Troubleshooting BGP

ISP Workshops

Last updated 21st May 2025

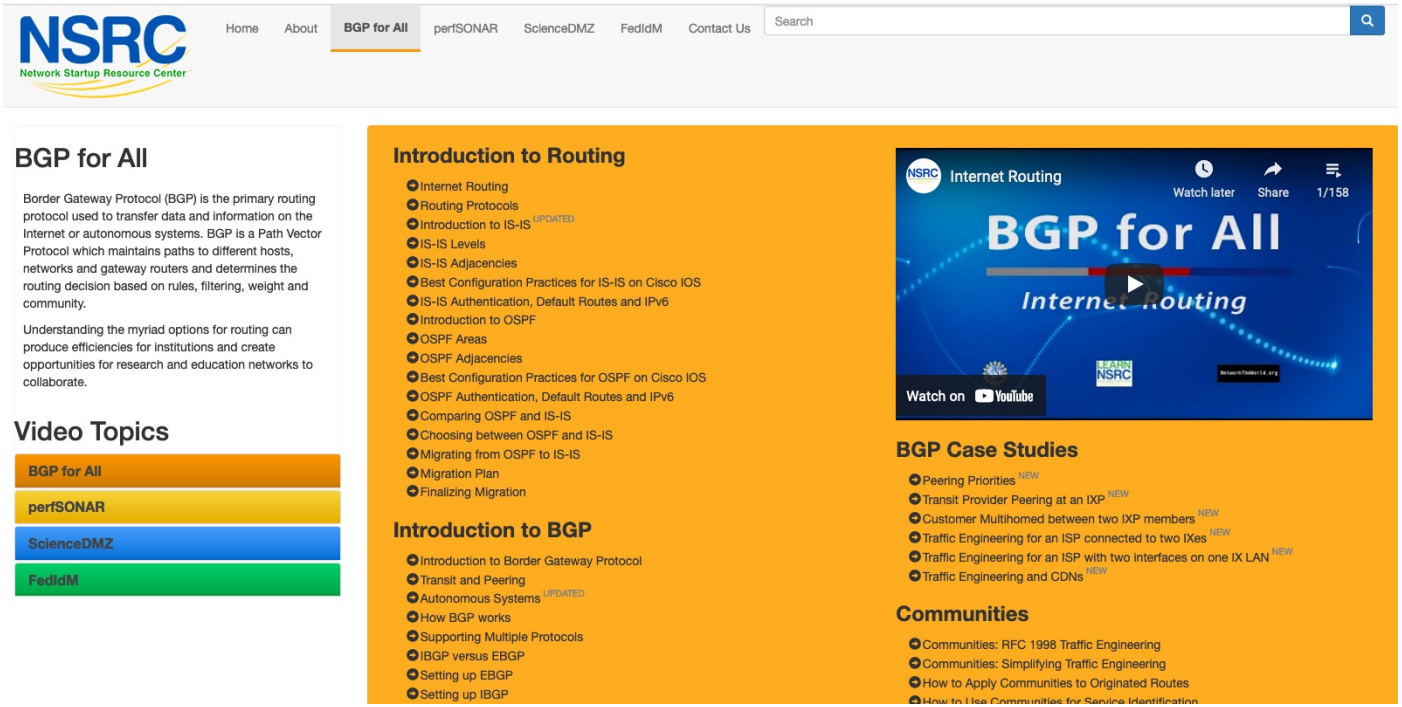
Acknowledgements

- This material originated from the Cisco ISP/IXP Workshop Programme developed by Philip Smith & Barry Greene and is based on the Cisco Networkers BGP Troubleshooting session
- Use of these materials is encouraged as long as the source is fully acknowledged, and this notice remains in place
- Bug fixes and improvements are welcomed
  - Please email *workshop (at) bgp4all.com*

Philip Smith

BGP Videos

- NSRC has made a video recording of many presentations to form a library of BGP videos for the whole community to use:
  - https://learn.nsrc.org/bgp



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
  - Most BGP implementations today have a Cisco IOS style 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, miscommunication between peer network operators
  - 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 the original RFC1771 and RFC4271
    - "Improvements" introduced by the software implementers
    - Incomplete or incorrect implementation of BGP standards and standardised BGP capabilities
  - 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,
  - Move on to the next,
  - Repeat until solved.

Fundamentals: Working on Solutions

- Remember! Troubleshooting is about:
  - Not panicking
  - Creating a checklist
  - Working to that checklist
  - Starting at the bottom and working up
- When problem is solved, document the checklist and the circumstances that caused the problem
  - Will help colleagues in the future deal with the issue when it reoccurs

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
    - Loghost? Are the logs easily searchable?
  - Last known good configuration
    - Stored off the router!
  - History trail of working configurations and all intermediate changes
    - Change recording/management system?
  - Record of commands entered on routers and other network devices
    - TACACS+ (or something else)?

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

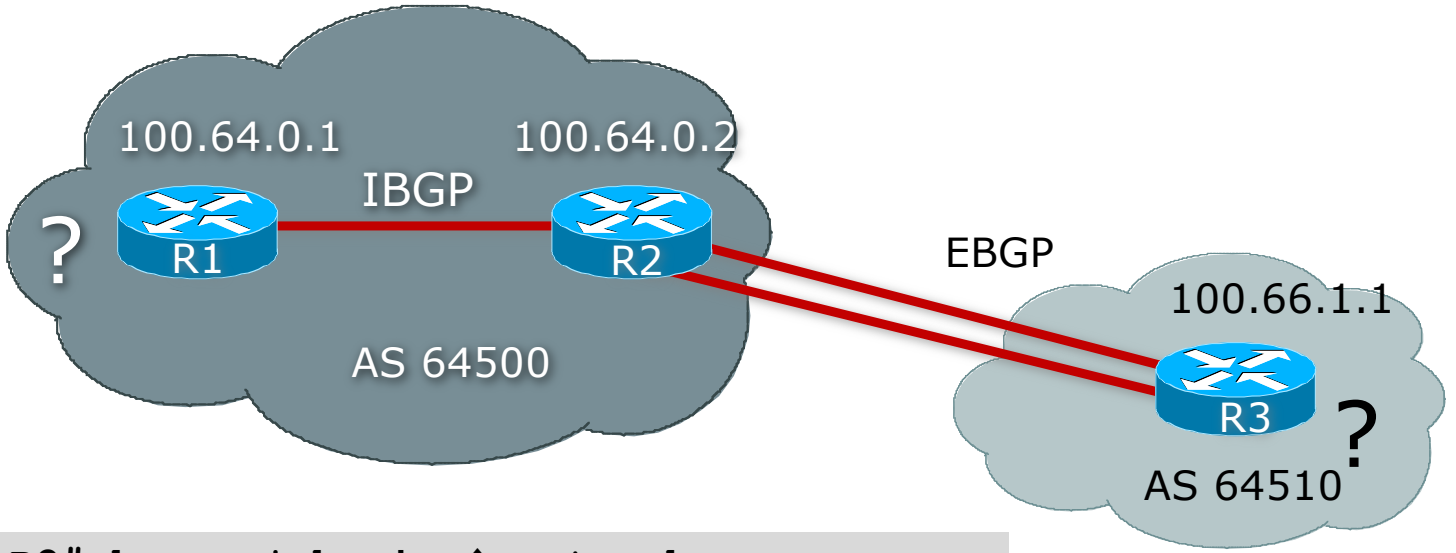
Peer Establishment

- Routers establish a TCP session
  - BGP uses port 179 – Permit in interface filters
  - IP connectivity (route from IGP)
- OPEN messages are exchanged
  - Both peers will attempt to initiate the peering session
    - The one that is first will likely prevail (random delay to start)
  - Peering addresses must match the TCP session
    - Although not all implementations enforce this
  - Local AS configuration parameters
    - Peer ASN must match what is configured

Common Problems

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

Peer Establishment: Diagram




```
R2#sh run | begin ^router bgp
router bgp 64500
 bgp log-neighbor-changes
 neighbor 100.64.0.1 remote-as 64500
 neighbor 100.66.1.1 remote-as 64510
```

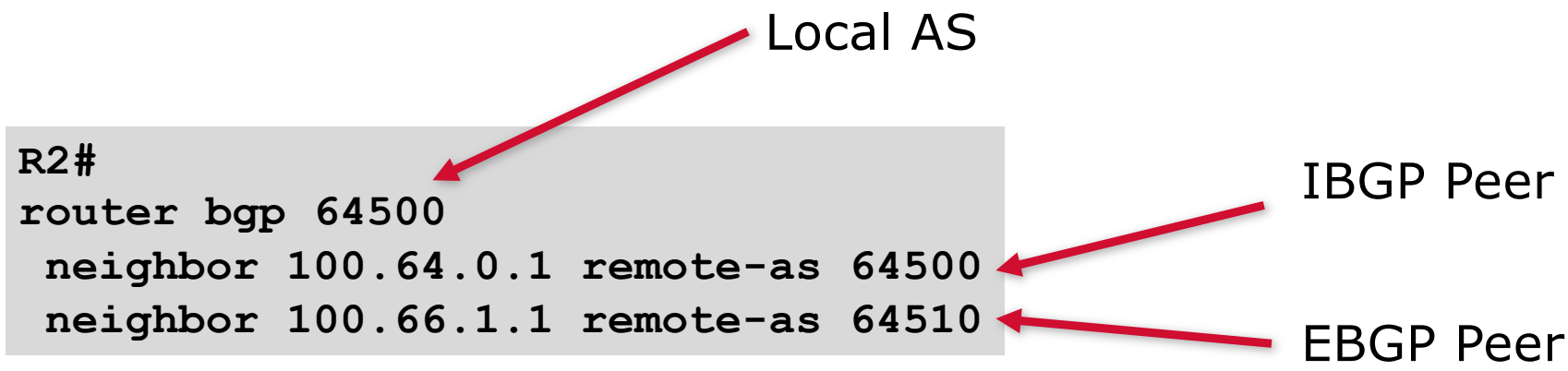
Peer Establishment: Symptoms

```
R2#show ip bgp summary
BGP router identifier 100.64.0.2, local AS number 64500
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State
100.64.0.1      4        64500       0       0        0    0    0 never    Active
100.66.1.1      4        64510       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          100.66.1.1.*       LISTEN
0063F3D4 *.179          100.64.0.1.*       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 100.66.255.1:26385 from 100.64.0.2:179
```

- Remote Is Trying to Open the Session from 100.66.255.1 Address…

Peer Establishment: IBGP

- What about us?

```
R2#debug ip bgp
BGP debugging is on
R2#
BGP: 100.64.0.1 open active, local address 100.66.255.2
BGP: 100.64.0.1 open failed: Connection refused by remote host
```

- We are trying to open the session from 100.66.255.2 address…

```
R2#sh ip route 100.64.0.1
Routing entry for 100.64.0.1/32
  Known via "static", distance 1, metric 0 (connected)
  * directly connected, via GigabitEthernet0
      Route metric is 0, traffic share count is 1

R2#show ip interface brief | include Gigabit
GigabitEthernet0    100.66.255.2     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

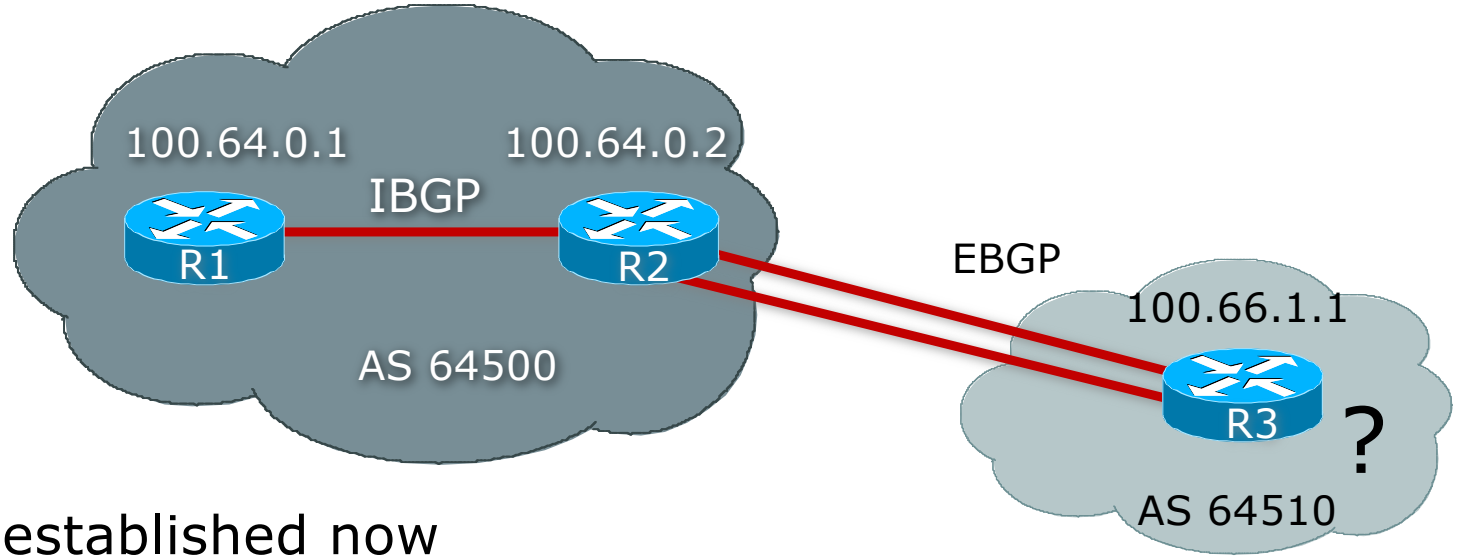
```
R2#
router bgp 64500
 neighbor 100.64.0.1 remote-as 64500
 neighbor 100.64.0.1 update-source Loopback0
 neighbor 100.66.1.1 remote-as 64510
 neighbor 100.66.1.1 update-source Loopback0
```

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 100.66.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.66.1.1, 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: 100.66.1.1
Extended commands [n]: y
Source address or interface: 100.64.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.66.1.1, 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, 100.64.0.2

Peer Establishment: EBGP

- Assume R3 added a route to 100.64.0.2
- Still having problems…

```
R2#sh ip bgp neigh 100.66.1.1
BGP neighbor is 100.66.1.1,  remote AS 64510, 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 64500
 neighbor 100.66.1.1 remote-as 64510
 neighbor 100.66.1.1 ebgp-multihop 2
 neighbor 100.66.1.1 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 100.64.0.2, local AS number 64500

