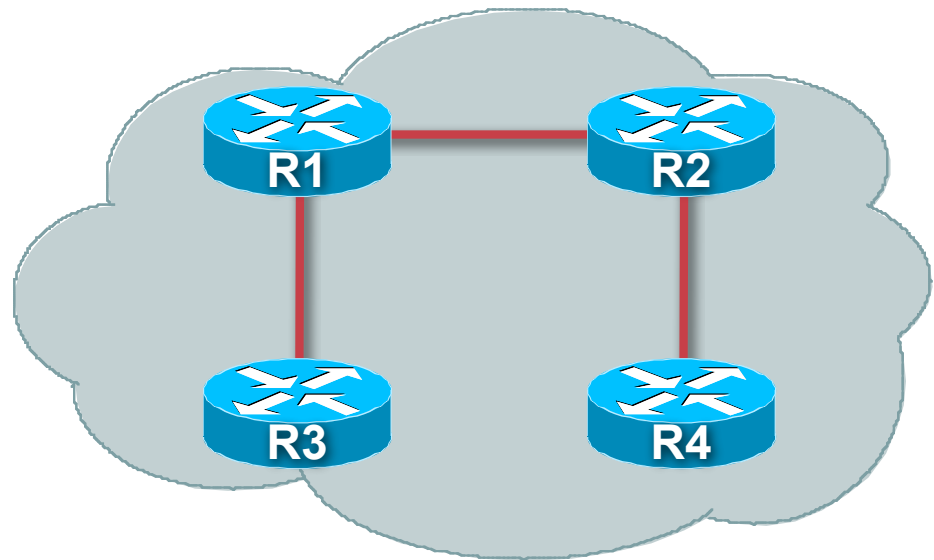 **This is how a route reflector attempts to avoid routing loops**

- ## Solution

  Do NOT set the router ID by hand unless you have a very good reason to do so and have a very good plan for deployment

  Router-ID is usually calculated automatically by router

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 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**

Missing Routes:
Example II – Summary

- ## R1 is not accepting the route when R2 sends it on

  If R1 sees its own router ID in the cluster-ID attribute in any received prefix, it will reject that prefix

  How a route reflector avoids redundant information

- ## Reason

  Early documentation claimed that RRC redundancy should be achieved by dual route reflectors in the same cluster

  This is fine and good, but then ALL clients must peer with both RRs, otherwise examples like this will occur

- ## Solution

  Use overlapping Route Reflector Clusters for redundancy, stay with defaults

Troubleshooting Tips

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

  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`"**

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

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

Troubleshooting Tips
"soft-reconfiguration"

- **Ideal for troubleshooting problems with inbound filters and attributes**

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

```
alpha#sh ip bgp neigh 192.168.12.1 routes

   Network          Next Hop      Metric LocPrf Weight Path

*>i1.0.0.0          192.168.12.1      0     50      0 i

*>i222.222.0.0/19  192.168.5.1            200      0 3 4 i
```

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

```
alpha#sh ip bgp neigh 192.168.12.1 received-routes

   Network          Next Hop      Metric LocPrf Weight Path

* i1.0.0.0          192.168.12.1      0     100     0 i

* i169.254.0.0      192.168.5.1       0     100     0 3 i

* i222.222.0.0/19  192.168.5.1            100     0 3 4 i
```

Missing Routes

- **Route Origination**

- **UPDATE Exchange**

- **Filtering**

- **iBGP mesh problems**

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**

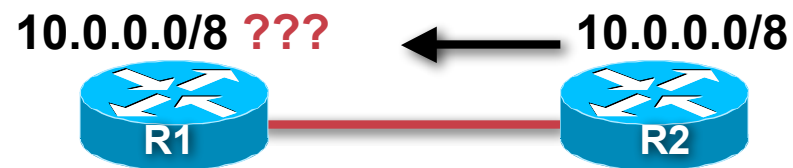   **show ip bgp neighbors x.x.x.x**

    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

10.0.0.0/8 ??? ← **10.0.0.0/8**

R1 ——— R2

- **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

```
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 it has the route**

```
R2#clear ip bgp 1.1.1.1 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

10.0.0.0/8 ??? ← **10.0.0.0/8**

R1 ———— R2

- ## 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

- **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**

Update Filtering:
Summary

- **If you suspect a filtering problem, become familiar with the router tools to find out what BGP filters are applied**

- **Tip: don't cut and paste!**

  Many filtering errors and diagnosis problems result from cut and paste buffer problems on the client, the connection, and even the router

Update Filtering:
Common Problems

- ## Typos in regular expressions

  Extra characters, missing characters, white space, etc
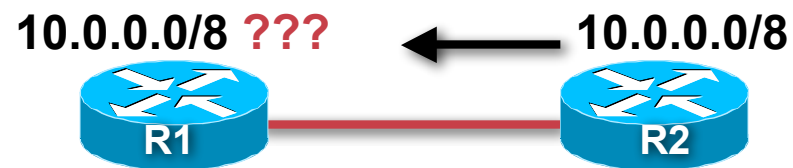
  In regular expressions every character matters, so accuracy is highly important

- ## Typos in prefix filters

  Watch the router CLI, and the filter logic – it may not be as obvious as you think, or as simple as the manual makes out

  Watch netmask confusion, and 255 profusion – easy to muddle 255 with 0 and 225!

Missing Routes
Community Problems

10.0.0.0/8 **???** ⬅ 10.0.0.0/8

R1 — R2

- **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
Community Problems

- ## R2 originates the route

```
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
```

- ## But the community is not set

   ### Would be displayed in the "show ip bgp" output

Missing Routes
Community Problems

- **Fix the configuration so community is set**

```
R2#show run | begin bgp
router bgp 2
 network 10.0.0.0 route-map set-community
...
route-map set-community permit 10
 set community 2:2 1:50
```

```
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
  Community 2:2 1:50
```

Missing Routes
Community Problems

- **R2 now advertises prefix with community to R1**

- **But R1 still doesn't see the prefix**

  R1 insists there is nothing wrong with their configuration

```
R1#show ip bgp neigh 2.2.2.2 routes


Total number of prefixes 0
```

- **Configuration verified on R2**

- **No filters blocking announcement on R2**

- **So what's wrong?**

Missing Routes
Community Problems

- **Check R2 configuration again!**

```
R2#show run | begin bgp
router bgp 2
 network 10.0.0.0 route-map set-community
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 prefix-list my-agg out
 neighbor 1.1.1.1 prefix-list their-agg in
!
ip prefix-list my-agg permit 10.0.0.0/8
ip prefix-list their-agg permit 20.0.0.0/8
!
route-map set-community permit 10
 set community 2:2 1:50
```

- **Looks okay — filters okay, route-map okay**

- **But forgotten "neighbor 1.1.1.1 send-community"**

  Cisco IOS does NOT send communities by default

Missing Routes
Community Problems

- **R2 now advertises prefix with community to R1**

- **But R1 still doesn't see the prefix**

  Nothing wrong on R2 now, so turn attention to R1

```
R1#show run | begin bgp
router bgp 1
 neighbor 2.2.2.2 remote-as 2
 neighbor 2.2.2.2 route-map R2-in in
 neighbor 2.2.2.2 route-map R1-out out
!
ip community-list 1 permit 1:150
!
route-map R2-in permit 10
 match community 1
 set local-preference 150
```

- **Community match on R1 expects 1:150 to be set on prefix**

- **But R2 is sending 1:50**

  Typo or miscommunication between operations?

- **R2 is also using the route-map to filter**

  If the prefix does not have community 1:150 set, it is dropped – there is no next step in the route-map

  Watch the route-map rules in Cisco IOS – they are basically:

  > **if <match> then <set> and exit route-map**

  > **else if <match> then <set> and exit route-map**

  > **else if <match> then <set> etc…**

  Blank route-map line means match everything, set nothing

Missing Routes
Community Problems

- **Fix configuration on R2 to set community 1:150 on announcements to R1**

- **Fix configuration on R1 to also permit prefixes not matching the route-map — troubleshooting is easier with prefix-filters doing the filtering**

```
R1#show run | begin ^route-map
route-map R2-in permit 10
 match community 1
 set local-preference 150
route-map R2-in permit 20
```

```
R1#show ip bgp neigh 2.2.2.2 routes

   Network      Next Hop      Metric LocPrf Weight Path
*  10.0.0.0     2.2.2.2                0              0 2 i


Total number of prefixes 1
```

Missing Routes
Community Problems

- ## Watch route-maps

  ### Route-map rules often catch out operators when they are used for filtering

  ### Absence of an appropriate match means the prefix will be discarded

- ## Remember to configure all routers to send BGP communities

  ### Include it in your default template for iBGP

  #### It should be iBGP default in a Service Provider Network

  ### Remember that it is required to send communities for eBGP too

Missing Routes:
Common Community Problems

- **Each router implementation has different defaults for when communities are sent**

  - Some don't send communities

  - Others do for iBGP and not for eBGP

  - Others do for both iBGP and eBGP peers

- **Watch how your implementation handles communities**

  - There may be implicit filtering rules

- **Each ISP has different community policies**

  - Never assume that because communities exist that people will use them, or pay attention to the ones you send

Missing Routes:
General Problems

- **Make and then Stick to simple policy rules:**

  Most router implementations have particular rules for filtering of prefixes, AS-paths, and for manipulating BGP attributes

  Try not to mix these rules

- **Rules for manipulating attributes can also be used for filtering prefixes and ASNs**