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State
100.66.1.1      4        64510      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: 100.66.1.1 open active, local address 100.64.0.2
14:06:37: BGP: 100.66.1.1 went from Active to OpenSent
14:06:37: BGP: 100.66.1.1 sending OPEN, version 4
14:06:37: BGP: 100.66.1.1 received NOTIFICATION 2/2
          (peer in wrong AS) 2 bytes FBF4
14:06:37: BGP: 100.66.1.1 remote close, state CLOSEWAIT
14:06:37: BGP: service reset requests
14:06:37: BGP: 100.66.1.1 went from OpenSent to Idle
14:06:37: BGP: 100.66.1.1 closing
```

- If an error is detected, a notification is sent, and the session is closed
- R3 is configured incorrectly
  - Has "`neighbor 100.64.0.2 remote-as 64600`"
  - Should have "`neighbor 100.64.0.2 remote-as 64500`"
- After R3 makes this correction, the session should come up
- Note: the debug error lists the ASN expected, but in hexadecimal

Peer Establishment: EBGP – Summary

- Remember to allow TCP/179 through edge filters

```
access-list 100 permit tcp host 100.66.1.1 eq 179 host 100.64.0.2
access-list 100 permit tcp host 100.66.1.1 host 100.64.0.2 eq 179
```

- Be very careful with multihop EBGP
  - Check IP connectivity (local and remote routing tables)
  - Remember to source updates from loopback
  - Watch for TCP/179 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 100.64.0.2, local AS number 64500

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
100.66.1.1      4        64510      10      26        0    0    0 never    Active
```

Peer Establishment: Passwords

```
R2#
router bgp 1
 neighbor 100.66.1.1 remote-as 64510
 neighbor 100.66.1.1 ebgp-multihop 2
 neighbor 100.66.1.1 update-source Loopback0
 neighbor 100.66.1.1 password 7 05080F1C221C
```

- Configuration on R2 looks fine!
- Check the log messages – enable **`log-neighbor-changes`**

```
%TCP-6-BADAUTH: No MD5 digest from 100.66.1.1:179 to 100.64.0.2:11272
%TCP-6-BADAUTH: No MD5 digest from 100.66.1.1:179 to 100.64.0.2:11272
%TCP-6-BADAUTH: No MD5 digest from 100.66.1.1:179 to 100.64.0.2:11272
```

Peer Establishment: Passwords

```
R3#
router bgp 64510
 neighbor 100.64.0.2 remote-as 64500
 neighbor 100.64.0.2 ebgp-multihop 2
 neighbor 100.64.0.2 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 100.66.1.1:11024 to 100.64.0.2:179
%TCP-6-BADAUTH: Invalid MD5 digest from 100.66.1.1:11024 to 100.64.0.2:179
%TCP-6-BADAUTH: Invalid MD5 digest from 100.66.1.1:11024 to 100.64.0.2:179
```

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

Peer Establishment: Passwords

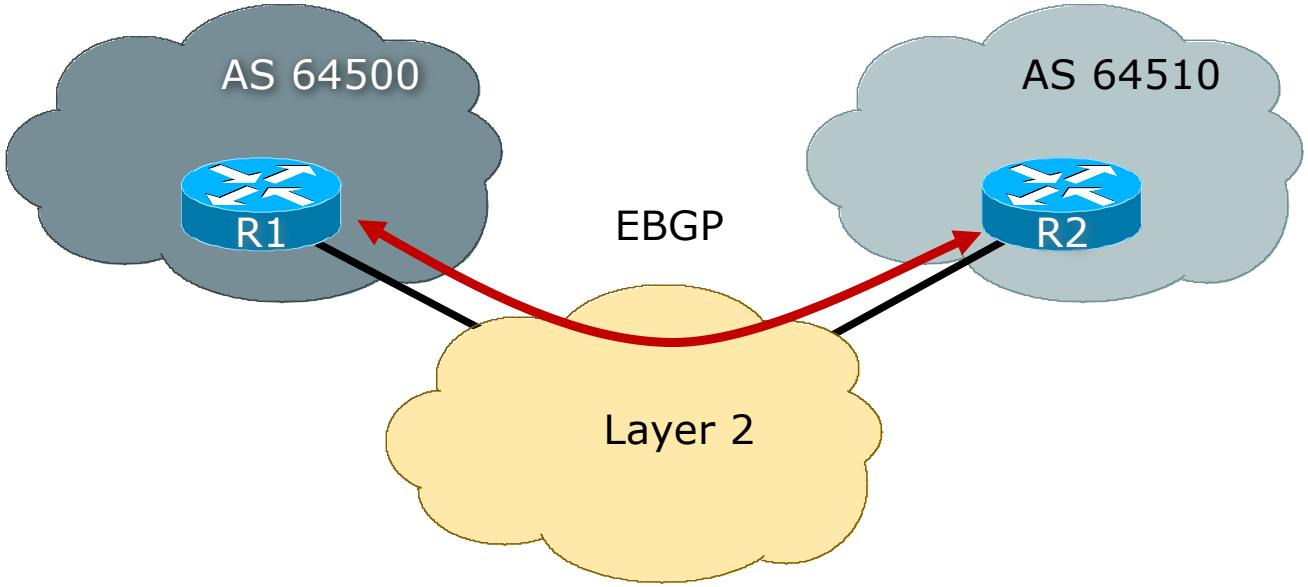
- We must have mis-typed the password on one of the peering routers
  - Fix the password – best to re-enter password on both routers
  - EBGP session now comes up

```
%TCP-6-BADAUTH: Invalid MD5 digest from 100.66.1.1:11027 to 100.64.0.2:179
%BGP-5-ADJCHANGE: neighbor 100.66.1.1 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 and paste

Flapping Peer: Common Symptoms



- 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 100.64.0.1 Down BGP Notification sent
%BGP-3-NOTIFICATION: sent to neighbor 100.64.0.1 4/0 (hold time expired) 0 bytes


R2#show ip bgp neighbor 100.64.0.1 | 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!
  - The clue is “hold time expired”

Flapping Peer

```
R1#show ip bgp summary
BGP router identifier 100.69.175.53, local AS number 64500
BGP table version is 10167, main routing table version 10167

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
100.64.0.2      4        64510      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
100.64.0.2      4        64510      53      284    10167    0    98 00:03:04        0
```

- Let's look at our peer more closely!
- Hellos are stuck in OutQ behind update packets!
- Notice that the MsgSent counter has not moved

Flapping Peer

```
R1#ping 100.64.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.64.0.2, 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: 100.64.0.2
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 100.64.0.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

- Normal pings work but a 1500byte ping fails?

Flapping Peer

- Trying reducing the ping packet size:

```
R1#ping ip
Target IP address: 100.64.0.2
Repeat count [5]:
Datagram size [100]: 1400
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1400-byte ICMP Echos to 100.64.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 46/51/54 ms
```

- Success! The link is allowing MTU 1400, but not MTU 1500.
- Try repeating until you find which MTU size fails
  - Now know where to fault find in the L2 infrastructure!

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
  - Try pinpoint the MTU size that fails, to help fault find in L2
  - Some BGP implementations may support setting the TCP MSS value:

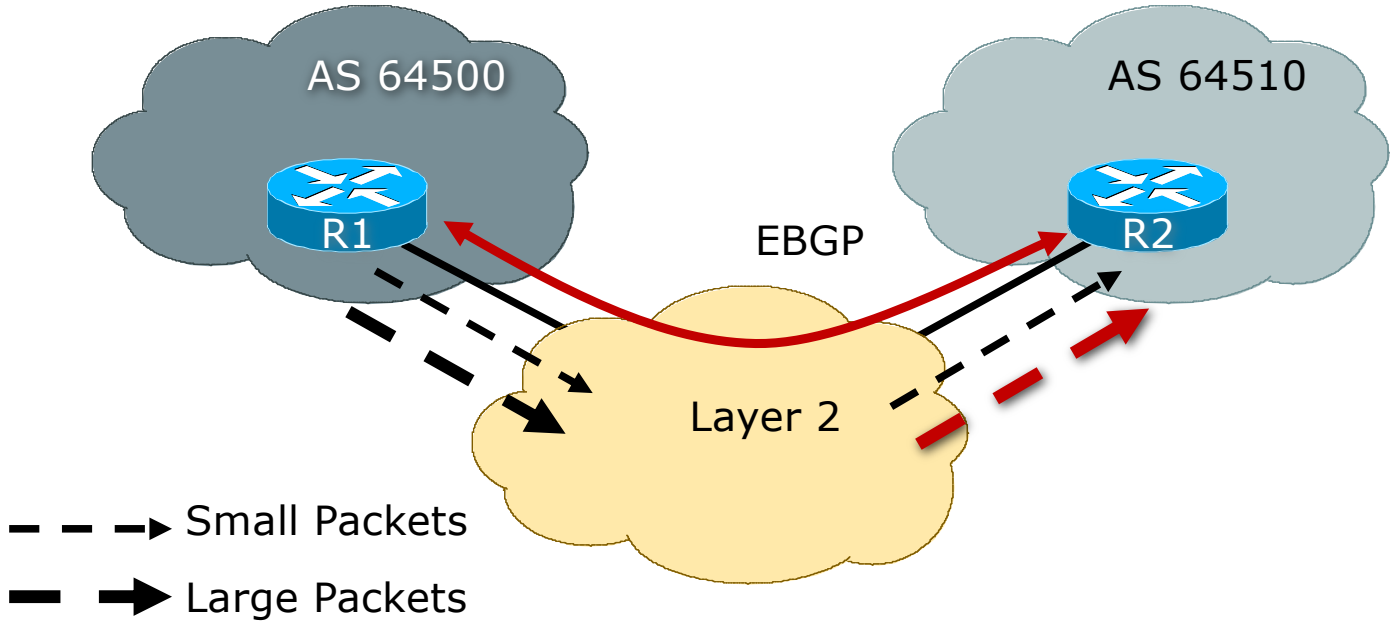
```
neighbor 100.64.0.1 tcp-mss 1400
```

  - Might be a possible temporary solution until the L2 is fixed

Flapping Peer: Other Common Problems

- Remote BGP process unstable, appears to restart every 3 to 5 minutes
  - Could be router software, or lacking sufficient memory, or router lacking CPU, or even hardware problems!
- Traffic Shaping & Rate Limiting parameters
- MTU incorrectly set on links, PMTU discovery disabled on router
- Instability on the point-to-point links
  - Faulty MUXes, poor fibre splicing, faulty SFPs, bad connectors, interoperability problems, satellite or radio problems, weather, etc
  - The list is endless – infrastruture team should know how to solve them
  - For you, ping is the tool to help diagnose

Flapping Peer: Fixed!



- 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: 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 100.67.0.0
 network 100.67.0.0 mask 255.255.252.0
```

- BGP is not originating the route???

```
R1# show ip bgp | include 100.67.0.0
R1#
```

- Do we have the exact route?

```
R1# show ip route 100.67.0.0 255.255.252.0
     % Network not in table
```

Route Origination: Example I

- Nail down routes you want to originate

```
ip route 100.67.0.0 255.255.252.0 Null0 254
```

- Check the RIB

```
R1# show ip route 100.67.0.0 255.255.252.0
       100.67.0.0/22 is subnetted, 1 subnets
S                100.67.0.0 [1/0] via Null 0
```

- BGP originates the route!!

```
R1# show ip bgp | include 100.67.0.0
*> 100.67.0.0/22           0.0.0.0                  0         32768
```

Route Origination: Example II

- Trying to originate an aggregate route

```
aggregate-address 100.70.0.0 255.255.0.0 summary-only
```

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

```
R1# show ip route 100.70.0.0 255.255.0.0 longer
      100.0.0.0/32 is subnetted, 1 subnets
C        100.70.7.7 [1/0] is directly connected, Loopback 0
```

```
R1# show ip bgp | i 100.70.0.0
R1#
```

Route Origination: Example II

- Remember, to have a BGP aggregate you need a BGP component route, not a RIB entry

```
R1# show ip bgp 100.70.0.0 255.255.0.0 longer-prefixes
R1#
```

- Once BGP has a component route we originate the aggregate

```
network 100.70.7.7 mask 255.255.255.255


R1# show ip bgp 100.70.0.0 255.255.0.0 longer-prefixes
*> 100.70.0.0/16 0.0.0.0                32768 i
s> 100.70.7.7/32 0.0.0.0  0             32768 i
```

- **s** means this entry is suppressed due to the **summary-only** argument
- Advice: **summary-only** is risky – better ways of aggregating exist

Troubleshooting Tips

- BGP Network statement rules
  - Always need an exact route (RIB)
- **`aggregate-address`** looks in the BGP table, not the RIB
- Showing RIB component routes:

```
show ip route x.x.x.x y.y.y.y longer-prefixes
```

- Showing BGP component routes:

```
show ip bgp x.x.x.x y.y.y.y longer-prefixes
```

Missing Routes

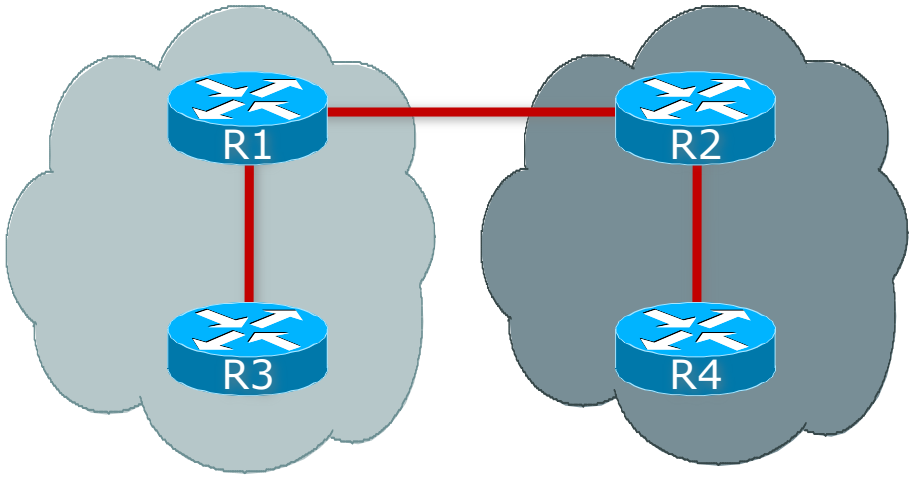
- Route Origination
- UPDATE Exchange
- Filtering
- IBGP mesh problems

Missing Routes: Update Exchange

- Ah, Route Reflectors…
  - Such a nice solution to help scale IBGP
  - But why do people insist on 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 100.64.0.0/16
  - R2 has the route
  - R1 and R3 do not



Missing Routes: Example I

- First, did R2 advertise the route to R1?

```
R2# show ip bgp neighbors 100.64.0.1 advertised-routes
BGP table version is 2, local router ID is 100.64.0.2
   Network          Next Hop      Metric LocPrf Weight Path
*>i100.64.0.0/16  100.64.0.4           0    100      0 i
```

- Did R1 receive it?

```
R1# show ip bgp neighbors 100.64.0.2 routes
Total number of prefixes 0
```

Missing Routes: Example I

- Time to debug!!

```
access-list 100 permit ip host 100.64.0.0 host 255.255.0.0
R1# debug ip bgp update 100
```

- Tell R2 to resend its UPDATEs

```
R2# clear ip bgp 100.64.0.1 out
```

- R1 debug messages show us the problem:

```
*Mar  1 21:50:12.410: BGP(0): 100.64.0.2 rcv UPDATE w/ attr: nexthop
100.64.0.4, origin i, localpref 100, metric 0, originator 100.64.1.1,
clusterlist 100.64.0.2, path , community , extended community
*Mar  1 21:50:12.410: BGP(0): 100.64.0.2 rcv UPDATE about 100.64.0.0/16 --
DENIED due to: ORIGINATOR is us;
```

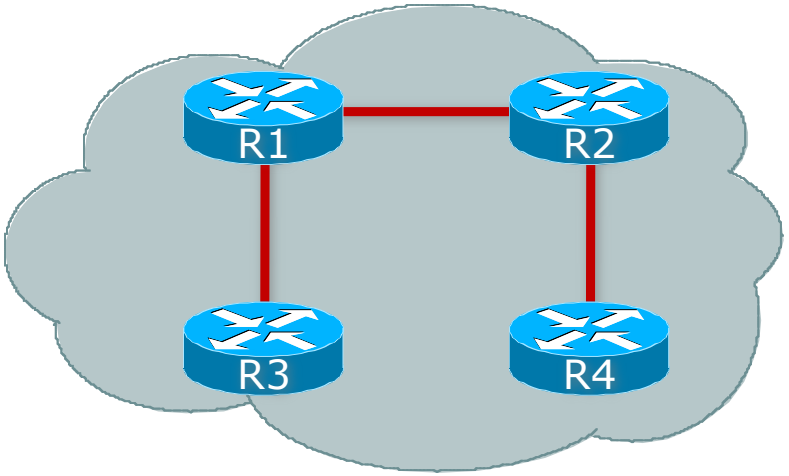
- Cannot accept an update with our Router-ID as the ORIGINATOR_ID. Another means of loop detection in BGP

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
    - Uses an existing IPv4 address (according to the implementation rules)

Missing Routes: Example II

- One RR cluster
- R1 and R2 are RRs
- R3 and R4 are RR Clients
- R4 is advertising 100.64.0.0/16
  - R2 has the route
  - 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 troubleshooting steps as for the previous example!
- Did R2 advertise it to R1?

```
R2# show ip bgp neighbors 100.64.0.1 advertised-routes
BGP table version is 2, local router ID is 100.64.0.2
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop    Metric  LocPrf  Weight Path
*>i100.64.0.0    100.64.0.4           0     100       0 i
```

- Did R1 receive it?

```
R1# show ip bgp neighbor 100.64.0.2 routes
Total number of prefixes 0
```

Missing Routes: Example II

- Time to debug!!

```
access-list 100 permit ip host 100.64.0.0 host 255.255.0.0
R1# debug ip bgp update 100
```

- Tell R2 to resend its UPDATEs

```
R2# clear ip bgp 100.64.0.1 out
```

- R1 debug messages show us the problem:

```
Mar  3 14:28:57.208: BGP(0): 100.64.0.2 rcv UPDATE w/ attr: nexthop
100.64.0.4, origin i, localpref 100, metric 0, originator 100.64.0.4,
clusterlist 0.0.0.10, path , community , extended community
Mar  3 14:28:57.208: BGP(0): 100.64.0.2 rcv UPDATE about 100.64.0.0/16 --
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 Route Reflectors, otherwise examples like this will occur
- Solution
  - Don't ever set the cluster-ID; instead,
  - Use overlapping Route Reflector Clusters for redundancy

Troubleshooting Tips

- The list of NLRI you sent a peer:

```
show ip bgp neighbor x.x.x.x advertised-routes
```

  - Note: The attribute values shown are taken from the BGP table; attribute modifications by outbound route-maps will not be shown
- Display the routes sent to us by neighbour x.x.x.x after processing by our inbound filters:

```
show ip bgp neighbor x.x.x.x routes
```

- Display the routes sent to us by neighbour x.x.x.x prior to processing by our inbound filters

```
show ip bgp neighbor x.x.x.x received-routes
```

  - Can only use if Soft-Reconfiguration is enabled, it is not available with the BGP standard (and default) Route Refresh

Troubleshooting Tips "soft-reconfiguration"

- Ideal for troubleshooting problems with inbound filters and attributes

```
alpha#sh ip bgp neigh 100.64.12.1 routes
   Network          Next Hop      Metric LocPrf Weight Path
*>i100.65.0.0       100.64.12.1        0     50       0 i
*>i100.70.0.0/19    100.64.5.1               200       0 64503 64504 i
```

```
alpha#sh ip bgp neigh 100.64.12.1 received-routes
   Network          Next Hop      Metric LocPrf Weight Path
* i100.65.0.0       100.64.12.1        0    100       0 i
* i100.70.0.0/19    100.64.5.1               100       0 64503 64504 i
* i169.254.0.0      100.64.5.1         0     100       0 64503 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
```

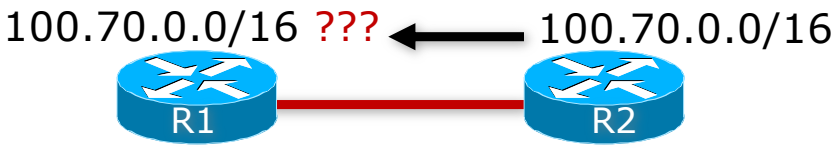
```
show run | include neighbor x.x.x.x
```

- 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



- Missing 100.70.0.0/16 in R1 (100.64.0.1)
- Not received from R2 (100.64.0.2)

```
R1#show ip bgp neigh 100.64.0.2 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 100.64.0.1 advertised-routes
Network   Next Hop    Metric LocPrf Weight Path

R2#show ip bgp 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 1660
Paths: (1 available, best #1)
 Not advertised to any peer
 Local
  0.0.0.0 from 0.0.0.0 (100.64.0.2)
  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 at first glance

```
R2#show run | include neighbor 100.64.0.1
 neighbor 100.64.0.1 remote-as 64503
 neighbor 100.64.0.1 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 100.64.0.2
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
*> 100.70.0.0/16     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

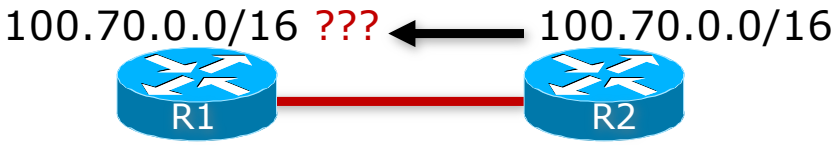
```
R2#clear ip bgp 100.64.0.1 out

R1#show ip bgp 100.70.0.0
% 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 100.64.0.2
 neighbor 100.64.0.2 remote-as 12
 neighbor 100.64.0.2 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 100.70.0.0 host 255.255.255.0
R1#show access-list 101
Extended IP access list 101
    permit ip 100.71.1.0 0.0.0.255 host 255.255.255.0
R1#show ip as-path 1
AS path access list 1
    permit ^64511$
```

Missing Routes
Update Filters



- Confused? Let's run some debugs

```
R1#show access-list 99
Standard IP access list 99
    permit 100.70.0.0

R1#debug ip bgp 100.64.0.2 update 99
BGP updates debugging is on for access list 99 for neighbor 100.64.0.2

R1#
4d00h: BGP(0): 100.64.0.2 rcvd UPDATE w/ attr: nexthop 100.64.0.2,
 origin i, metric 0, path 12
4d00h: BGP(0): 100.64.0.2 rcvd 100.70.0.0/16 -- DENIED due to: route-map;
```

Missing Routes
Update Filters

```
R1#sh run | include neighbor 100.64.0.2
 neighbor 100.64.0.2 remote-as 12
 neighbor 100.64.0.2 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 100.70.0.0 host 255.255.255.0
R1#sh access-list 101
Extended IP access list 101
    permit ip 100.71.1.0 0.0.0.255 host 255.255.255.0
R1#sh ip as-path 1
AS path access list 1
    permit ^64511$
```

Missing Routes
Update Filters

- Wrong mask! Needs to be /16 and the ACL allows a /24 only!

```
access list 100
permit ip host 100.70.0.0 host 255.255.255.0
```

- Should be

```
access list 100
permit ip host 100.70.0.0 host 255.255.0.0
```

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

```
ip prefix-list my_filter permit 100.70.0.0/16
```

- 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

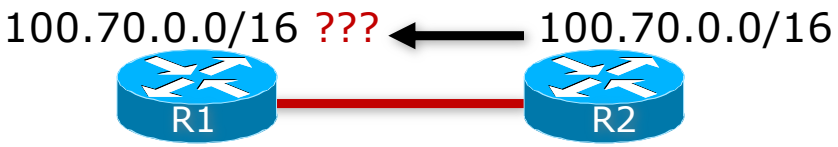
Update Filtering: Summary

- 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
  - Every character matters in a regular expression, 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



- Missing 100.70.0.0/16 in R1 (100.64.0.1)
- Not received from R2 (100.64.0.2)

```
R1#show ip bgp neigh 100.64.0.2 routes

Total number of prefixes 0
```

Missing Routes
Community Problems

- R2 originates the route

```
R2#show ip bgp 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 1660
Paths: (1 available, best #1)
 Not advertised to any peer
 Local
  0.0.0.0 from 0.0.0.0 (100.64.0.2)
  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 64502
 network 100.70.0.0 route-map set-community
...
route-map set-community permit 10
 set community 64502:2 64501:50
```

```
R2#show ip bgp 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 1660
Paths: (1 available, best #1)
 Not advertised to any peer
 Local
  0.0.0.0 from 0.0.0.0 (100.64.0.2)
  Origin IGP, metric 0, localpref 100, weight 32768, valid,
   sourced, local, best
  Community 64502:2 64501:50
```

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 neighbor 100.64.0.2 routes

Total number of prefixes 0
```

- Configuration verified on R2
- No filters blocking announcement on R2
- What's wrong?

Missing Routes
Community Problems

- Check R2 configuration again!

```
R2#show run | begin bgp
router bgp 64502
 network 100.70.0.0 mask 255.255.0.0 route-map set-community
 neighbor 100.64.0.1 remote-as 64501
 neighbor 100.64.0.1 prefix-list my-agg out
 neighbor 100.64.0.1 prefix-list their-agg in
!
ip prefix-list my-agg permit 100.70.0.0/16
ip prefix-list their-agg permit 100.80.0.0/16
!
route-map set-community permit 10
 set community 64502:2 64501:50
```

- Looks okay – filters okay, route-map okay
- But forgotten the **`neighbor 100.64.0.1 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 64501
 neighbor 100.64.0.2 remote-as 64502
 neighbor 100.64.0.2 route-map R2-in in
 neighbor 100.64.0.2 route-map R1-out out
!
ip community-list 1 permit 64501:150
!
route-map R2-in permit 10
 match community 1
 set local-preference 150
```

Missing Routes
Community Problems

- Community match on R1 expects 64501:150 to be set on prefix
- But R2 is sending 64501:50
  - Typo or miscommunication between operations?
- R2 is also using the route-map to filter
  - If the prefix does not have community 64501:150 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…
  - route-map line with no <match> condition means match everything, set nothing

Missing Routes
Community Problems

- Fix configuration on R2 to set community 64501:150 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 100.64.0.2 routes

   Network          Next Hop      Metric LocPrf Weight Path
*  100.70.0.0/16    100.64.0.2           0             0 64502 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 network operator 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

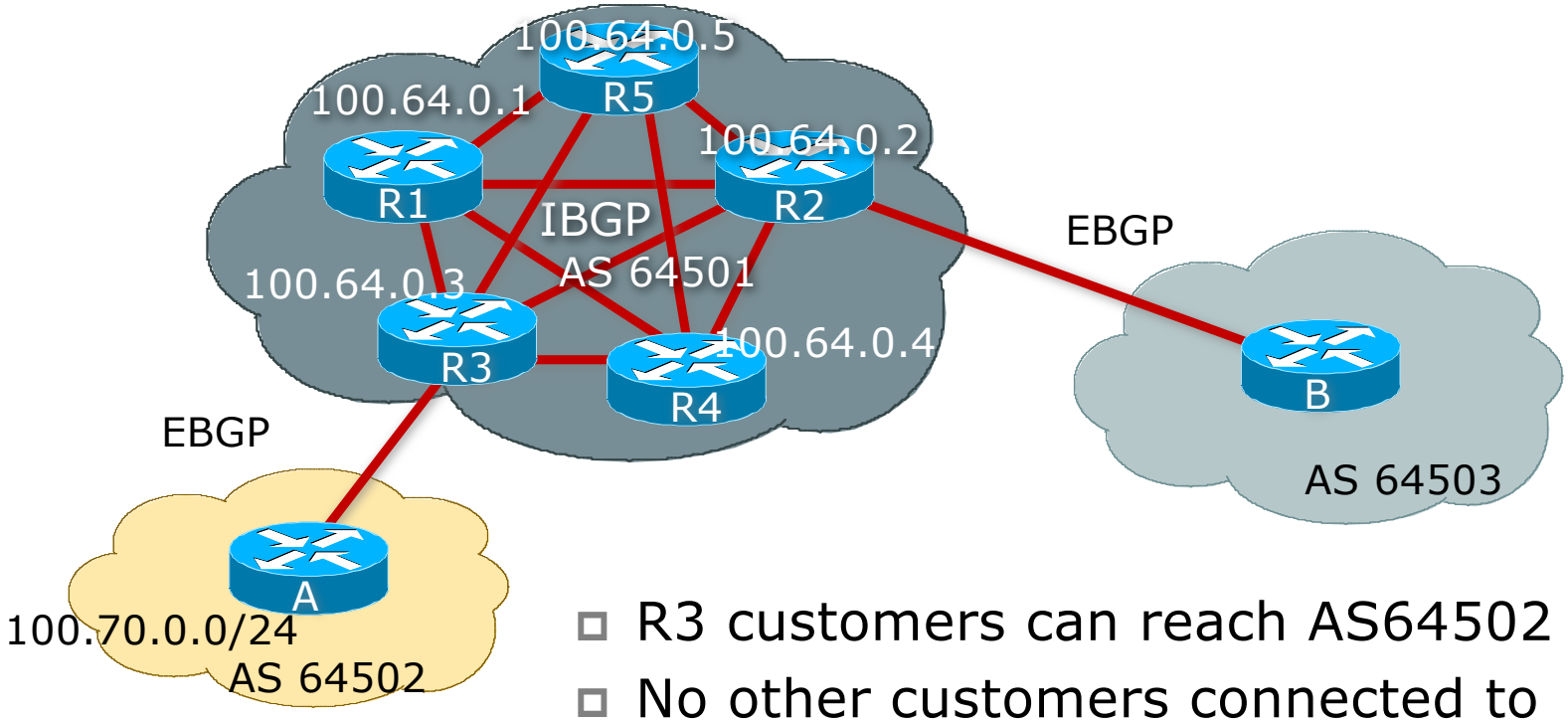
Missing Routes

- 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 AS64502
- No other customers connected to AS64501 or AS64503 can reach AS64502