  These can be very powerful, but can also become very confusing

Missing Routes

- **Route Origination**

- **UPDATE Exchange**

- **Filtering**

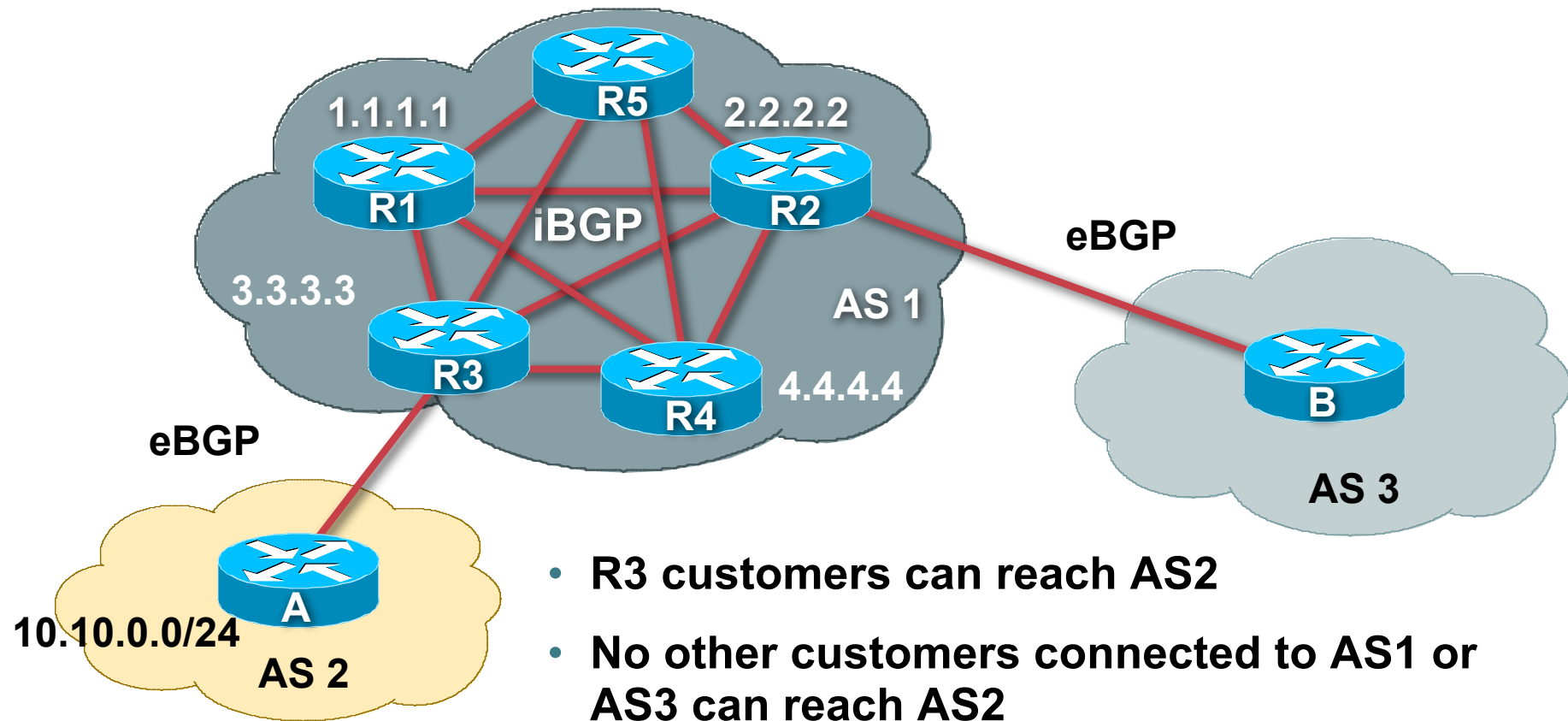- **iBGP mesh problems**

Missing Routes
iBGP Example I

- **Symptom: prefixes seen across network, but no connectivity**

    Prefixes learned from eBGP peer are passed across iBGP mesh

    But no connectivity to those prefixes

Missing Routes
iBGP Example I

- **R3 customers can reach AS2**

- **No other customers connected to AS1 or AS3 can reach AS2**

Missing Routes
iBGP Example I

- ## Looking at R3

```
R3#show ip bgp
Status codes: * valid, > best, i - internal,
   Network          Next Hop        Metric LocPrf Weight Path
*> 3.0.0.0          10.10.10.10                        0 2 5 i
*> 4.0.0.0          10.10.10.10                        0 2 5 i
*> 10.10.0.0/24     10.10.10.10                        0 2 i
*> 10.20.0.0/16     10.10.10.10                        0 2 i
```

- ## Looking at R4

```
R4#show ip bgp
   Network          Next Hop        Metric LocPrf Weight Path
*  i3.0.0.0         10.10.10.10            100        0 2 5 i
*  i4.0.0.0         10.10.10.10            100        0 2 5 i
*  i10.10.0.0/24    10.10.10.10            100        0 2 i
*  i10.20.0.0/16    10.10.10.10            100        0 2 i
```
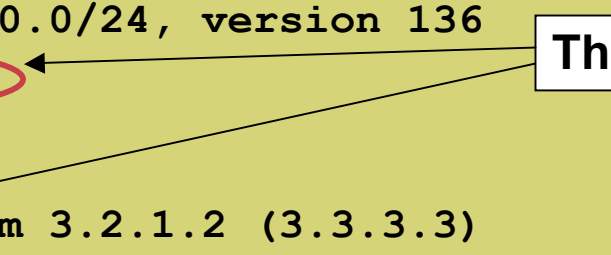
Missing Routes:
iBGP Example I

- **Notice that R3 reports the prefixes learned from AS2**

    Paths are valid (*) and best (>)

- **Notice that R4 reports the prefixes learned from R3**

    Paths are valid (*) and internal (i)

    But no best path

    This is the clue…

Missing Routes:
iBGP Example I

- ## Look at the BGP table entry:

```
R4#sh ip bgp 10.10.0.0/24
BGP routing table entry for 10.10.0.0/24, version 136
Paths: (1 available, no best path)
  Not advertised to any peer
  2, (received & used)
    10.10.10.10 (inaccessible) from 3.2.1.2 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal
```

The clues

- ## Look at the Routing Table entry

```
R4#sh ip route 10.10.0.0 255.255.255.0
% Network not in table
```

- ## The next hop?

```
R4#sh ip route 10.10.10.10
% Network not in table
```

Missing Routes:
iBGP Example I – Diagnosis

- **R4 does not use the 10.10.0.0/24 destination because there is no valid next-hop**

- **Configuration on R3 has:**

  Either no routing information on how to reach the 10.10.10.10/30 point to point link

  By forgetting to put the link into the IGP

  Or not excluded external next-hops from the internal network

  By forgetting to set itself as the next-hop for all externally learned prefixes on the iBGP session with R4

- **Make sure that all the BGP NEXT_HOPs are known by the IGP**

  (whether OSPF/ISIS, static or connected routes)

  If NEXT_HOP is also in iBGP, ensure the iBGP distance is longer than the IGP distance

  —or—

- **Don't carry external NEXT_HOPs in your network**

  Replace eBGP next_hop with local router address on all the edge BGP routers

  (Cisco IOS "next-hop-self")

Missing Routes
iBGP Example I – Solution

- **R3 now includes the missing "next-hop-self" configuration**

- **Looking at R4 now:**

```
R4#show ip bgp
   Network          Next Hop         Metric LocPrf Weight Path
*>i3.0.0.0          3.3.3.3                  100        0 2 5 i
*>i4.0.0.0          3.3.3.3                  100        0 2 5 i
*>i10.10.0.0/24     3.3.3.3                  100        0 2 i
*>i10.20.0.0/16     3.3.3.3                  100        0 2 i
```
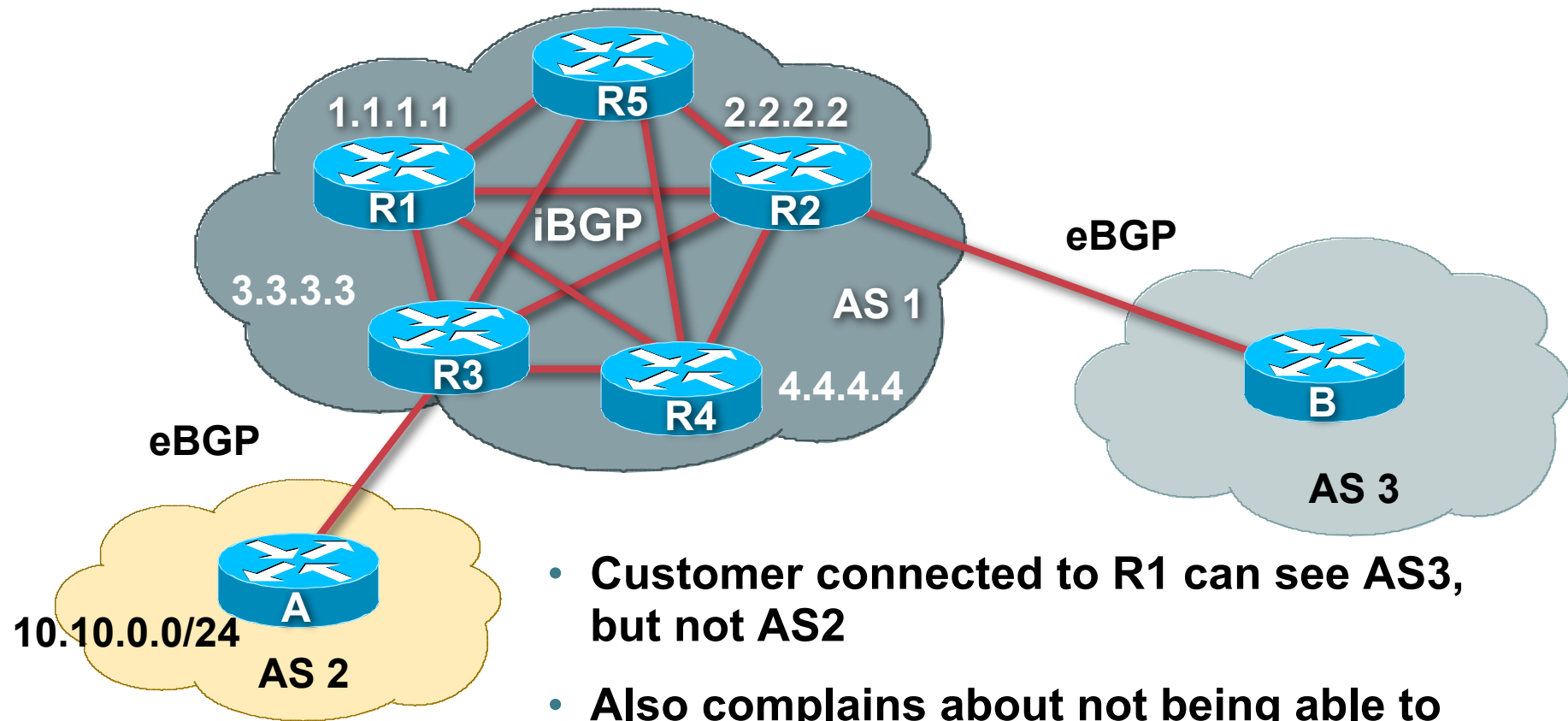
Missing Routes
iBGP Example II

- **Symptom: customer complains about patchy Internet access**

  Can access some, but not all, sites connected to backbone

  Can access some, but not all, of the Internet

Missing Routes
iBGP Example II

1.1.1.1   **R5**   2.2.2.2

R1   **iBGP**   **R2**

eBGP

3.3.3.3   **AS 1**

R3   **R4**   4.4.4.4

B

eBGP   **AS 3**

A

10.10.0.0/24

AS 2

- **Customer connected to R1 can see AS3, but not AS2**

- **Also complains about not being able to see sites connected to R5**

- **No complaints from other customers**

- **Diagnosis: This is the classic iBGP mesh problem**

    The full mesh isn't complete — how do we know this?