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          100.70.0.10                        0 64502 64505 i
*> 4.0.0.0          100.70.0.10                        0 64502 64505 i
*> 100.70.0.0/24    100.70.0.10                        0 64502 i
*> 100.72.0.0/16    100.70.0.10                        0 64502 i
```

- Looking at R4

```
R4#show ip bgp
   Network          Next Hop      Metric LocPrf Weight Path
* i3.0.0.0          100.70.0.10              100      0 64502 64505 i
* i4.0.0.0          100.70.0.10              100      0 64502 64505 i
* i100.70.0.0/24    100.70.0.10              100      0 64502 i
* i100.72.0.0/16    100.70.0.10              100      0 64502 i
```

Missing Routes: IBGP Example I

- Notice that R3 reports the prefixes learned from AS64502
  - 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 100.70.0.0/24
BGP routing table entry for 100.70.0.0/24, version 136
Paths: (1 available, no best path)
  Not advertised to any peer
  2, (received & used)
    100.70.0.10 (inaccessible) from 100.64.1.2 (100.77.0.1)
      Origin IGP, metric 0, localpref 100, valid, internal
```

- Look at the Routing Table entry

```
R4#sh ip route 100.70.0.0 255.255.255.0
% Network not in table
```

- The next hop?

```
R4#sh ip route 100.70.0.10
% Network not in table
```

Missing Routes: IBGP Example I – Diagnosis

- R4 does not use the 100.70.0.0/24 destination because there is no valid next-hop
- Configuration on R3 has:
  - Either no routing information on how to reach the 100.70.0.10/30 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

Missing Routes: IBGP Example I – 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 calls it **`next-hop-self`**)

Missing Routes
IBGP Example I – Solution

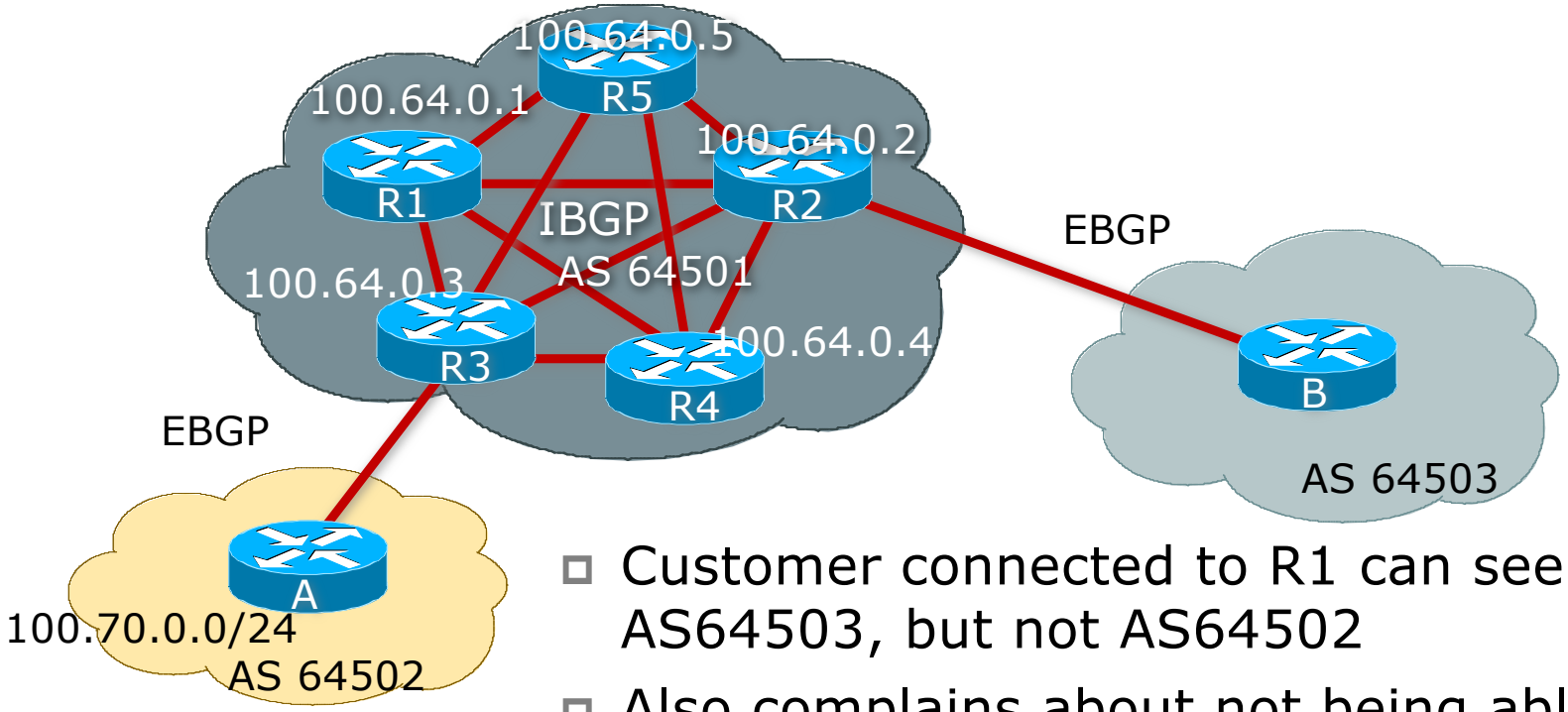
- R3 now includes the missing **`next-hop-self`** configuration
- Looking at R4 now:

```
R4#show ip bgp
   Network          Next Hop       Metric LocPrf Weight Path
*>i3.0.0.0         100.64.0.3                100      0 64502 64505 i
*>i4.0.0.0         100.64.0.3                100      0 64502 64505 i
*>i100.70.0.0/24   100.64.0.3                100      0 64502 i
*>i100.72.0.0/16   100.64.0.3                100      0 64502 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



- Customer connected to R1 can see AS64503, but not AS64502
- Also complains about not being able to see sites connected to R5
- No complaints from other customers

Missing Routes
IBGP Example II

- 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 AS64502 ⇒ R3 is somehow not passing routing information about AS64502 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 ⇒ their 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
100.64.0.1      4        64501     200      20       32    0    0 3d10h     Active
100.64.0.2      4        64501     210      25       32    0    0 3d16h           15
100.64.0.4      4        64501     213      22       32    0    0 3d16h           12
100.64.0.5      4        64501     215      19       32    0    0 3d16h            0
100.64.1.2      4        64502    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
  - Something has broken between R1 and R3

Missing Routes
IBGP Example II

- Now check configuration on R1

```
R1#sh conf | b bgp
router bgp 64501
 neighbor IBGP-ipv4-peers peer-group
 neighbor IBGP-ipv4-peers remote-as 64501
 neighbor IBGP-ipv4-peers update-source Loopback0
 neighbor IBGP-ipv4-peers send-community
 neighbor IBGP-ipv4-peers prefix-list IBGP-prefixes out
 neighbor 100.64.0.2 peer-group IBGP-ipv4-peers
 neighbor 100.64.0.4 peer-group IBGP-ipv4-peers
 neighbor 100.64.0.5 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
100.64.0.1      4        64501     200      20       32    0    0 00:00:50        8
100.64.0.2      4        64501     210      25       32    0    0 3d16h          15
100.64.0.4      4        64501     213      22       32    0    0 3d16h          12
100.64.0.5      4        64501     215      19       32    0    0 3d16h           0
100.64.1.2      4        64502    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 100.64.0.3
 neighbor 100.64.0.3 remote-as 64501
 neighbor 100.64.0.3 update-source loopback0
 neighbor 100.64.0.3 prefix-list EBGP-filters out
 neighbor 100.64.0.4 remote-as 64501
 neighbor 100.64.0.4 update-source loopback0
 neighbor 100.64.0.4 prefix-list IBGP-filters out
!
ip prefix-list EBGP-filters permit 100.72.0.0/16
ip prefix-list IBGP-filters permit 100.70.0.0/16
```

- 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
100.64.0.1      4        64501     200      20       32    0    0 00:01:53        8
100.64.0.2      4        64501     210      25       32    0    0 3d16h          15
100.64.0.4      4        64501     213      22       32    0    0 3d16h          12
100.64.0.5      4        64501     215      19       32    0    0 3d16h           6
100.64.1.2      4        64502    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
- RFC1771 defined the decision algorithm
- Every vendor has tweaked the algorithm
- Route selection problems can result from oversights by RFC1771
- RFC1771 was made obsolete by RFC4271 in January 2006
  - How is compliance with RFC4271 today?

Inconsistent Route Selection: Example I

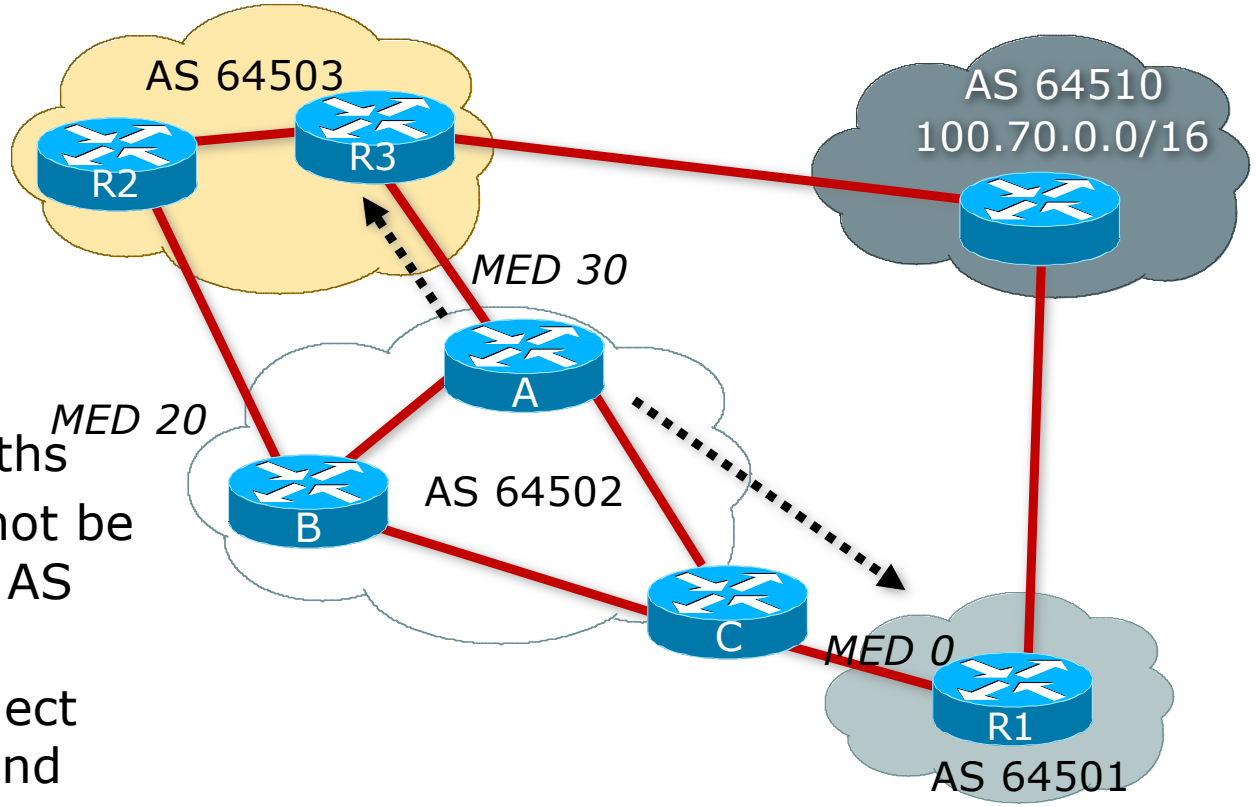
- RFC1771 said that MED is not always compared
- As a result, the ordering of the paths can affect the decision process
- For example, the default in Cisco IOS is to compare the prefixes in order of arrival (most recent to oldest)
  - This can result in inconsistent route selection
  - Symptom is that 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
    - Recommend enabling on all the routers in the AS
    - Default behaviour on most other BGP implementations today
  - The bestpath is recalculated as soon as the command is entered

Symptom I: Diagram

- RouterA will have three paths
- MEDs from AS 64503 will not be compared with MEDs from AS 64501
- 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 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 40
Paths: (3 available, best #3, advertised over IBGP, EBGP)
  64503 64510
    100.64.0.2 from 100.64.0.2
      Origin IGP, metric 20, localpref 100, valid, internal
  64503 64510
    100.64.0.3 from 100.64.0.3
      Origin IGP, metric 30, valid, external
  64501 64510
    100.64.0.1 from 100.64.0.1
      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 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 40
Paths: (3 available, best #3, advertised over IBGP, EBGP)
  64501 64510
    100.64.0.1 from 100.64.0.1
      Origin IGP, metric 0, localpref 100, valid, internal
  64503 64510
    100.64.0.2 from 100.64.0.2
      Origin IGP, metric 20, localpref 100, valid, internal
  64503 64510
    100.64.0.3 from 100.64.0.3
      Origin IGP, metric 30, valid, external, best
```

- 100.64.0.1 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

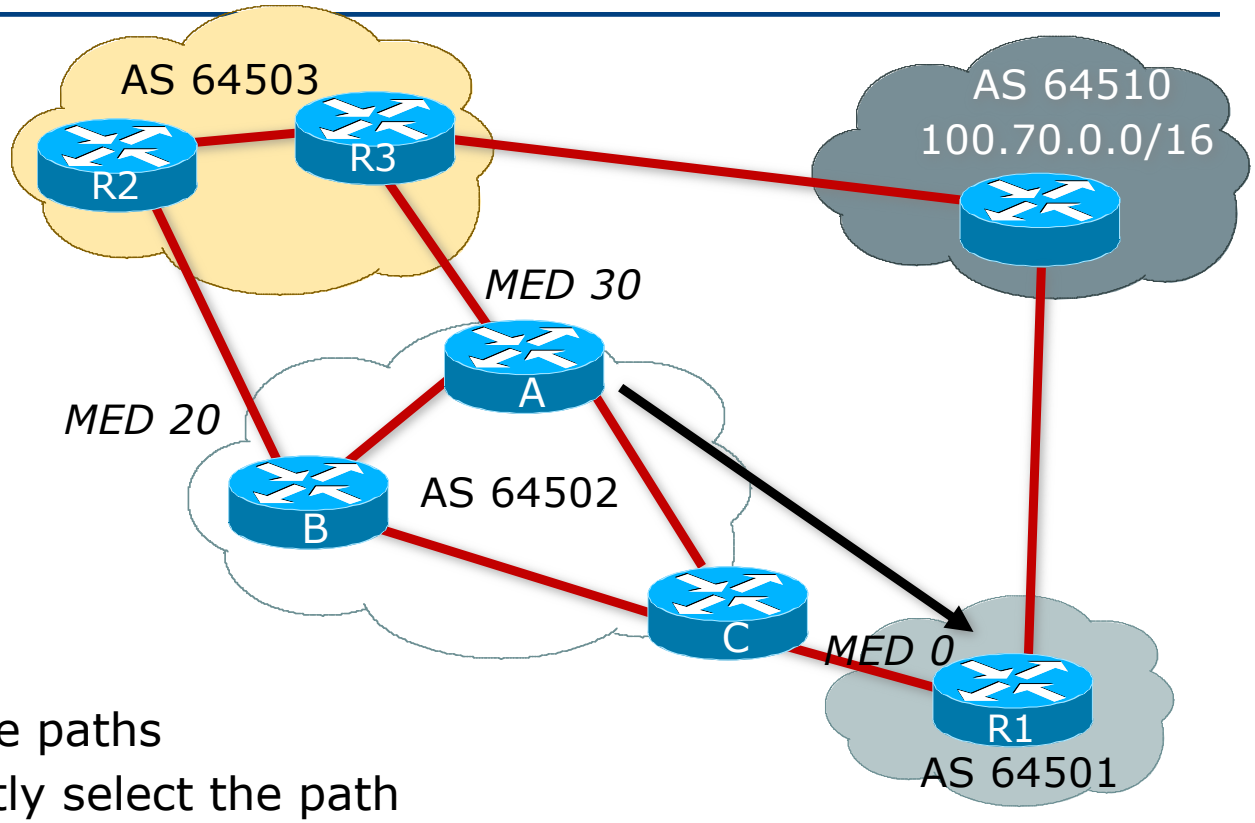
```
RouterA#sh ip bgp 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 40
Paths: (3 available, best #1, advertised over IBGP, EBGP)
  64501 64510
    100.64.0.1 from 100.64.0.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
  64503 64510
    100.64.0.2 from 100.64.0.2
      Origin IGP, metric 20, localpref 100, valid, internal
  64503 64510
    100.64.0.3 from 100.64.0.3
      Origin IGP, metric 30, valid, external
```

- Path 1 is best for AS 64501
- Path 2 beats Path 3 for AS 64503 – 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
- Recommended to be a default feature of router configuration template

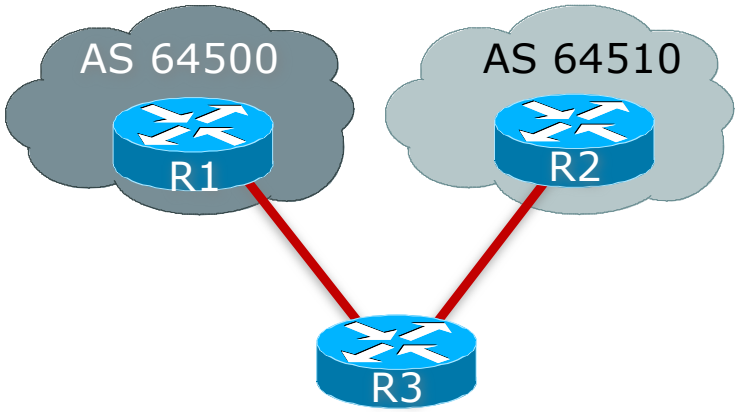
Inconsistent Route Selection: Solution – Diagram



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

Inconsistent Route Selection: Example II

- The bestpath changes every time the peering is reset




```
R3#show ip bgp 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 15
  64500 65540
    100.64.0.1 from 100.64.0.1
      Origin IGP, metric 0, localpref 100, valid, external
  64510 65540
    100.64.0.2 from 100.64.0.2
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Inconsistent Route Selection: Example II

```
R3#show ip bgp 100.70.0.0/16
BGP routing table entry for 100.70.0.0/16, version 17
Paths: (2 available, best #2)
  Not advertised to any peer
  64500 65540
    100.64.0.2 from 100.64.0.2
      Origin IGP, metric 0, localpref 100, valid, external
  64500 65540
    100.64.0.1 from 100.64.0.1
      Origin IGP, metric 0, localpref 100, valid, external, best
```

- The "oldest" external is the bestpath
  - All other attributes are the same
  - Stability enhancement – introduced in Cisco IOS 12.0(1)
- **`bgp bestpath compare-router-id`** will disable this enhancement – introduced in Cisco IOS 12.0(11)S and 12.1(3)

Inconsistent Path Selection

- Summary:
  - RFC1771 wasn't prefect when it came to path selection – early years of operational experience showed this
  - Vendors and network operators have worked to put in stability enhancements documented 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

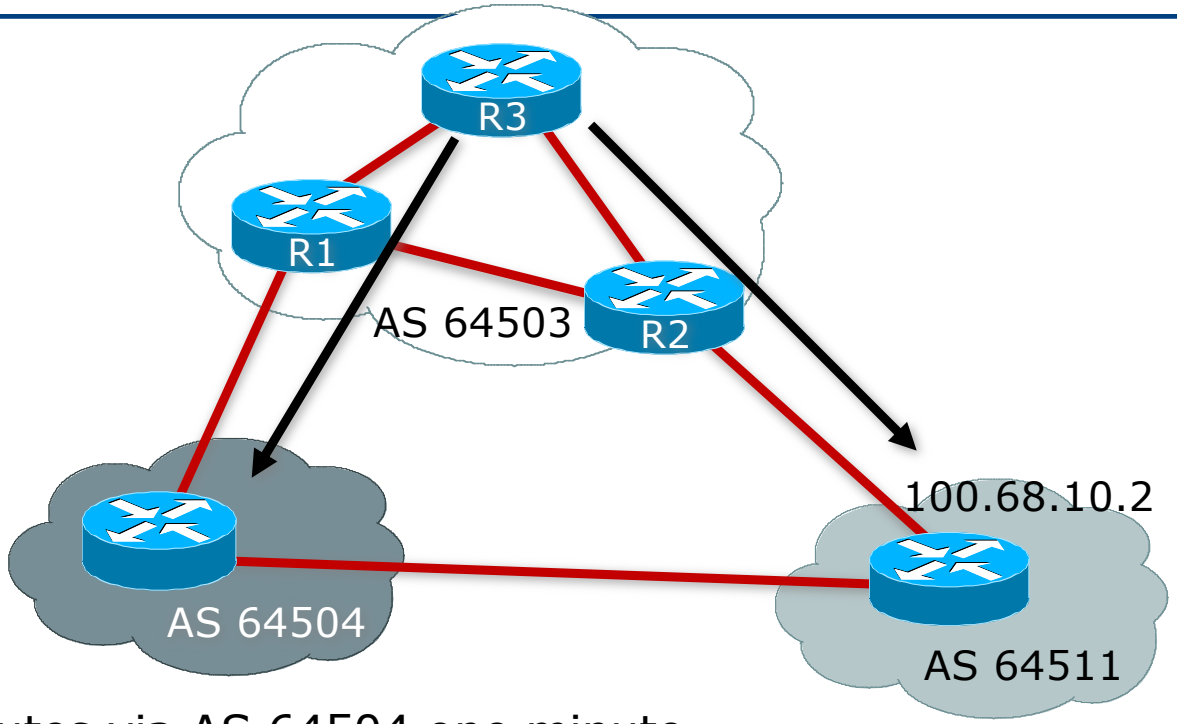
Local Configuration Problems

- 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 network operator 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 prefers routes via AS 64504 one minute
- BGP scanner runs then R3 prefers routes via AS 64511
- The entire table oscillates every 60 seconds

Route Oscillation: Diagnosis

```
R3#show ip bgp summary
BGP router identifier 100.64.0.3, local AS number 64503
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 64503       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 100.65.0.0/16
Routing entry for 100.65.0.0/16
  Known via "bgp 64503", distance 200, metric 0
 Routing Descriptor Blocks:
  * 100.64.0.1, from 100.64.0.1, 00:00:53 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2, BGP network version 474

R3#show ip bgp 100.65.0.0/16
BGP routing table entry for 100.65.0.0/16, version 474
Paths: (2 available, best #1)
  Advertised to non peer-group peers:
    100.64.0.2
  64504 64511
    100.64.0.1 from 100.64.0.1 (100.64.0.1)
      Origin IGP, localpref 100, valid, internal, best
  64512
    100.68.10.2 (inaccessible) from 100.64.0.2 (100.64.0.2)
      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 100.65.0.0/16
Routing entry for 100.65.0.0/16
  Known via "bgp 64503", distance 200, metric 0
    Routing Descriptor Blocks:
  * 100.68.10.2, from 100.64.0.2, 00:00:27 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 478

R3#sh ip bgp 100.65.0.0
BGP routing table entry for 100.65.0.0/16, version 478
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    100.64.0.1
  64504 64511
    100.64.0.1 from 100.64.0.1 (100.64.0.1)
      Origin IGP, localpref 100, valid, internal
  64511
    100.68.10.2 from 100.64.0.2 (100.64.0.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation: Troubleshooting