- **Customer is connected to R1**

    Can't see AS2 ⇒ R3 is somehow not passing routing information about AS2 to R1

    Can't see R5 ⇒ R5 is somehow not passing routing information about sites connected to R5

    But can see rest of the Internet ⇒ his prefix is being announced to some places, so not an iBGP origination problem

Missing Routes
iBGP Example II

```
R3#sh ip bgp sum | begin ^Neigh
Neighbor      V   AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
1.1.1.1       4    1     200      20     32   0    0 3d10h    Active
2.2.2.2       4    1     210      25     32   0    0 3d16h        15
4.4.4.4       4    1     213      22     32   0    0 3d16h        12
5.5.5.5       4    1     215      19     32   0    0 3d16h         0
10.10.10.10   4    2    2501    2503     32   0    0 3d16h       100
R3#
```

- **BGP summary shows that the peering with router R1 is down**

    Up/Down is 3 days 10 hours, yet active

    Which means it was last up 3 days and 10 hours ago

    So something has broken between R1 and R3

Missing Routes
iBGP Example II

- **Now check configuration on R1**

```
R1#sh conf | b bgp
router bgp 1
 neighbor iBGP-ipv4-peers peer-group
 neighbor iBGP-ipv4-peers remote-as 1
 neighbor iBGP-ipv4-peers update-source Loopback0
 neighbor iBGP-ipv4-peers send-community
 neighbor iBGP-ipv4-peers prefix-list ibgp-prefixes out
 neighbor 2.2.2.2 peer-group iBGP-ipv4-peers
 neighbor 4.4.4.4 peer-group iBGP-ipv4-peers
 neighbor 5.5.5.5 peer-group iBGP-ipv4-peers
```

- **Where is the peering with R3?**

- **Restore the missing line, and the iBGP with R3 comes back up**

Missing Routes
iBGP Example II

```
R3#sh ip bgp sum | begin ^Neigh
Neighbor      V  AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down   State/PfxRcd
1.1.1.1       4   1     200      20       32    0    0 00:00:50             8
2.2.2.2       4   1     210      25       32    0    0 3d16h               15
4.4.4.4       4   1     213      22       32    0    0 3d16h               12
5.5.5.5       4   1     215      19       32    0    0 3d16h                0
10.10.10.10   4   2    2501    2503       32    0    0 3d16h              100
R3#
```

• **BGP summary shows that no prefixes are being heard from R5**

   This could be due to inbound filters on R3 on the iBGP with R5

   But there were no filters in the configuration on R3

   This must be due to outbound filters on R5 on the iBGP with R3

Missing Routes
iBGP Example II

- ## Now check configuration on R5

```
R5#sh conf | b neighbor 3.3.3.3
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source loopback0
 neighbor 3.3.3.3 prefix-list ebgp-filters out
 neighbor 4.4.4.4 remote-as 1
 neighbor 4.4.4.4 update-source loopback0
 neighbor 4.4.4.4 prefix-list ibgp-filters out
!
ip prefix-list ebgp-filters permit 20.0.0.0/8
ip prefix-list ibgp-filters permit 10.0.0.0/8
```

- ## Error in prefix-list in R3 iBGP peering

  ### Ebgp-filters has been used instead of ibgp-filters

  ### Typo — another advantage of using peer-groups!

Missing Routes
iBGP Example II

- **Fix the prefix-list on R5**

- **Check the iBGP again on R3**

  Peering with R1 is up

  Peering with R5 has prefixes

- **Confirm that all is okay with customer**

```
R3#sh ip bgp sum | begin ^Neigh
Neighbor       V   AS MsgRcvd MsgSent     TblVer   InQ OutQ Up/Down    State/PfxRcd
1.1.1.1        4   1     200      20        32      0    0 00:01:53            8
2.2.2.2        4   1     210      25        32      0    0 3d16h              15
4.4.4.4        4   1     213      22        32      0    0 3d16h              12
5.5.5.5        4   1     215      19        32      0    0 3d16h               6
10.10.10.10    4   2    2501    2503        32      0    0 3d16h             100
R3#
```

Troubleshooting Tips

- **Watch the iBGP full mesh**

    Use peer-groups both for efficiency and to avoid making policy errors within the iBGP mesh

    Use route reflectors to avoid accidentally missing iBGP peers, especially as the mesh grows in size

- **Watch the next-hop for external paths**

Local Configuration Problems

- **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 defined 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**

- **RFC1771 is now obsoleted by RFC4271**

  Hopefully compliance with RFC4271 will help avoid future issues

Inconsistent Route Selection: Example I

- **RFC1771 said that MED is not always compared**

- **As a result, the ordering of the paths can effect the decision process**

- **For example, the default in Cisco IOS is to compare the prefixes in order of arrival (most recent to oldest)**

  This can result in inconsistent route selection

  Symptom is that the best path chosen after each BGP reset is different

Inconsistent Route Selection: Example I

- **Inconsistent route selection may cause problems**

  Routing loops

  Convergence loops — i.e. the protocol continuously sends updates in an attempt to converge
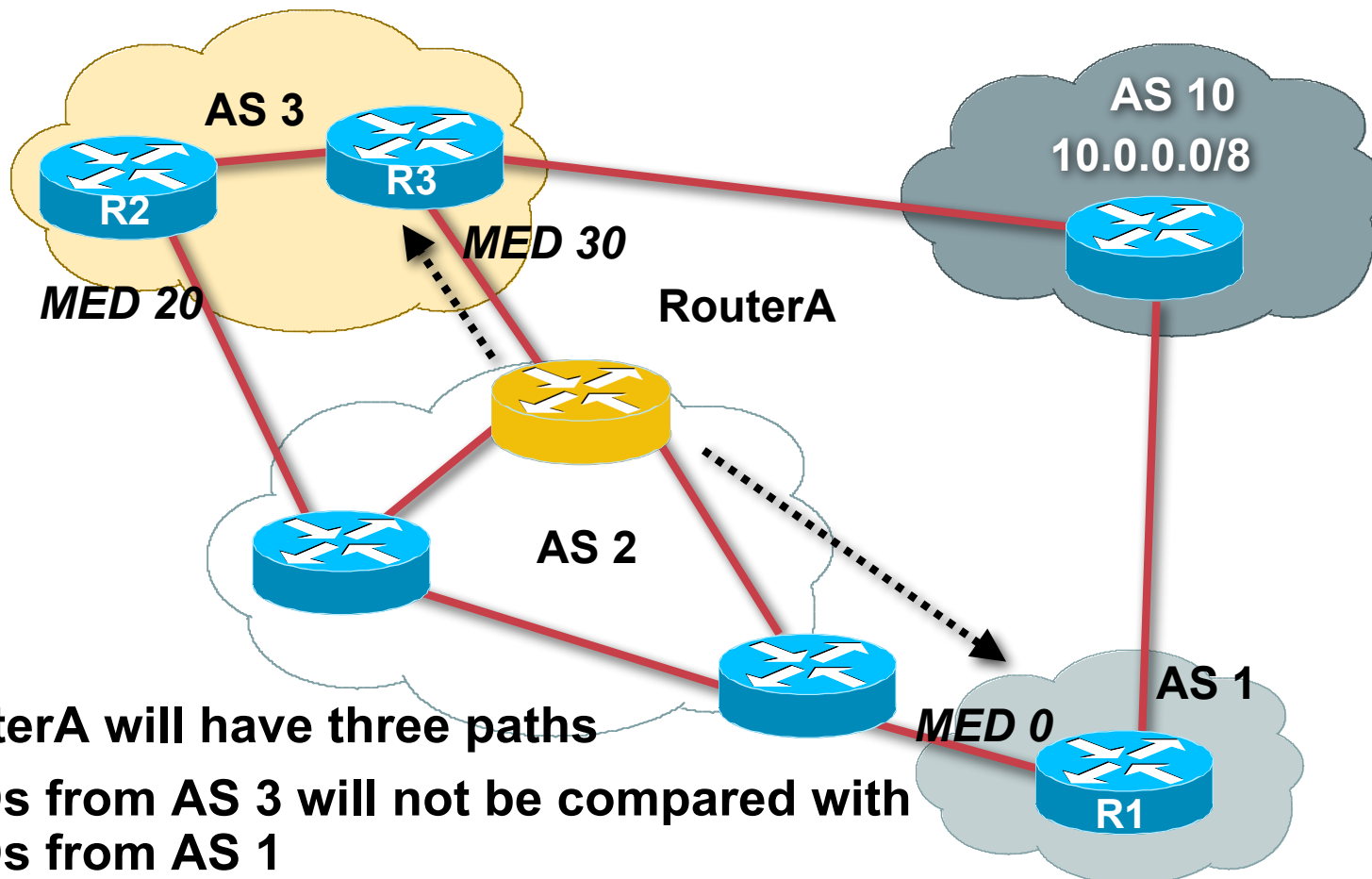
  Changes in traffic patterns

- **Difficult to catch and troubleshoot**

  In Cisco IOS, the deterministic-med configuration command is used to order paths consistently