- Let's take a closer look at the nexthop

```
R3#show ip route 100.68.10.2
Routing entry for 100.68.0.0/16
  Known via "bgp 64503", distance 200, metric 0
 Routing Descriptor Blocks:
  * 100.68.10.2, from 100.64.0.2, 00:00:50 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 476

R3#show ip bgp 100.68.10.2
BGP routing table entry for 100.68.0.0/16, version 476
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    100.64.0.1
  64504 64511
    100.64.0.1 from 100.64.0.1 (100.64.0.1)
      Origin IGP, localpref 100, valid, internal
  64511
    100.68.10.2 from 100.64.0.2 (100.64.0.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation: Troubleshooting

- BGP nexthop is known via BGP
- This recursive lookup is not permitted in BGP
- Scanner will notice and install the other path in the RIB

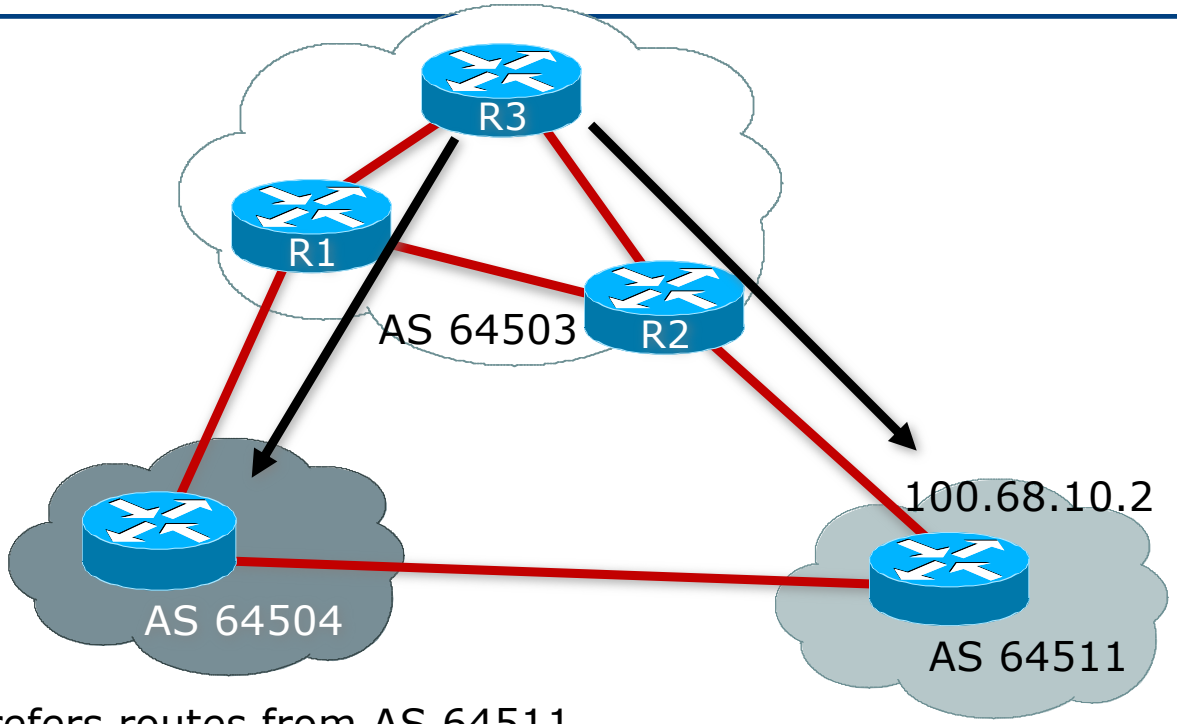
```
R3#sh debug
  BGP events debugging is on
  BGP updates debugging is on
  IP routing debugging is on
R3#
BGP: scanning routing tables
BGP: nettable_walker 100.68.0.0/16 calling revise_route
RT: del 100.68.0.0 via 100.68.10.2, bgp metric [200/0]
BGP: revise route installing 100.68.0.0/16 -> 100.64.0.1
RT: add 100.68.0.0/16 via 100.64.0.1, bgp metric [200/0]
RT: del 100.65.0.0 via 100.68.10.2, bgp metric [200/0]
BGP: revise route installing 100.65.0.0/16 -> 100.64.0.1
RT: add 100.65.0.0/16 via 100.64.0.1, 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 100.68.0.0/16 calling revise_route
RT: del 100.68.0.0 via 100.64.0.1, bgp metric [200/0]
BGP: revise route installing 100.68.0.0/16 -> 100.68.10.2
RT: add 100.68.0.0/16 via 100.68.10.2, bgp metric [200/0]
BGP: nettable_walker 100.65.0.0/16 calling revise_route
RT: del 100.65.0.0 via 100.64.0.1, bgp metric [200/0]
BGP: revise route installing 100.65.0.0/16 -> 100.68.10.2
RT: add 100.65.0.0/16 via 100.68.10.2, bgp metric [200/0]
```

Route Oscillation: Step by Step



- R3 naturally prefers routes from AS 64511
- R3 does not have an IGP route to 100.68.10.2 which is the next-hop for routes learned via AS 64511
- R3 learns 100.68.0.0/16 via AS 64504 so 100.68.10.2 becomes reachable

Route Oscillation: Step by Step

- R3 then prefers the AS 64511 route for 100.68.0.0/16 whose next-hop is 100.68.10.2
- This is an illegal recursive lookup
- BGP detects the problem when scanner runs and flags 100.68.10.2 as inaccessible
- Routes through AS 64504 are now preferred
- The cycle continues forever…

Route Oscillation: Solution

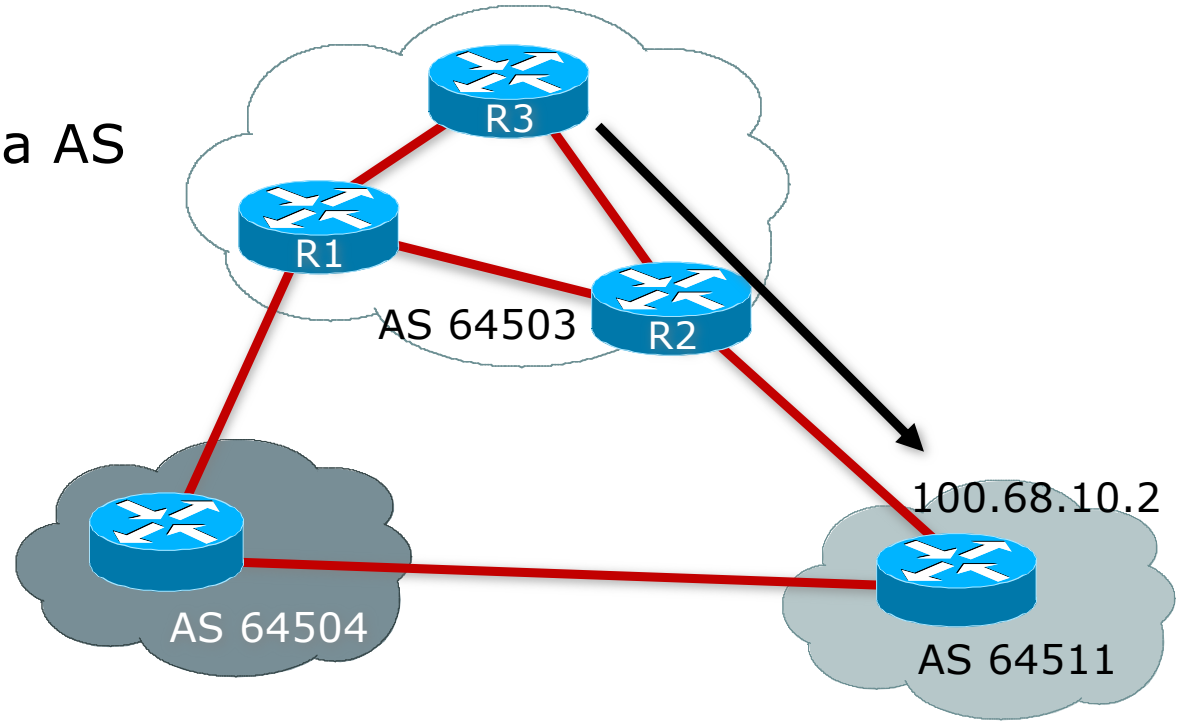
- Make sure that all the BGP NEXT_HOPs are known by the IGP
  - (whether OSPF/ISIS, static or connected routes)
  - If NEXT_HOP is also in IBGP, ensure the IBGP distance is longer than the IGP distance

—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 calls it **`next-hop-self`**)

Route Oscillation: Solution

- R3 now has IGP route to AS 64511 next-hop or R2 is using **`next-hop-self`**
- R3 now prefers routes via AS 64511 all the time
- No more oscillation!!



Troubleshooting Tips

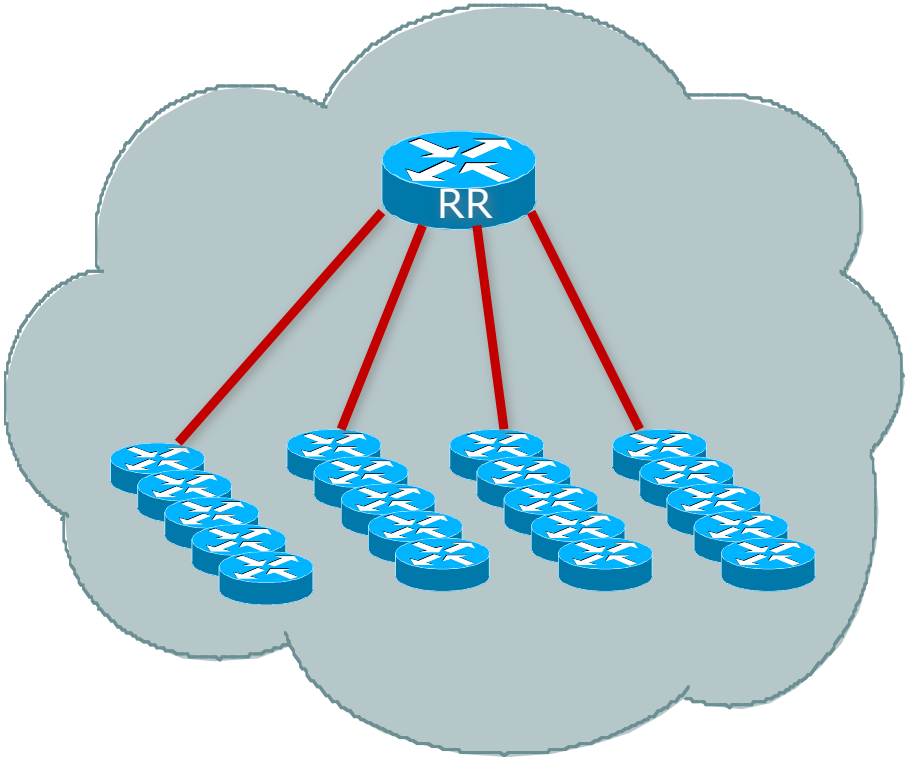
- High CPU utilisation in the BGP process is normally a sign of a convergence problem
- Find a prefix that changes every minute
- Troubleshoot/debug that one prefix

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 BGP 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
100.73.1.160    4 64500      10    5416     9419    0    0 00:00:12  Closing
100.73.1.161    4 64500      11    4418     8055    0  335 00:10:34        0
100.73.1.162    4 64500      12    4718     8759    0  128 00:10:34        0
100.73.1.163    4 64500       9    3517        0    1    0 00:00:53  Connect
100.73.1.164    4 64500      13    4789     8759    0  374 00:10:37        0
100.73.1.165    4 64500      13    3126        0    0  161 00:10:37        0
100.73.1.166    4 64500       9    5019     9645    0    0 00:00:13  Closing
100.73.1.167    4 64500       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 100.73.1.118 Down— BGP Notification sent
*May  3 15:27:16: %BGP-3-NOTIFICATION: sent to neighbor 100.73.1.118 4/0 (hold time expired) 0 byte
*May  3 15:28:10: %BGP-5-ADJCHANGE: neighbor 100.73.1.52 Down— BGP Notification sent
*May  3 15:28:10: %BGP-3-NOTIFICATION: sent to neighbor 100.73.1.52 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 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 100.70.7.156 255.255.255.0
 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 (maximum 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 maximum workable 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 MEDs are enabled
- If the entire table is having a 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
- Recommended 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 best 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 harder for humans to comprehend
  - Whereas AS:xx format is more intuitive/recognisable
  - Note: "AS" by convention is your ASN – if you have a 32-bit ASN, then simply use a private ASN instead for the first 16-bits of the community

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

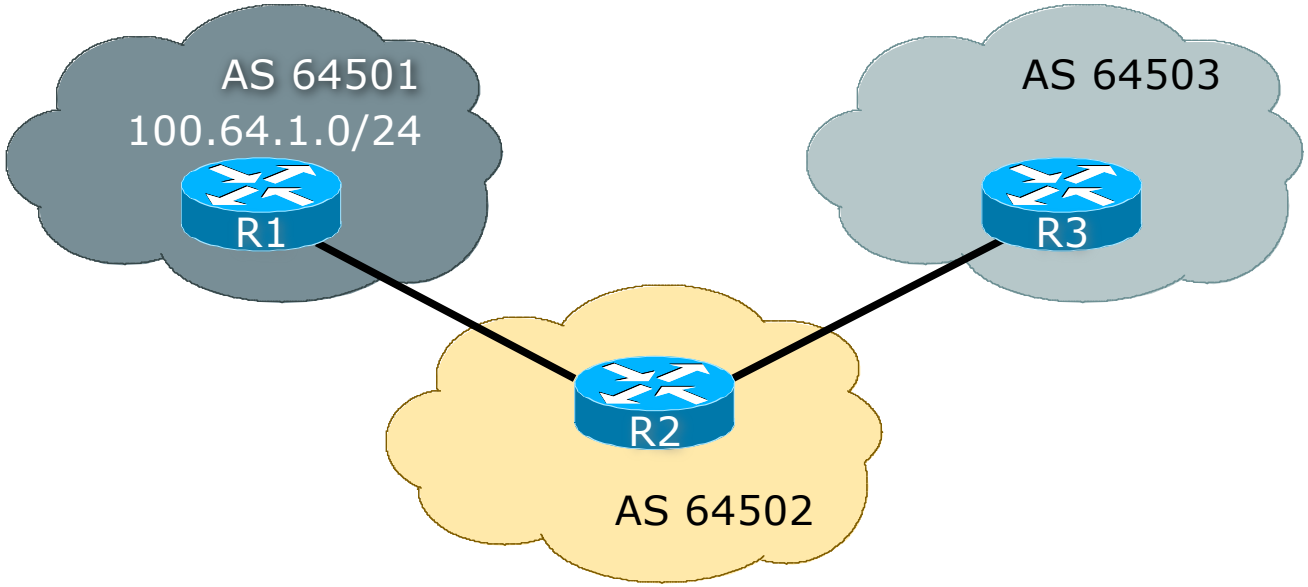
- 20 prepends will not lessen the priority of your path any more than 10 prepends will – check it out at a Looking Glass
  - The Internet is on average only 5 ASes deep; the maximum AS_PATH prepend most network operators need to use is around this too
  - Know you BGP path selection algorithm
- Some network operators limit AS path lengths
  - For example, to drop prefixes with AS paths longer than 15 ASNs:

```
bgp maxas-limit 15
```

Internet Reachability Problems
Private ASNs

- Private ASNs should not ever appear on 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
- Apparent non-removal of private ASNs may not be a bug, but a configuration error somewhere else

Troubleshooting Connectivity Example I



- Symptom: AS64501 announces 100.64.1.0/24 to AS64502 but AS64503 cannot see the network

Troubleshooting Connectivity Example I

- Checklist:
  - AS64501 announces, but does AS64502 see it?

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

Troubleshooting Connectivity Example I

- Checklist:
  - AS64501 announces, but does AS64502 see it?

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

  - Does AS64502 see it over entire network?

    We are checking IBGP across AS64502'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 AS64502 send it to AS64503?

> 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

Troubleshooting Connectivity Example I

- Checklist:
  - Does AS64502 send it to AS64503?

    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

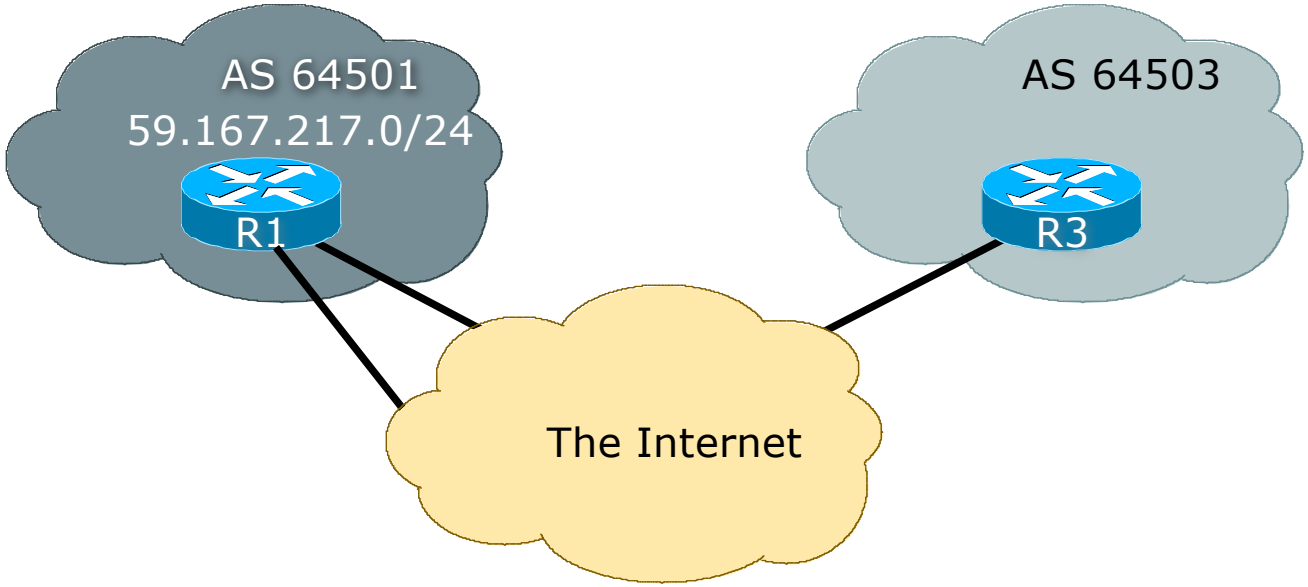
  - Does AS64503 see all of AS64502's originated prefixes?

    We are checking EBGP configuration on R3. Maybe AS64503 does not know to expect prefixes from AS64501 in the peering with AS64502, 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 ASes

Troubleshooting Connectivity Example II



- Symptom: AS64501 announces 59.167.217.0/24 to its upstreams but AS64503 cannot see the network
  - Note: example uses a real IP address for demonstration purposes, but ASN has been replaced for documentation purposes

Troubleshooting Connectivity Example II

- Checklist:
  - AS64501 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

Troubleshooting Connectivity Example II

- Checklist:
  - AS64501 announces, but do its upstreams see it?

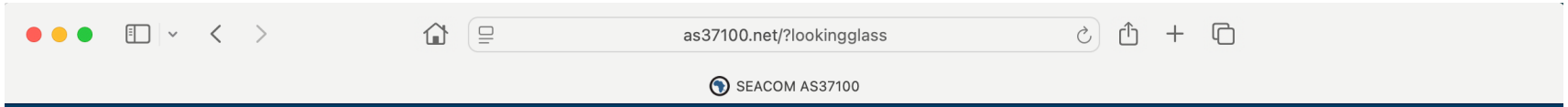
    We are checking EBGP filters on R1 and upstreams. Remember that upstreams will need to be able to help you with this

  - 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
  - Now out of date resources (sadly):
    - www.traceroute.org and www.bgp4.as/looking-glasses
- Some network operators, especially those with large and diverse networks, often provide Looking Glasses for public use to aid with network troubleshooting
- Next slides have some examples of a typical looking glass in action



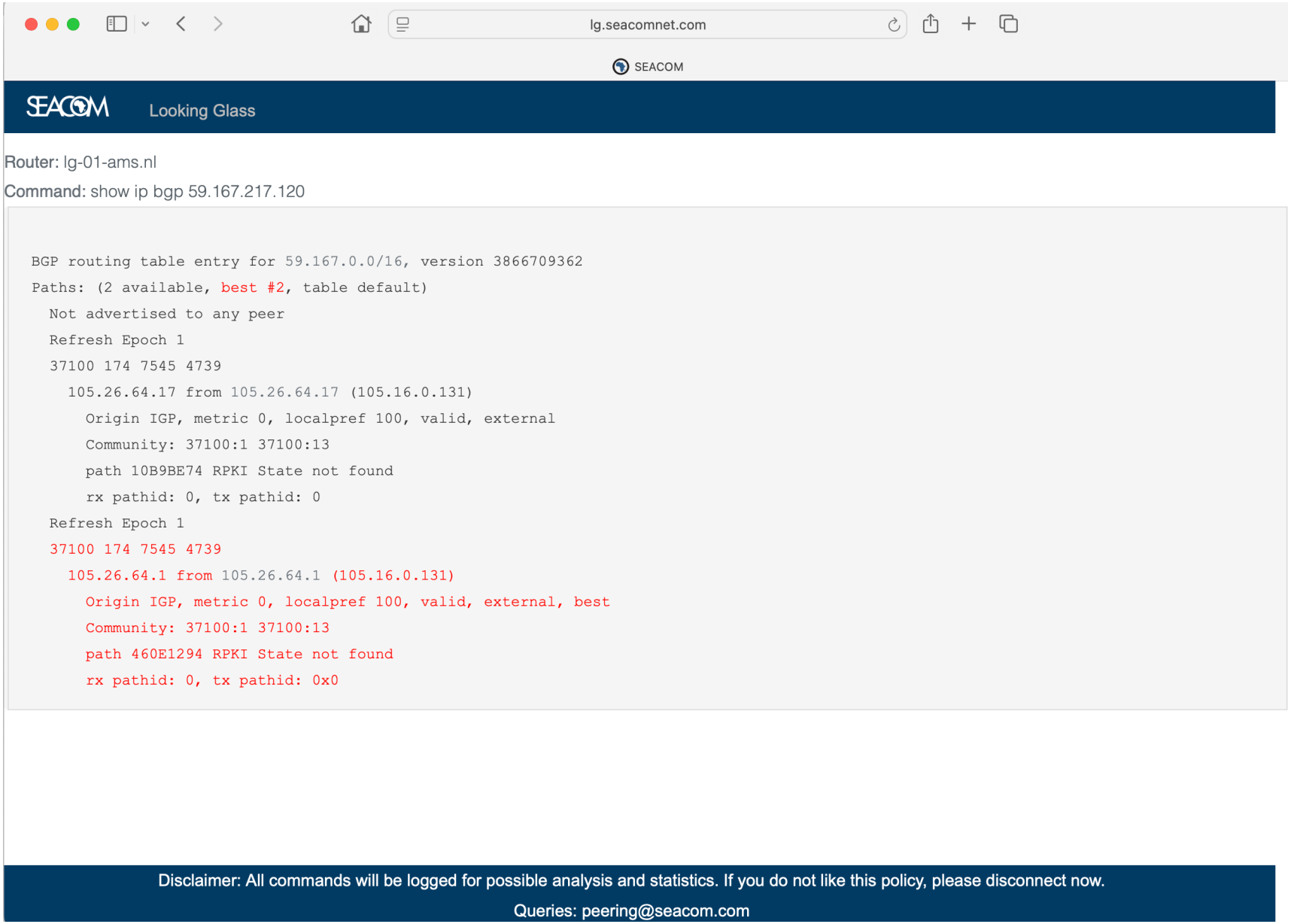
SEACOM AS37100

eBGP-BASED LOOKING GLASS

- For all normal BGP troubleshooting.
- For BGP troubleshooting from any of SEACOM's PoP's.
- SEACOM customer point-of-view.
- Support for command line interface via Telnet.

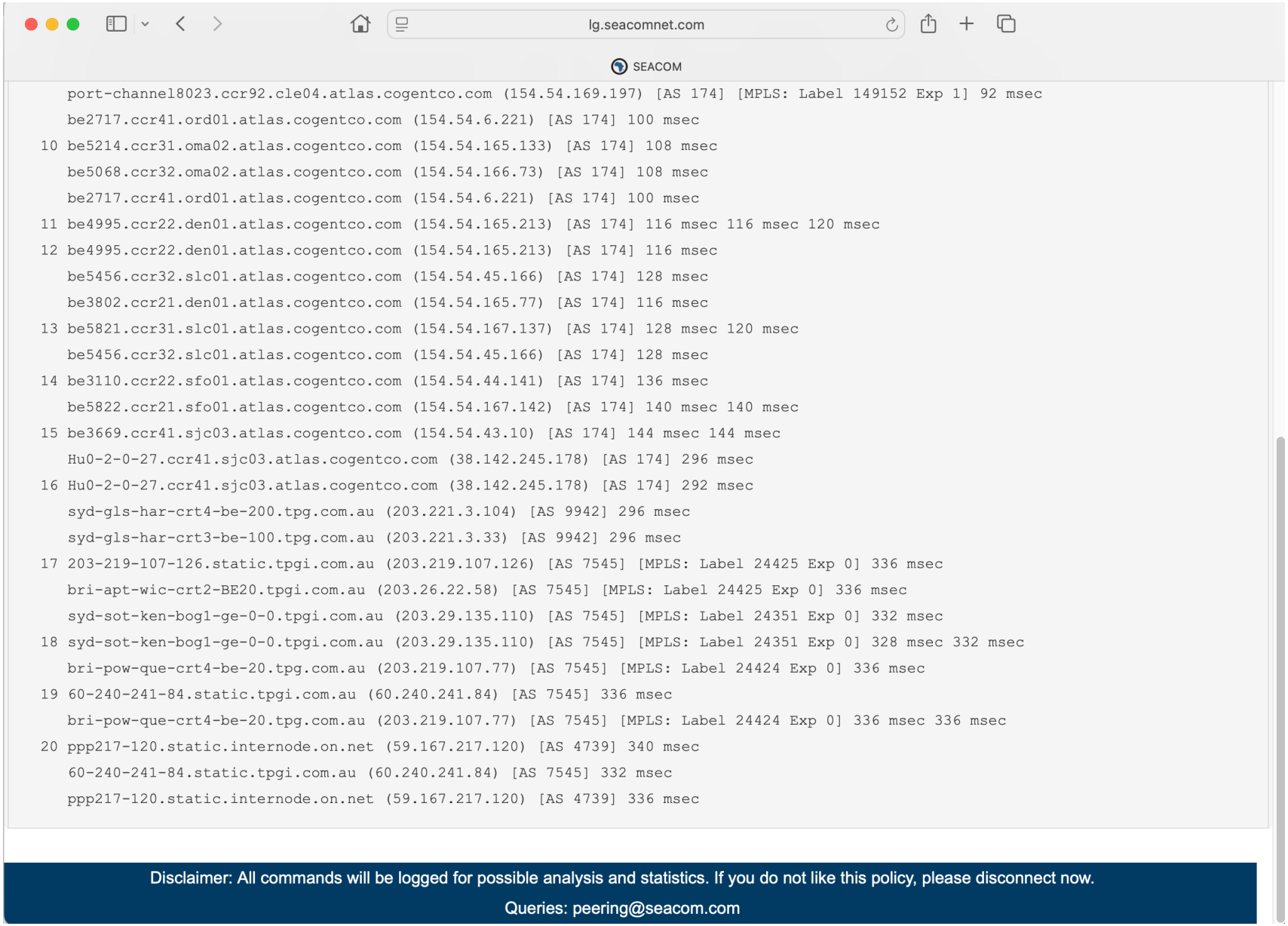
iBGP-BASED LOOKING GLASS

- For troubleshooting BGP attributes that are non-transitive, e.g., LOCAL_PREF.
- For troubleshooting blackholed routes announced to SEACOM.
- SEACOM internal backbone point-of-view.
- Support for command line interface via Telnet.


lg.seacomnet.com
SEACOM
SEACOM Looking Glass
Router: lg-01-ams.nl
Command: show ip bgp 59.167.217.120
BGP routing table entry for 59.167.0.0/16, version 3866709362
Paths: (2 available, best #2, table default)
Not advertised to any peer
Refresh Epoch 1
37100 174 7545 4739
105.26.64.17 from 105.26.64.17 (105.16.0.131)
Origin IGP, metric 0, localpref 100, valid, external
Community: 37100:1 37100:13
path 10B9BE74 RPKI State not found
rx pathid: 0, tx pathid: 0
Refresh Epoch 1
37100 174 7545 4739
105.26.64.1 from 105.26.64.1 (105.16.0.131)
Origin IGP, metric 0, localpref 100, valid, external, best
Community: 37100:1 37100:13
path 460E1294 RPKI State not found
rx pathid: 0, tx pathid: 0x0
Disclaimer: All commands will be logged for possible analysis and statistics. If you do not like this policy, please disconnect now.
Queries: peering@seacom.com

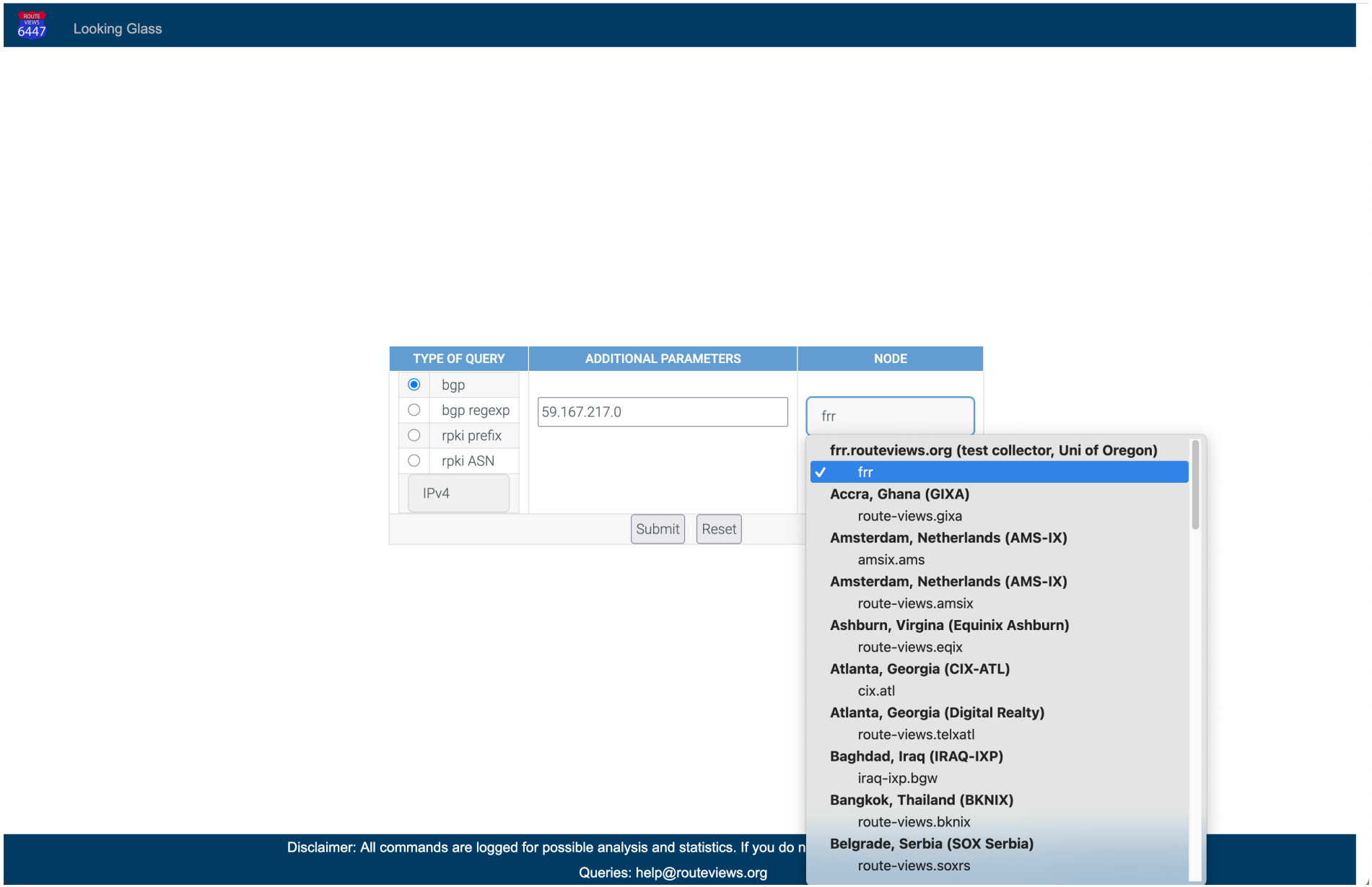
Troubleshooting Connectivity Example II

- Hmmm….
- Looking Glass can see 59.167.0.0/16
  - This includes 59.167.217.0/24
  - The problem must be with AS64503, or AS64503's upstream
- A traceroute confirms the connectivity


lg.seacomnet.com
SEACOM
port-channel8023.ccr92.cle04.atlas.cogentco.com (154.54.169.197) [AS 174] [MPLS: Label 149152 Exp 1] 92 msec
be2717.ccr41.ord01.atlas.cogentco.com (154.54.6.221) [AS 174] 100 msec
10 be5214.ccr31.oma02.atlas.cogentco.com (154.54.165.133) [AS 174] 108 msec
be5068.ccr32.oma02.atlas.cogentco.com (154.54.166.73) [AS 174] 108 msec
be2717.ccr41.ord01.atlas.cogentco.com (154.54.6.221) [AS 174] 100 msec
11 be4995.ccr22.den01.atlas.cogentco.com (154.54.165.213) [AS 174] 116 msec 116 msec 120 msec
12 be4995.ccr22.den01.atlas.cogentco.com (154.54.165.213) [AS 174] 116 msec
be5456.ccr32.slc01.atlas.cogentco.com (154.54.45.166) [AS 174] 128 msec
be3802.ccr21.den01.atlas.cogentco.com (154.54.165.77) [AS 174] 116 msec
13 be5821.ccr31.slc01.atlas.cogentco.com (154.54.167.137) [AS 174] 128 msec 120 msec
be5456.ccr32.slc01.atlas.cogentco.com (154.54.45.166) [AS 174] 128 msec
14 be3110.ccr22.sfo01.atlas.cogentco.com (154.54.44.141) [AS 174] 136 msec
be5822.ccr21.sfo01.atlas.cogentco.com (154.54.167.142) [AS 174] 140 msec 140 msec
15 be3669.ccr41.sjc03.atlas.cogentco.com (154.54.43.10) [AS 174] 144 msec 144 msec
Hu0-2-0-27.ccr41.sjc03.atlas.cogentco.com (38.142.245.178) [AS 174] 296 msec
16 Hu0-2-0-27.ccr41.sjc03.atlas.cogentco.com (38.142.245.178) [AS 174] 292 msec
syd-gls-har-crt4-be-200.tpg.com.au (203.221.3.104) [AS 9942] 296 msec
syd-gls-har-crt3-be-100.tpg.com.au (203.221.3.33) [AS 9942] 296 msec
17 203-219-107-126.static.tpgi.com.au (203.219.107.126) [AS 7545] [MPLS: Label 24425 Exp 0] 336 msec
bri-apt-wic-crt2-BE20.tpgi.com.au (203.26.22.58) [AS 7545] [MPLS: Label 24425 Exp 0] 336 msec
syd-sot-ken-bog1-ge-0-0.tpgi.com.au (203.29.135.110) [AS 7545] [MPLS: Label 24351 Exp 0] 332 msec
18 syd-sot-ken-bog1-ge-0-0.tpgi.com.au (203.29.135.110) [AS 7545] [MPLS: Label 24351 Exp 0] 328 msec 332 msec
bri-pow-que-crt4-be-20.tpg.com.au (203.219.107.77) [AS 7545] [MPLS: Label 24424 Exp 0] 336 msec
19 60-240-241-84.static.tpgi.com.au (60.240.241.84) [AS 7545] 336 msec
bri-pow-que-crt4-be-20.tpg.com.au (203.219.107.77) [AS 7545] [MPLS: Label 24424 Exp 0] 336 msec 336 msec
20 ppp217-120.static.internode.on.net (59.167.217.120) [AS 4739] 340 msec
60-240-241-84.static.tpgi.com.au (60.240.241.84) [AS 7545] 332 msec
ppp217-120.static.internode.on.net (59.167.217.120) [AS 4739] 336 msec
Disclaimer: All commands will be logged for possible analysis and statistics. If you do not like this policy, please disconnect now.
Queries: peering@seacom.com

Troubleshooting Connectivity Example II

- More help is at hand – RouteViews
  - www.routeviews.org explains the project
- RouteViews collects BGP feeds from approximately 50 collectors from around the world
  - Collector BGP feeds are accessed via a Looking Glass (https://lg.routeviews.org) and an API (https://api.routeviews.org)
  - Can check BGP reachability at 50+ locations from one interface
- RouteViews also operates a Cisco router
  - Gives access to a real router via telnet (route-views.routeviews.org)
  - Allows any provider to find out how their prefixes are seen in various parts of the Internet
  - Complements the Looking Glass facilities


ROUTE VIEWS 6447
Looking Glass
TYPE OF QUERY
ADDITIONAL PARAMETERS
NODE
bgp
bgp regexp
rpki prefix
rpki ASN
IPv4
59.167.217.0
frr
Submit
Reset
frr.routeviews.org (test collector, Uni of Oregon)
frr
Accra, Ghana (GIXA)
route-views.gixa
Amsterdam, Netherlands (AMS-IX)
amsix.ams
Amsterdam, Netherlands (AMS-IX)
route-views.amsix
Ashburn, Virgina (Equinix Ashburn)
route-views.eqix
Atlanta, Georgia (CIX-ATL)
cix.atl
Atlanta, Georgia (Digital Realty)
route-views.telxatl
Baghdad, Iraq (IRAQ-IXP)
iraq-ixp.bgw
Bangkok, Thailand (BKNIX)
route-views.bknix
Belgrade, Serbia (SOX Serbia)
route-views.soxrs
Disclaimer: All commands are logged for possible analysis and statistics. If you do n
Queries: help@routeviews.org

ROUTE VIEWS 6447 Looking Glass

Router: route-views.flix

Command: show bgp ipv4 unicast 59.167.217.0