  Enable in all the routers in the AS

  The bestpath is recalculated as soon as the command is entered

Symptom I: Diagram

AS 3

R2

R3

MED 30

MED 20

RouterA

AS 10
10.0.0.0/8

AS 2

AS 1

MED 0

R1

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

Inconsistent Route Selection: 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
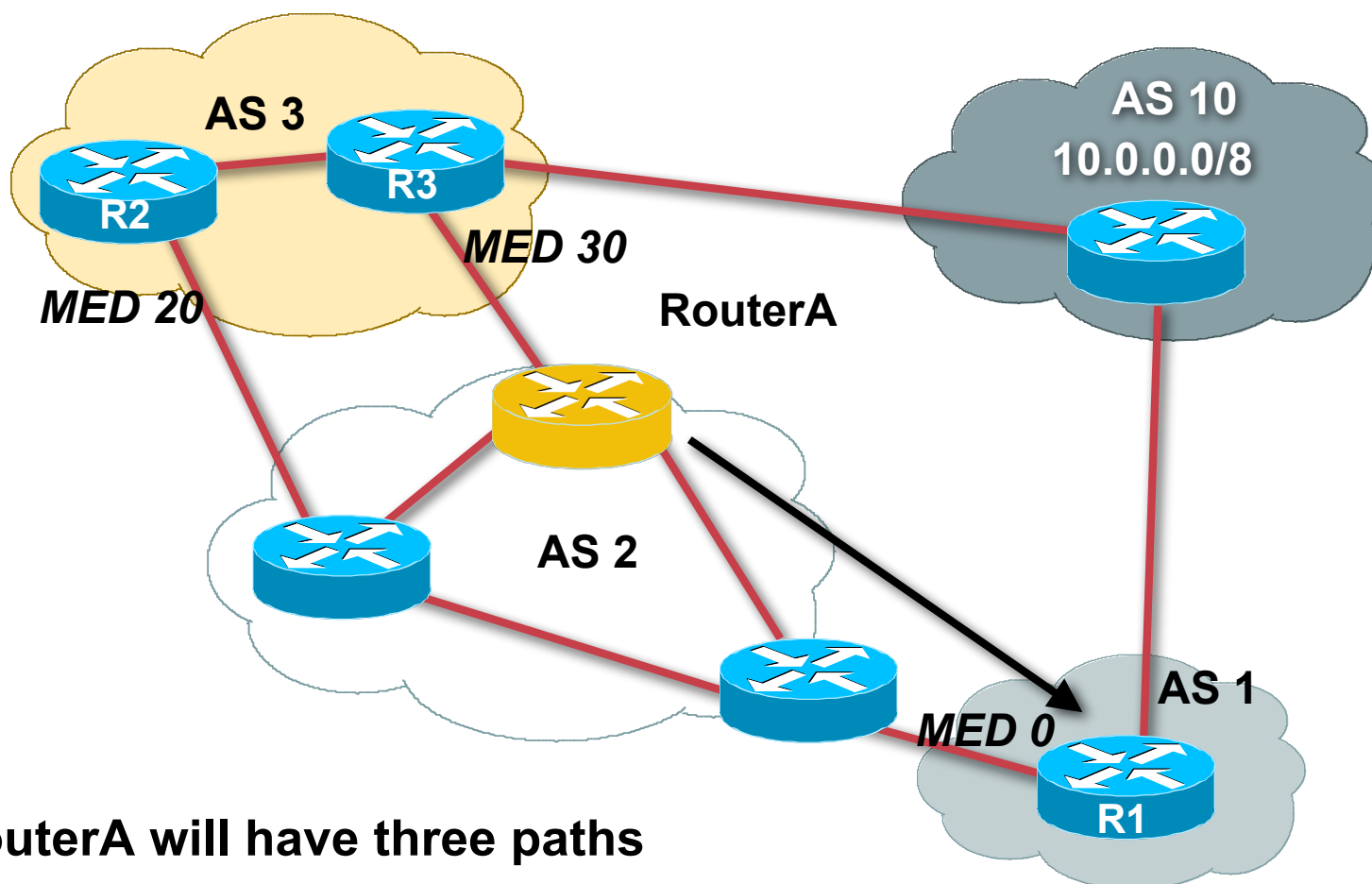
Inconsistent Route Selection: Example I

```
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 Neighbour AS**

- **The bestpath for each Neighbour 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**

Deterministic MED: Summary

- **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 behaviour cannot be changed so the knob must be configured by the user**

Inconsistent Route Selection: Solution – Diagram

AS 3

R2

R3

MED 30

MED 20

RouterA

AS 10
10.0.0.0/8

AS 2

AS 1

MED 0

R1

- **RouterA will have three paths**

- **RouterA will consistently select the path from R1 as best!**

Inconsistent Route Selection: Example II

- **The bestpath changes every time the peering is reset**

AS 10
R1

AS 20
R2

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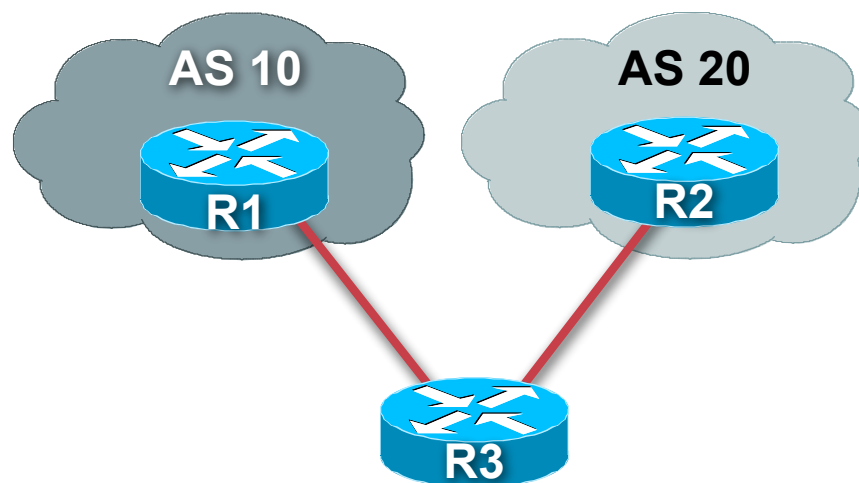
```
R3#show ip bgp 7.0.0.0
BGP routing table entry for 7.0.0.0/8, version 15
  10 100
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, external
  20 100
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Inconsistent Route Selection: 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 Path Selection

- ## Summary:

    RFC1771 wasn't prefect when it came to path selection — years of operational experience have shown this

    Vendors and ISPs have worked to put in stability enhancements, now reflected in RFC4271

    But these can lead to interesting problems

    And of course some defaults linger much longer than they ought to — so never assume that an out of the box default configuration will be perfect for your network

Local Configuration Problems

- **Peer Establishment**

- **Missing Routes**

- **Inconsistent Route Selection**

- **Loops and Convergence Issues**

Route Oscillation

- **One of the most common problems**

- **Main symptom is that traffic exiting the network oscillates every minute between two exit points**

    This is almost *always* caused by the BGP NEXT_HOP being known only by BGP

    Common problem in ISP networks — but if you have never seen it before, it can be a nightmare to debug and fix

- **Other symptom is high CPU utilisation for the BGP router process**

Route Oscillation:
Diagram

R3

R1

AS 3   R2

142.108.10.2

AS 4
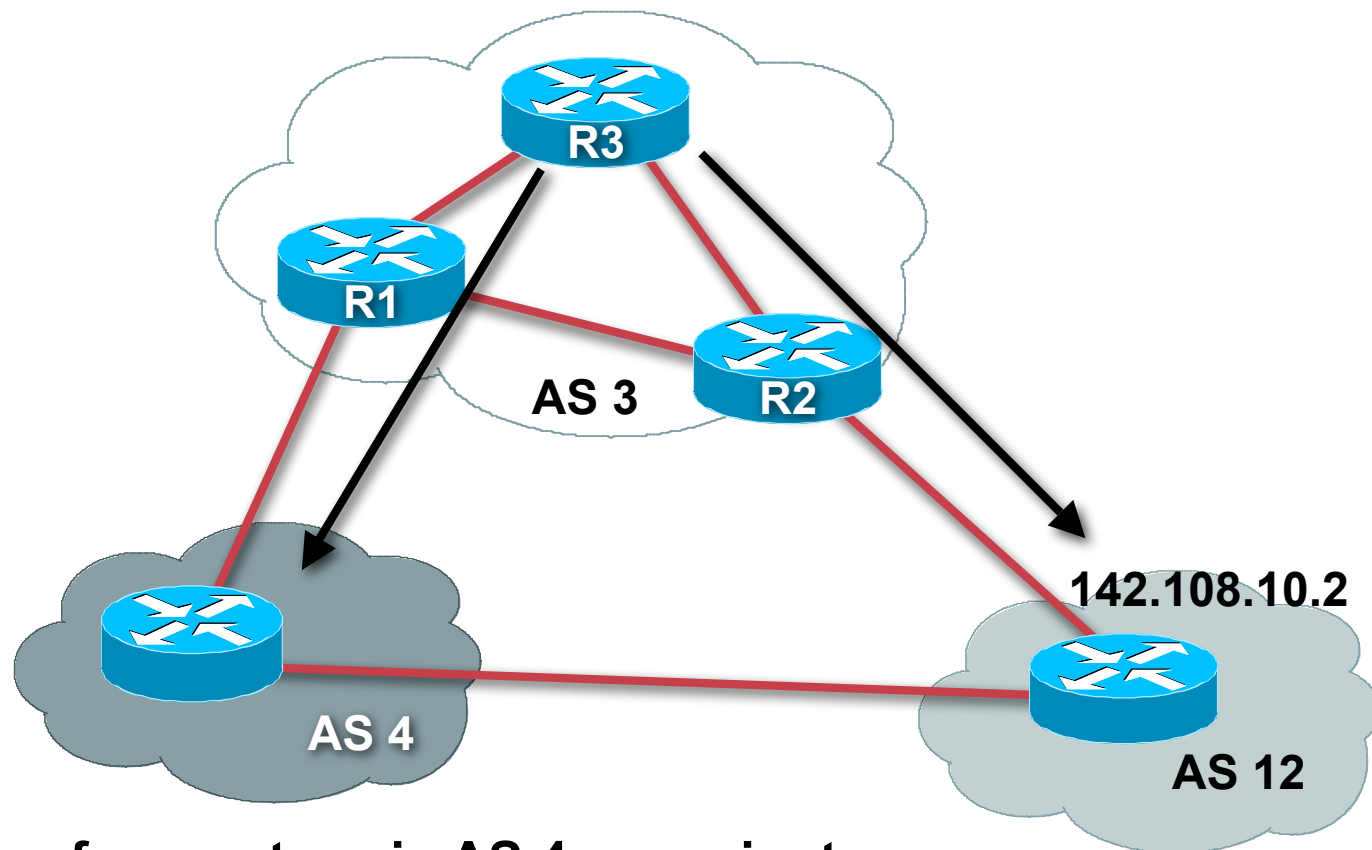
AS 12

- **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: Diagnosis

```
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

- **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]
```
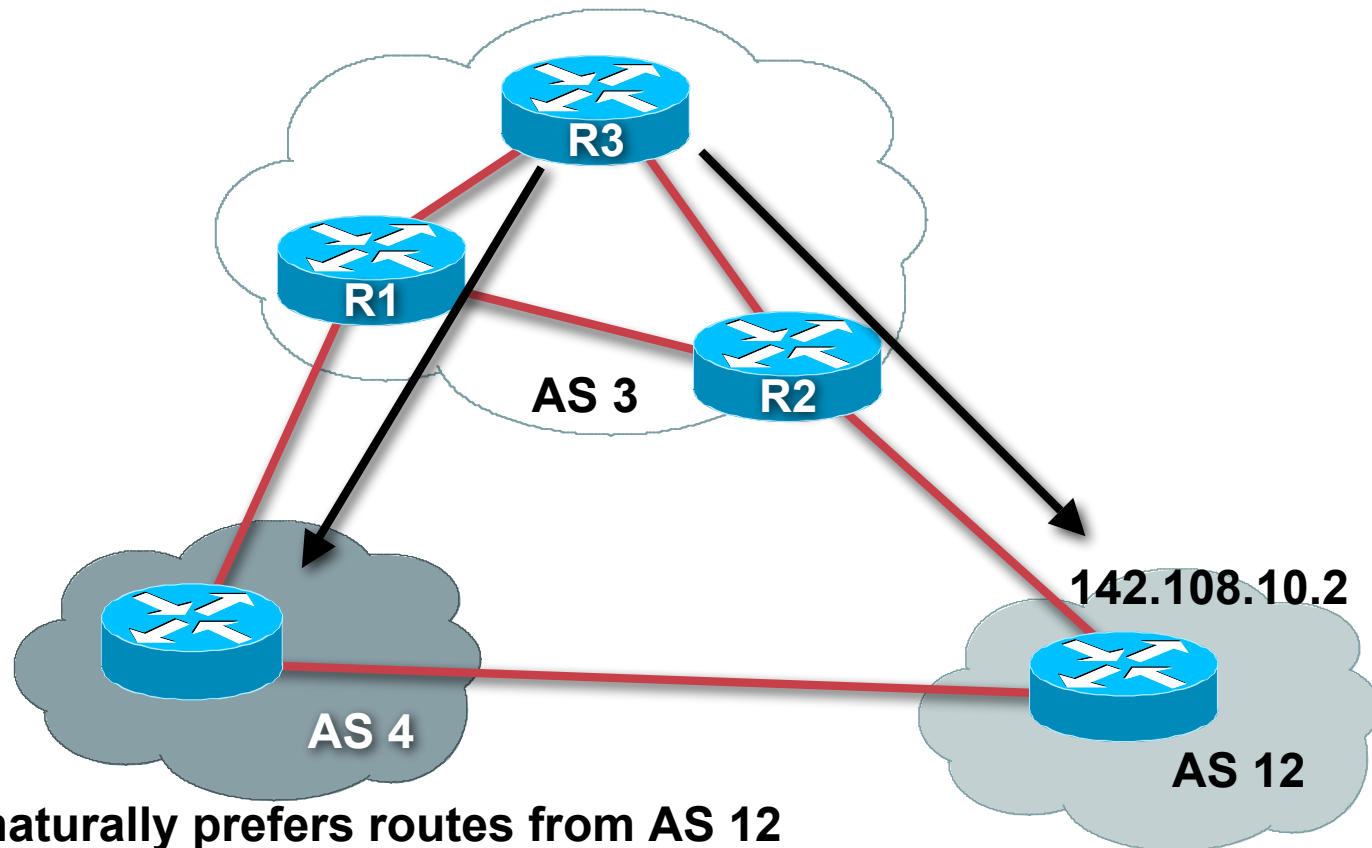
Route Oscillation:
Step by Step

- **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

- **Make sure that all the BGP NEXT_HOPs are known by the IGP**

    (whether OSPF/ISIS, static or connected routes)

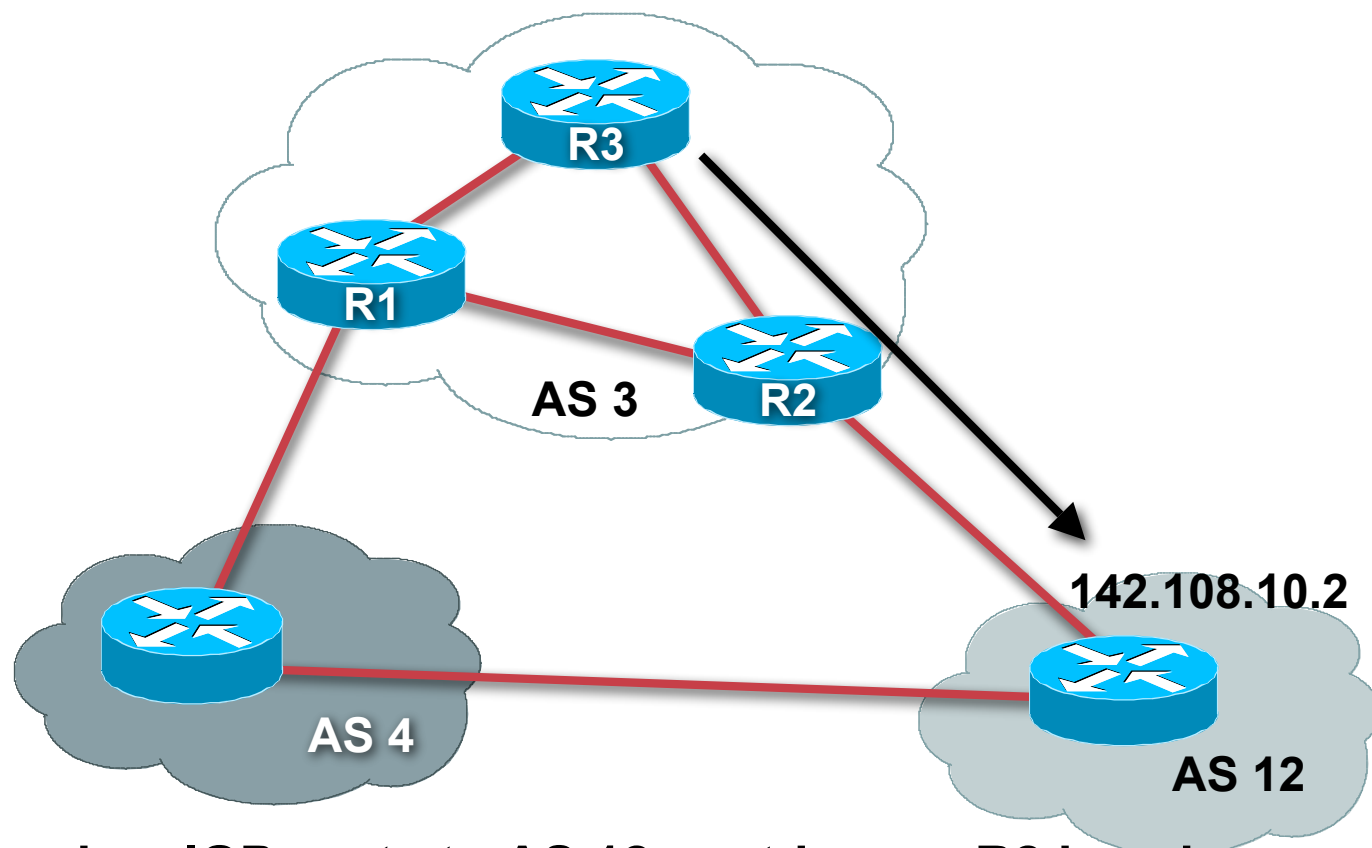   **If NEXT_HOP is also in iBGP, ensure the iBGP distance is longer than the IGP distance**

    —or—

- **Don't carry external NEXT_HOPs in your network**

    Replace eBGP next_hop with local router address on all the edge BGP routers

    (Cisco IOS "next-hop-self")

Route Oscillation: Solution

R3

R1

AS 3   R2

142.108.10.2

AS 4

AS 12

- **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!!**

Troubleshooting Tips

- **High CPU utilisation in the BGP process is normally a sign of a convergence problem**

- **Find a prefix that changes every minute**

- **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**

    The routing table entry

    The BGP table entry

    The route to the NEXT_HOP

Convergence Problems

- **Route reflector with 250 route reflector clients**

- **100k routes**

- **BGP will not converge**
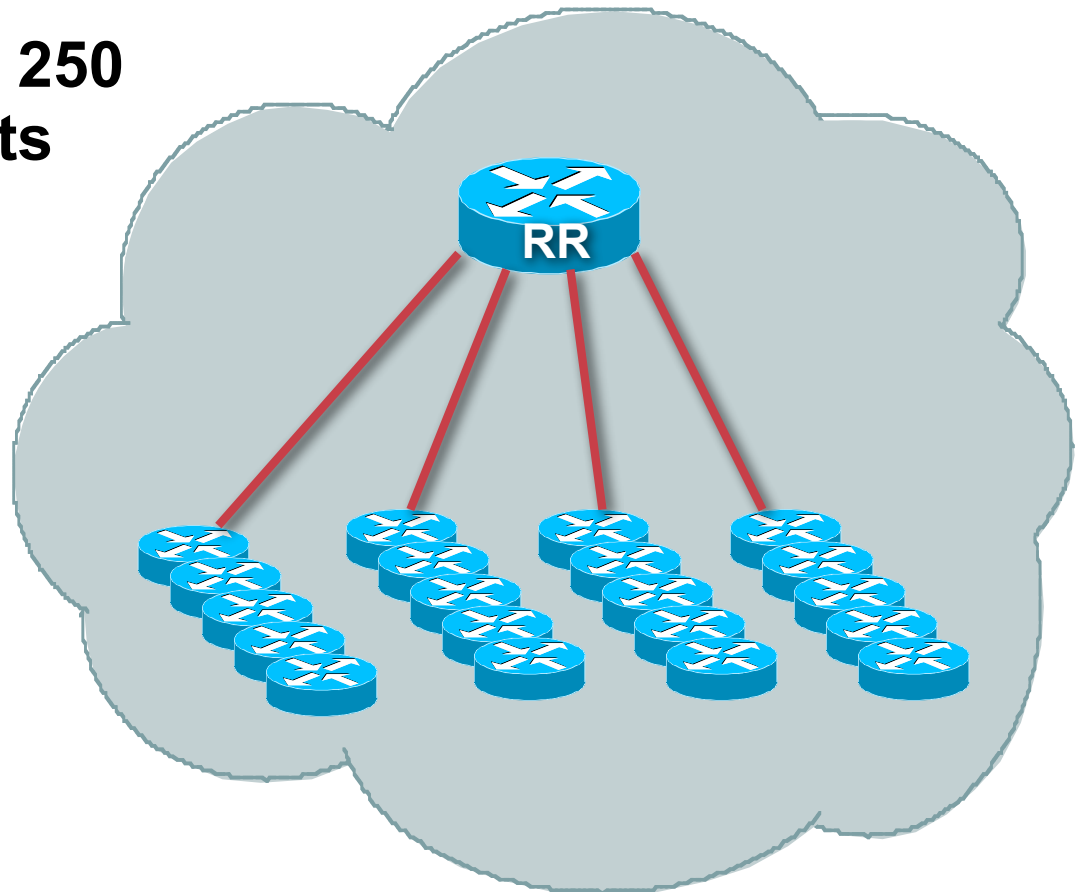
Convergence Problems

- **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 byt
*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 byte
```

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!**

- **More respectable time for 250 peers and 100k routes**

Summary/Tips

- **Use ACLs when enabling debug commands**

- **Ensure that BGP logging is switched on**

- **Ensure that deterministic MED's are enabled**

- **If the entire table is having problem pick one prefix and troubleshoot it**

Agenda

- **Fundamentals**

- **Local Configuration Problems**

- **Internet Reachability Problems**

Internet Reachability Problems

- **BGP Attribute Confusion**

  To Control Traffic in → Send MEDs and AS-PATH prepends on outbound announcements

  To Control Traffic out → Attach local-preference to inbound announcements

- **Troubleshooting of multihoming and transit is often hampered because the relationship between routing information flow and traffic flow is forgotten**

Internet Reachability Problems
BGP Path Selection Process

- **Each vendor has "tweaked" the path selection process**

  Know it for your router equipment — saves time later

  Especially applies with networks with more than one BGP implementation present

  Best policy is to use supplied "knobs" to ensure consistency — and avoid steps in the process which can lead to inconsistency

Internet Reachability Problems
MED Confusion

- **Default MED on Cisco IOS is ZERO**

    It may not be this on your router, or your peer's router

- **Best not to rely on MEDs for multihoming on multiple links to upstream**

    Their default might be $2^{32}-1$ resulting in your hoped for best path being their worst path

    "Workaround", i.e. current good practice, is to use communities rather than MEDs

Internet Reachability Problems
Community Confusion I

- **Set community** in a route-map does just that — it overwrites any other community set on the prefix

  Use **additive** keyword to add community to existing list

- **Use Internet format for community (AS:xx) not the 32-bit IETF format**

  32-bit format is hard for humans to comprehend

  Whereas AS:xx format is more intuitive/recognisable

Internet Reachability Problems
Community Confusion II

- **Cisco IOS never sends community by default**

    Some implementations send community by default for iBGP peerings

    Some implementations also send community by default for eBGP peerings

- **Never assume that your neighbouring AS will honour your no-export community — ask first!**

    If you leak iBGP prefixes to your upstream for loadsharing purposes, this could result in your iBGP prefixes leaking to the Internet

Internet Reachability Problems
AS-PATH prepending

- **20 prepends will not lessen the priority of your path any more than 10 prepends will — check it out at a Looking Glass**

    - The Internet is on average only 5 ASes deep, maximum AS prepend most ISPs have to use is around this too

    - Know you BGP path selection algorithm

- **Some ISPs use bgp maxas-limit 15 to drop prefixes with AS-paths longer than 15 ASNs**

# Internet Reachability Problems
Private ASNs

- **Private ASes should not ever appear in the Internet**

- **Cisco IOS remove-private-AS command does not remove every instance of a private AS**

  e.g. won't remove private AS appearing in the middle of a path surrounded by public ASNs

  www.cisco.com/warp/public/459/32.html

- **Apparent non-removal of private-ASNs may not be a bug, but a configuration error somewhere else**

Troubleshooting Connectivity
Example I

AS 1
192.168.1.0/24
R1

AS 3
R3

R2

AS 2
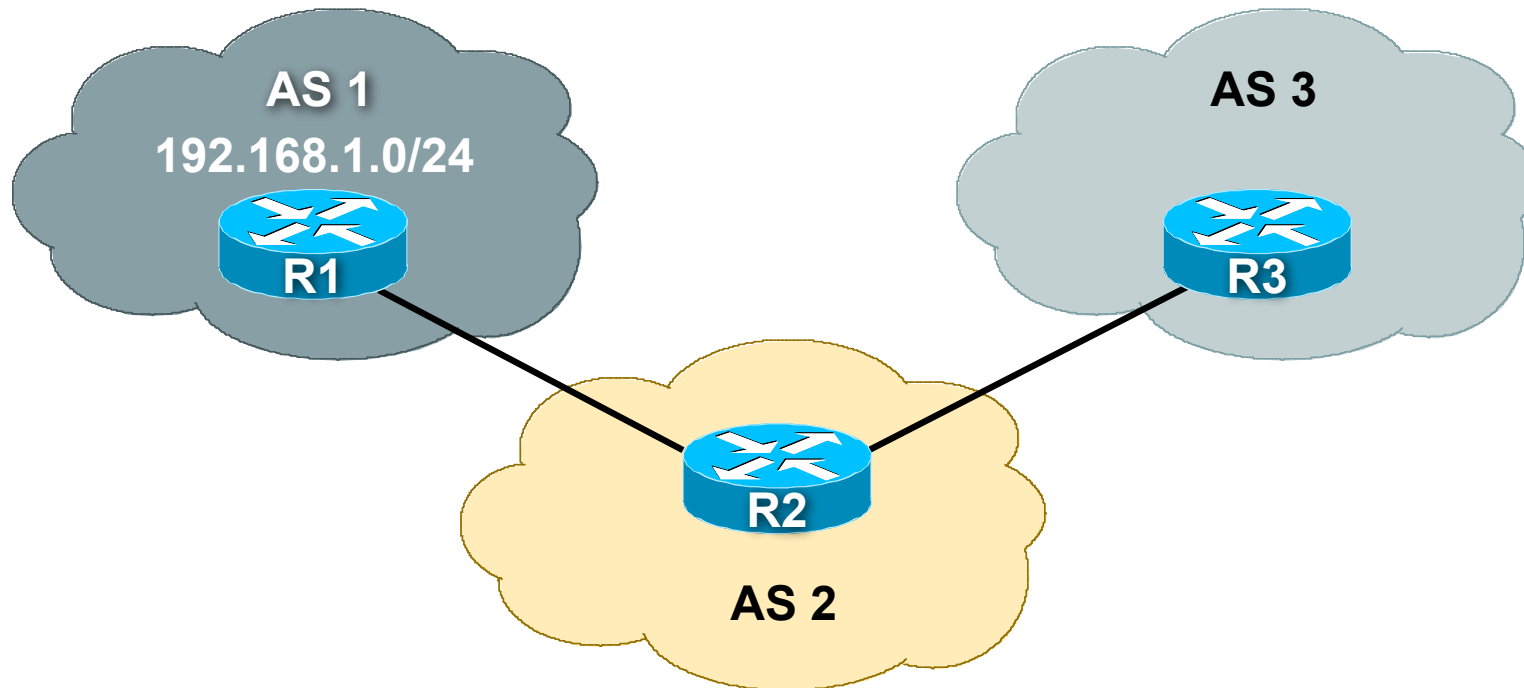
- **Symptom: AS1 announces 192.168.1.0/24 to AS2 but AS3 cannot see the network**

Troubleshooting Connectivity
Example I

- **Checklist:**

   AS1 announces, but does AS2 see it?

   > **We are checking eBGP filters on R1 and R2. Remember that R2 access will require cooperation and assistance from your peer**

   Does AS2 see it over entire network?

   > **We are checking iBGP across AS2's network (unneeded step in this case, but usually the next consideration). Quite often iBGP is misconfigured, lack of full mesh, problems with RRs, etc.**

Troubleshooting Connectivity
Example I

- **Checklist:**

  Does AS2 send it to AS3?

  We are checking eBGP configuration on R2. There may be a configuration error with as-path filters, or prefix-lists, or communities such that only local prefixes get out

  Does AS3 see all of AS2's originated prefixes?

  We are checking eBGP configuration on R3. Maybe AS3 does not know to expect prefixes from AS1 in the peering with AS2, or maybe it has similar errors in as-path or prefix or community filters

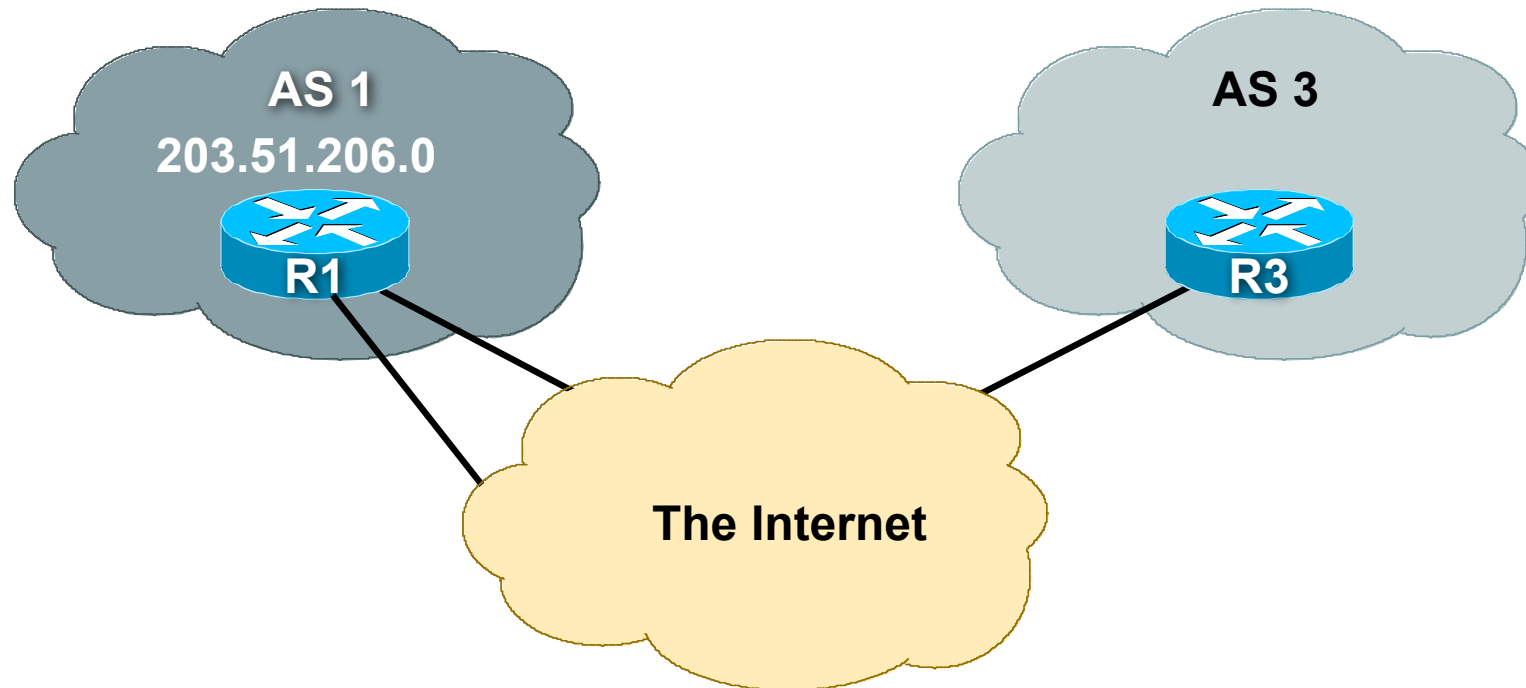# Troubleshooting Connectivity
Example I

- **Troubleshooting connectivity beyond immediate peers is much harder**

  Relies on your peer to assist you — they have the relationship with their BGP peers, not you

  Quite often connectivity problems are due to the private business relationship between the two neighbouring ASNs

Troubleshooting Connectivity Example II

AS 1
203.51.206.0

R1

AS 3

R3

The Internet

- **Symptom: AS1 announces 203.51.206.0/24 to its upstreams but AS3 cannot see the network**

Troubleshooting Connectivity
Example II

- ## Checklist:

  AS1 announces, but do its upstreams see it?

  > **We are checking eBGP filters on R1 and upstreams. Remember that upstreams will need to be able to help you with this**

  Is the prefix visible anywhere on the Internet?

  > **We are checking if the upstreams are announcing the network to anywhere on the Internet. See next slides on how to do this.**

Troubleshooting Connectivity
Example II

- **Help is at hand — the Looking Glass**

- **Many networks around the globe run Looking Glasses**

    These let you see the BGP table and often run simple ping or traceroutes from their sites

    www.traceroute.org for IPv4

    Some IPv6 Looking Glasses listed at www.bgp4.as/looking-glasses

- **Some ISPs, especially those with large and diverse networks, run their own internal Looking Glass to aid internal troubleshooting**

- **Next slides have some examples of a typical looking glass in action**

Routing Information Service

RIPE NCC Homepage -> RIS

RIS:

- **RIS Home Page**
- **Tools**
- **Statistics**
- **RIS Raw Data**
- **Documentation**
- **Presentations**
- **Miscellaneous**
- **News**
- **Contact Us**
- **Disclaimer**

RIS - Looking Glass

RRC Box: RRC00, Amsterdam ▼

RRC00, Amsterdam
RRC01, LINX
RRC02, SFINX
RRC03, AMS-IX
RRC04, CIXP
RRC05, VIX
RRC06, NSPIXP2
RRC07, Netnod
RRC10, MIX
RRC11, NYIIX
RRC12, DE-CIX
RRC14, PAIX

Query:

- ○ show ip
- ● show ip
- ○ show bg
- ○ show ip
- ○ show ipv
- ○ show ipv
- ○ show ipv
- ○ show version
- ○ traceroute
- ○ ping

Argument: [＿＿＿＿＿＿] Execute

Multi-Router Looking Glass
Written by: John Fraizer - EnterZone, Inc

RIS - Looking Glass

RRC Box: RRC01, LINX ▼

Query:
- ⦿ show ip bgp
- ○ show ip bgp summary
- ○ show bgp neighbors
- ○ show ip bgp regexp
- ○ show ipv6 bgp
- ○ show ipv6 bgp summary
- ○ show ipv6 bgp regexp
- ○ show version
- ○ traceroute
- ○ ping

Argument: 202.173.147.0   [Execute]

```
BGP routing table entry for 202.173.144.0/21
Paths: (4 available, best #3, table Default-IP-Routing-Table)
  Not advertised to any peer
  13237 1668 4648 2764 9543
    195.66.224.99 from 195.66.224.99 (82.197.136.1)
      Origin IGP, localpref 100, valid, external
      Community: 1668:31000 13237:44088 13237:46881
      Last update: Fri Jan 14 01:48:12 2005

  286 209 1239 4648 2764 9543
    195.66.224.54 from 195.66.224.54 (134.222.86.174)
      Origin IGP, localpref 100, valid, external
      Last update: Wed Jan  5 13:52:52 2005

  5511 10026 4648 2764 9543
    195.66.224.83 from 195.66.224.83 (193.251.245.1)
      Origin IGP, localpref 100, valid, external, best
      Last update: Mon Jan 17 02:15:07 2005

  8342 702 701 1239 4648 2764 9543
    195.66.224.90 from 195.66.224.90 (195.161.1.152)
      Origin IGP, localpref 100, valid, external
      Last update: Wed Dec 29 00:13:04 2004
```

Multi-Router Looking Glass

Troubleshooting Connectivity
Example II

- **Hmmm….**

- **Looking Glass can see 202.173.144.0/21**

  This includes 202.173.147.0/24

  So the problem must be with AS3, or AS3's upstream

- **A traceroute confirms the connectivity**

- **Tools**
- **Statistics**
- **RIS Raw Data**
- **Documentation**
- **Presentations**
- **Miscellaneous**
- **News**
- **Contact Us**
- **Disclaimer**

RRC Box: `RRC01, LINX`

Query:
- ○ show ip bgp
- ○ show ip bgp summary
- ○ show bgp neighbors
- ○ show ip bgp regexp
- ○ show ipv6 bgp
- ○ show ipv6 bgp summary
- ○ show ipv6 bgp regexp
- ○ show version
- ● traceroute
- ○ ping

Argument: `202.173.147.216` [Execute]

Traceroute from **RRC01** to **202.173.147.216**.

```
traceroute to 202.173.147.216 (202.173.147.216), 30 hops max, 38 byte packets
 1  collector.linx.net (195.66.225.254)  0.752 ms  0.487 ms  0.567 ms
 2  fa2-1-112.transit1.thn.linx.net (195.66.248.226)  0.641 ms  0.778 ms  0.745 ms
 3  demon-transit.thn.linx.net (195.66.248.26)  0.654 ms  0.643 ms  0.518 ms
 4  tele-border-2-g1-0-0.router.demon.net (194.70.98.182)  0.981 ms  1.082 ms  1.212 ms
 5  sl-gw22-lon-2-2.sprintlink.net (213.206.156.49)  0.945 ms  1.105 ms  0.946 ms
 6  sl-bb21-lon-9-0.sprintlink.net (213.206.128.98)  1.117 ms  0.933 ms  1.030 ms
 7  sl-bb21-tuk-10-0.sprintlink.net (144.232.19.69)  73.652 ms  73.803 ms  73.570 ms
 8  sl-bb20-tuk-15-0.sprintlink.net (144.232.20.132)  82.147 ms  81.515 ms  73.878 ms
 9  sl-bb21-rly-14-0.sprintlink.net (144.232.20.115)  81.549 ms  81.799 ms  81.536 ms
10  sl-bb22-rly-13-0.sprintlink.net (144.232.7.254)  81.302 ms  81.898 ms  81.816 ms
11  sl-bb22-sj-10-0.sprintlink.net (144.232.20.186)  143.283 ms  143.680 ms  143.041 ms
12  144.232.20.47 (144.232.20.47)  164.658 ms  148.663 ms  148.483 ms
13  sl-newzeal-1-0.sprintlink.net (144.223.243.18)  151.380 ms  151.648 ms  151.394 ms
14  p5-1.sjbr1.global-gateway.net.nz (202.37.245.229)  306.191 ms  307.392 ms  305.750 ms
15  p1-5.sybr3.global-gateway.net.nz (202.37.247.81)  306.225 ms  306.216 ms  306.239 ms
16  con2.sybr3.global-gateway.net.nz (202.37.246.242)  306.370 ms  307.952 ms  306.693 ms
17  so-3-0-3.crel.syd.connect.com.au (202.10.4.11)  308.144 ms  306.429 ms  307.282 ms
18  so-3-0-2.crel.hay.connect.com.au (202.10.0.63)  306.027 ms  306.267 ms  307.442 ms
19  so-1-1-0.crel.for.connect.com.au (202.10.0.34)  322.587 ms  327.149 ms  325.830 ms
20  so-0-0-1.dst2.bri.connect.com.au (202.10.0.102)  331.707 ms  322.102 ms  322.023 ms
21  gigabitethernet0-1.cor2.bri.connect.com.au (203.63.11.82)  322.028 ms  323.343 ms  323.508 ms
22  DWES131845-8.gw.connect.com.au (210.8.13.61)  325.219 ms  323.865 ms  323.619 ms
23  gi0-1.bri-lns1.qld.westnet.com.au (202.173.144.82)  323.118 ms  323.777 ms  323.458 ms
24  dsl-202-173-147-216.qld.westnet.com.au (202.173.147.216)  337.079 ms  337.940 ms  *
```

Troubleshooting Connectivity
Example II

- **Help is at hand — RouteViews**

- **The main RouteViews router has BGP feeds from around 60 peers**

  www.routeviews.org explains the project

  Gives access to a real router, and allows any provider to find out how their prefixes are seen in various parts of the Internet

  Complements the Looking Glass facilities

- **Anyway, back to our problem…**

Troubleshooting Connectivity
Example II

- **Checklist:**

   Does AS3's upstream send it to AS3?

   We are checking eBGP configuration on AS3's upstream. There may be a configuration error with as-path filters, or prefix-lists, or communities such that only local prefixes get out. This needs AS3's assistance

   Does AS3 see any of AS1's originated prefixes?

   We are checking eBGP configuration on R3. Maybe AS3 does not know to expect the prefix from AS1 in the peering with its upstream, or maybe it has some errors in as-path or prefix or community filters

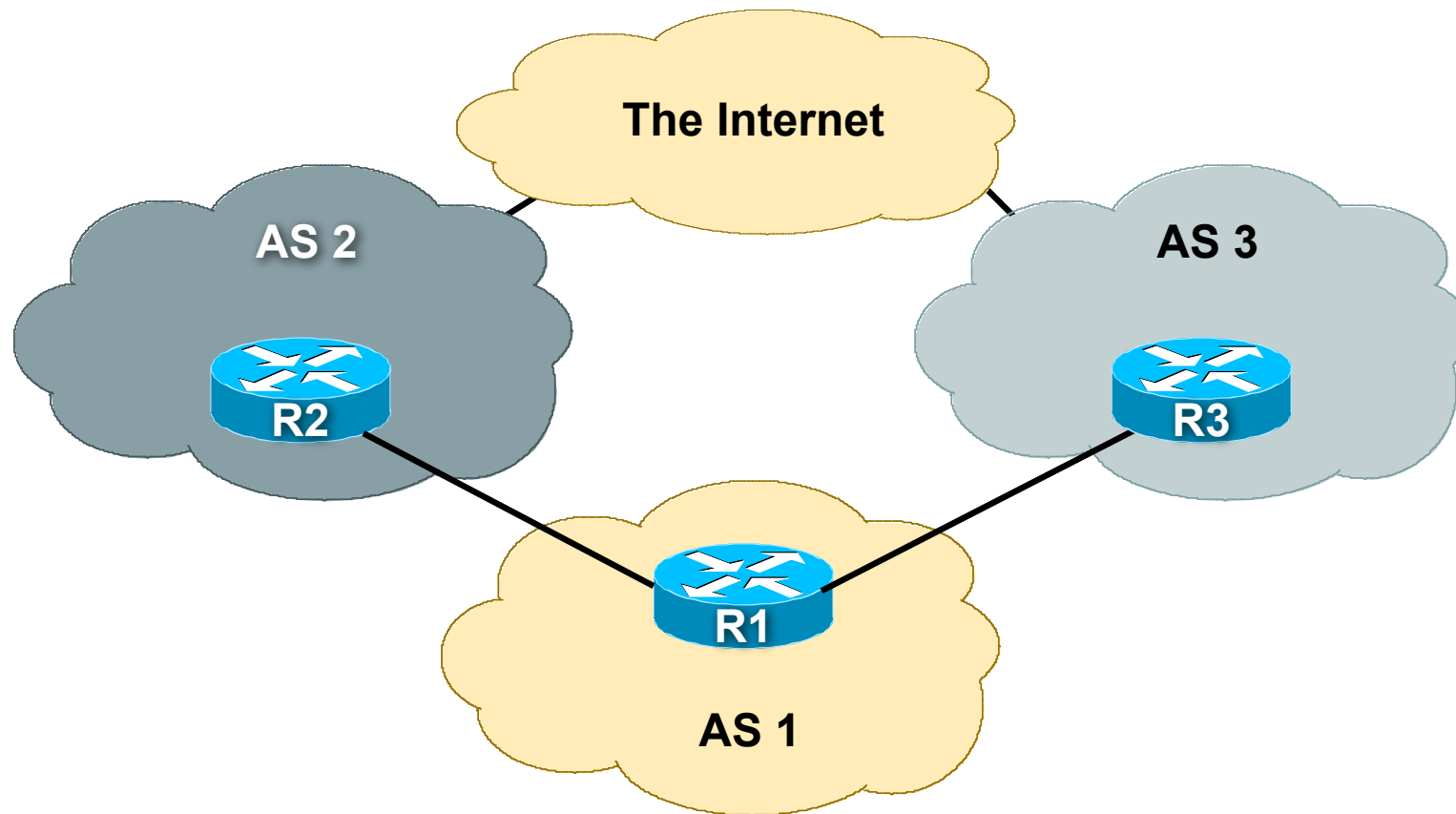# Troubleshooting Connectivity
Example II

- **Troubleshooting across the Internet is harder**

    But tools are available

- **Looking Glasses, offering traceroute, ping and BGP status are available all over the globe**

    Most connectivity problems seem to be found at the edge of the network, rarely in the transit core

    Problems with the transit core are usually intermittent and short term in nature

Troubleshooting Connectivity
Example III

The Internet

AS 2

R2

AS 3

R3

R1

AS 1

- **Symptom: AS1 is trying to loadshare between its upstreams, but has trouble getting traffic through the AS2 link**

Troubleshooting Connectivity
Example III

- **Checklist:**

    What does "trouble" mean?

- **Is outbound traffic loadsharing okay?**

    Can usually fix this with selectively rejecting prefixes, and using local preference

    Generally easy to fix, local problem, simple application of policy

- **Is inbound traffic loadsharing okay?**

    Errummm, bigger problem if not

    Need to do some troubleshooting if configuration with communities, AS-PATH prepends, MEDs and selective leaking of subprefixes don't seem to help

Troubleshooting Connectivity
Example III

- ## Checklist:

    AS1 announces, but does AS2 see it?

    We are checking eBGP filters on R1 and R2. Remember that R2 access will require cooperation and assistance from your peer

    Does AS2 see it over entire network?

    We are checking iBGP across AS2's network. Quite often iBGP is misconfigured, lack of full mesh, problems with RRs, etc.

Troubleshooting Connectivity
Example III

- **Checklist:**

    Does AS2 send it to its upstream?

    We are checking eBGP configuration on R2. There may be a configuration error with as-path filters, or prefix-lists, or communities such that only local prefixes get out

    Does the Internet see all of AS2's originated prefixes?

    We are checking eBGP configuration on other Internet routers. This means using looking glasses. And trying to find one as close to AS2 as possible.

Troubleshooting Connectivity
Example III

- **Checklist:**

  Repeat all of the above for AS3

- **Stopping here and resorting to a huge prepend towards AS3 won't solve the problem**

- **There are many common problems — listed on next slide**

  And tools to help decipher the problem

Troubleshooting Connectivity
Example III

- ## No inbound traffic from AS2

  AS2 is not seeing AS1's prefix, or is blocking it in inbound filters

- ## A trickle of inbound traffic

  Switch on NetFlow (if the router has it) and check the origin of the traffic

  If it is just from AS2's network blocks, then is AS2 announcing the prefix to its upstreams?

  If they claim they are, ask them to ask their upstream for a "show ip bgp" output — or use a Looking Glass to check

- **A light flow of traffic from AS2, but 50% less than from AS3**

  Looking Glass comes to the rescue

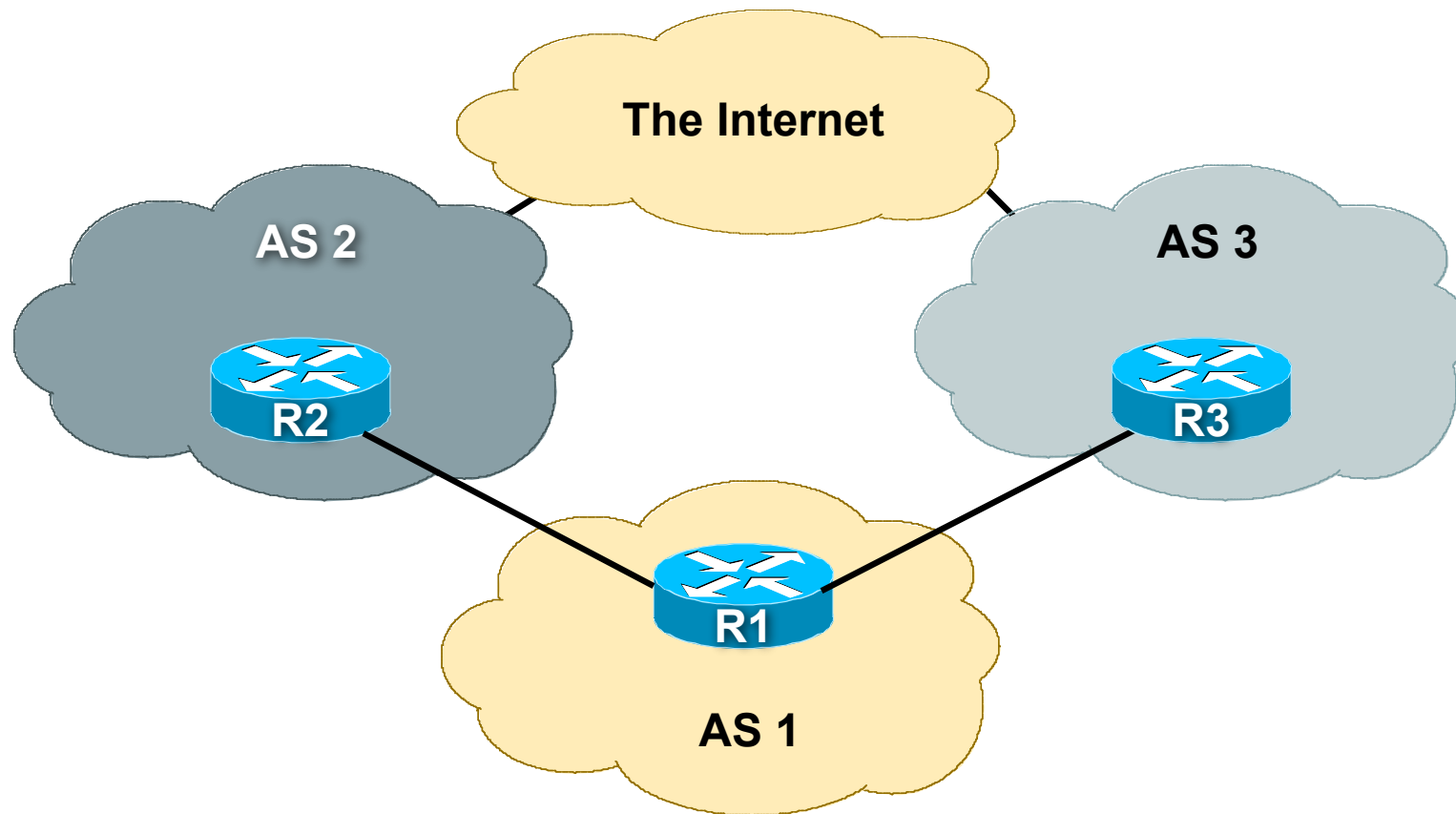 LG will let you see what AS2, or AS2's upstreams are announcing

  AS1 may choose this as primary path, but AS2 relationship with their upstream may decide otherwise

  NetFlow comes to the rescue

  Allows AS1 to see what the origins are, and with the LG, helps AS1 to find where the prefix filtering culprit might be

Troubleshooting Connectivity
Example IV

- **Symptom: AS1 is loadsharing between its upstreams, but the traffic load swings randomly between AS2 and AS3**

Troubleshooting Connectivity
Example IV

- **Checklist:**

  Assume AS1 has done everything in this tutorial so far

  > **All the configurations look fine, the Looking Glass outputs look fine, life is wonderful… Apart from those annoying traffic swings every hour or so**

  L2 problem? Route Flap Damping?

  > **Since BGP is configured fine, and the net has been stable for so long, can only be an L2 problem, or Route Flap Damping side-effect**

Troubleshooting Connectivity
Example IV

- **L2 — upstream somewhere has poor connectivity between themselves and the rest of the Internet**

    Only real solution is to impress upon upstream that this isn't good enough, and get them to fix it

    Or change upstreams

Troubleshooting Connectivity
Example IV

- ## Route Flap Damping

  ### Many ISPs implement route flap damping

  ### Many ISPs simply use the vendor defaults

  ### Vendor defaults are generally far too severe

  ### There is real concern that the "more lenient" RIPE-229 values are too severe

  ### Opinion is growing that flap damping does more harm than good

    e.g. **www.cs.berkeley.edu/~zmao/Papers/sig02.pdf**

- ## Again Looking Glasses come to the operator's assistance

Query Results:

```
sl-bb20-sj>sh ip bgp flap
 % NOTE: This command will be deprecated soon. Please use 'show ip bgp dampening [dampened-paths|flap-statistics]'
 BGP table version is 87689246, local router ID is 144.228.241.64
 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          From            Flaps Duration Reuse    Path
  h 12.44.243.0/24    144.232.9.2     1     00:13:12          701 26144
  h 12.104.113.0/24   144.232.9.2     1     00:45:12          701 27358 27358 27358 27358 27358
  h 12.104.114.0/24   144.232.9.2     1     00:45:12          701 27358 27358 27358 27358 27358
  h 12.108.254.0/24   144.232.9.2     1     00:26:32          701 6389 6197 26829
  h 15.130.192.0/20   144.232.9.2     1     00:52:38          701 1273 1889
  h 15.195.176.0/20   144.232.9.2     1     00:52:28          701 1273 1889
  h 15.197.192.0/18   144.232.9.2     1     00:52:28          701 1273 1889
  h 15.198.0.0/17     144.232.9.2     1     00:52:38          701 1273 1889
  h 15.203.128.0/18   144.232.9.2     1     00:52:38          701 1273 1889
  h 15.204.96.0/19    144.232.9.2     1     00:52:28          701 1273 1889
  h 15.204.128.0/17   144.232.9.2     1     00:52:28          701 1273 1889
  h 16.0.0.0/12       144.232.9.2     1     00:52:28          701 1273 1889 1889
  h 16.6.0.0/15       144.232.9.2     1     00:52:38          701 1273 1889
  h 16.8.0.0/15       144.232.9.2     1     00:52:38          701 1273 1889
  h 16.14.0.0/15      144.232.9.2     1     00:52:38          701 1273 1889
  *   59.81.0.0/18    144.232.9.2     20    05:01:05          701 9800 17773
  *   59.81.64.0/18   144.232.9.2     20    05:01:05          701 9800 17773
  *   59.81.128.0/18  144.232.9.2     20    05:01:05          701 9800 17773
  *   59.81.192.0/18  144.232.9.2     20    05:01:05          701 9800 17773
  *   59.82.0.0/18    144.232.9.2     20    05:01:05          701 9800 17773
  *   59.82.64.0/18   144.232.9.2     20    05:01:05          701 9800 17773
  *   59.82.128.0/18  144.232.9.2     20    05:01:05          701 9800 17773
  *   59.82.192.0/18  144.232.9.2     20    05:01:05          701 9800 17773
  *   59.83.0.0/18    144.232.9.2     20    05:01:05          701 9800 17773
  *   59.83.64.0/18   144.232.9.2     20    05:01:05          701 9800 17773
  *   59.83.128.0/18  144.232.9.2     20    05:01:05          701 9800 17773
  *   59.83.192.0/18  144.232.9.2     20    05:01:05          701 9800 17773
  *>  61.1.176.0/20   144.232.9.2     1     00:33:24          701 22351 4755 9829
  *>  61.1.192.0/19   144.232.9.2     1     00:33:24          701 22351 4755 9829
  *>  61.2.208.0/20   144.232.9.2     1     00:33:24          701 22351 4755 9829
  *>  61.3.224.0/20   144.232.9.2     1     00:33:24          701 22351 4755 9829
  *   62.24.32.0/22   144.232.9.2     2     00:33:45          701 22351
  *   62.24.36.0/24   144.232.9.2     2     00:33:45          701 22351
```

Troubleshooting Connectivity
Example IV

- **Several Looking Glasses allow the operators to check the flap or damped status of their announcements**

  Many oscillating connectivity issues are usually caused by L2 problems

  Route flap damping will cause connectivity to persist via alternative paths even though primary paths have been restored

  Quite often, the exponential back off of the flap damping timer will give rise to bizarre routing

  Common symptom is that bizarre routing will often clear away by itself

Troubleshooting Summary

- **Most troubleshooting is about:**

- **Experience**

  Recognising the common problems

- **Not panicking**

- **Logical approach**

  Check configuration first

  Check locally first before blaming the peer

  Troubleshoot layer 1, then layer 2, then layer 3, etc

Troubleshooting Summary

- **Most troubleshooting is about:**

- **Using the available tools**

  The debugging tools on the router hardware

  Internet Looking Glasses

  Colleagues and their knowledge

  Public mailing lists where appropriate

Closing Comments

- **Presentation has covered the most common troubleshooting techniques used by ISPs today**

- **Once these have been mastered, more complex or arcane problems are easier to solve**

- **Feedback and input for future improvements is encouraged and very welcome**

Troubleshooting BGP

Philip Smith   <pfs@cisco.com>

NANOG 36

12-15 February 2006

Dallas