```
route-views.flix> show bgp ipv4 unicast 59.167.217.0
BGP routing table entry for 59.167.0.0/16, version 91503700
Paths: (8 available, best #1, table default)
  Not advertised to any peer
  19151 174 7545 4739
    206.41.108.11 from 206.41.108.11 (66.216.0.33)
      Origin IGP, metric 0, valid, external, best (AS Path), rpki validation-state: not found
      Last update: Wed Mar  5 09:47:10 2025
  1031 6453 7545 4739
    206.41.108.179 from 206.41.108.179 (10.31.0.207)
      Origin IGP, metric 0, valid, external, rpki validation-state: not found
      Community: 1031:703 1031:800 6453:50 6453:1000 6453:1300 6453:1303
      Last update: Mon Apr 28 21:43:56 2025
  16552 6453 7545 4739
    206.41.108.147 from 206.41.108.147 (46.31.238.57)
      Origin IGP, valid, external, rpki validation-state: not found
      Community: 6453:50 6453:1000 6453:1300 6453:1303 16552:9100 16552:10500 16552:20840 16552:40024
      Last update: Sun May  4 15:14:06 2025
  264409 174 7545 4739
    206.41.108.178 from 206.41.108.178 (131.221.47.237)
      Origin IGP, valid, external, rpki validation-state: not found
      Community: 174:21001 174:22013 54510:1 54510:174
      Last update: Sat May 17 23:09:01 2025
  52320 1299 7545 4739
    206.41.108.60 from 206.41.108.60 (200.16.68.6)
      Origin IGP, metric 0, valid, external, rpki validation-state: not found
      Community: 52320:11912
      Last update: Thu Apr 10 02:37:21 2025
  6939 4635 7545 4739
    206.41.108.23 from 206.41.108.23 (216.218.253.12)
      Origin IGP, valid, external, rpki validation-state: not found
      Last update: Sun May  4 13:35:37 2025
  52468 6762 174 7545 4739
    206.41.108.150 from 206.41.108.150 (62.115.46.18)
```

Troubleshooting Connectivity Example II

- Checklist:
  - Does AS64503's upstream send it to AS64503?

    We are checking EBGP configuration on AS64503'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 AS64503's assistance

Troubleshooting Connectivity Example II

- Checklist:
  - Does AS64503's upstream send it to AS64503?

    We are checking EBGP configuration on AS64503'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 AS64503's assistance

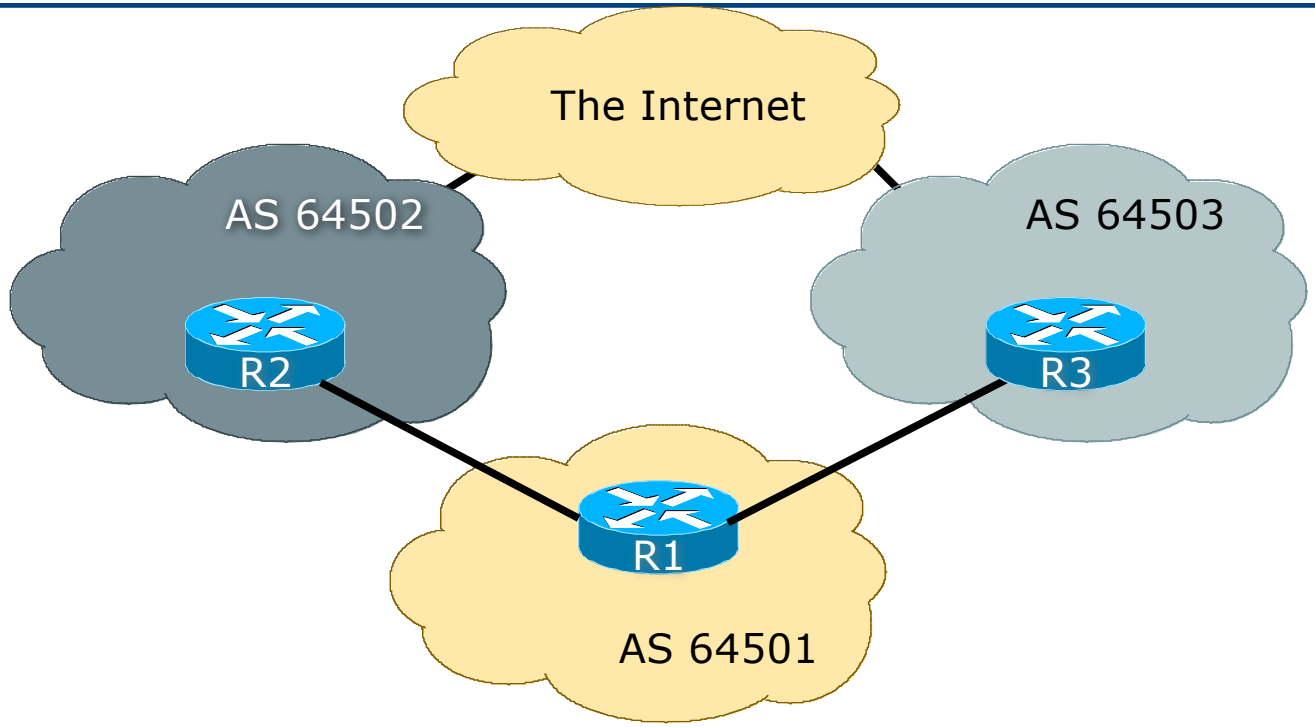
  - Does AS64503 see any of AS64501's originated prefixes?

    We are checking EBGP configuration on R3. Maybe AS64503 does not know to expect the prefix from AS64501 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



- Symptom: AS64501 is trying to loadshare between its upstreams, but has trouble getting traffic through the AS64502 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?
  - 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:
  - AS64501 announces, but does AS64502 see it?

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

Troubleshooting Connectivity Example III

- Checklist:
  - AS64501 announces, but does AS64502 see it?

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

  - Does AS64502 see it over entire network?

    We are checking IBGP across AS64502's network. Quite often IBGP is misconfigured, lack of full mesh, problems with RRs, etc.

Troubleshooting Connectivity Example III

- Checklist:
  - Does AS64502 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

Troubleshooting Connectivity Example III

- Checklist:
  - Does AS64502 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 AS64502'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 AS64502 as possible.

Troubleshooting Connectivity Example III

- Checklist:
  - Repeat all of the above for AS64503
- Stopping here and resorting to a huge prepend towards AS64503 won't solve the problem
- There are many common problems – listed on next slide
  - And tools to help decipher the problem

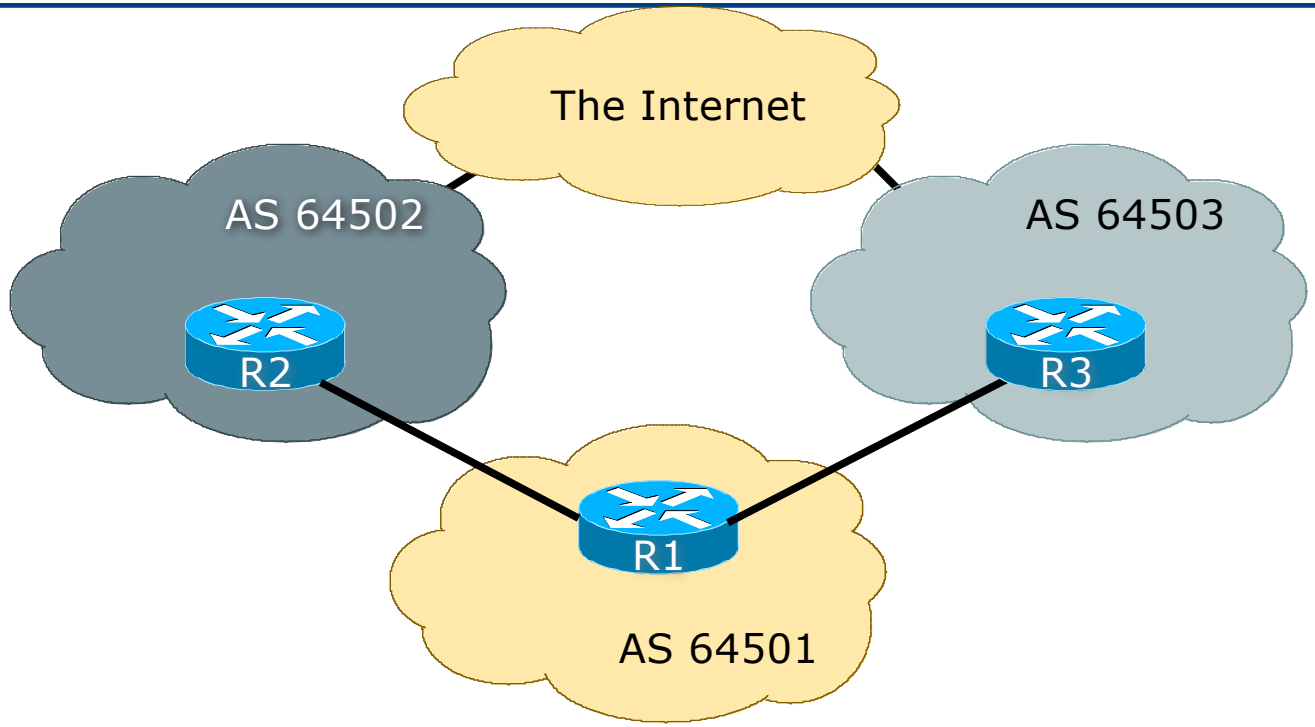
Troubleshooting Connectivity Example III

- No inbound traffic from AS64502
  - AS64502 is not seeing AS64501'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 AS64502's network blocks, then is AS64502 announcing the prefix to its upstreams?
  - If they claim they are, ask them to ask their upstream for a BGP RIB dump showing the relevant prefixes – or use a Looking Glass to check

Troubleshooting Connectivity Example III

- A light flow of traffic from AS64502, but 50% less than from AS64503
  - Looking Glass comes to the rescue
    - LG will let you see what AS64502, or AS64502's upstreams are announcing
    - AS64501 may choose this as primary path, but AS64502 relationship with their upstream may decide otherwise
  - NetFlow comes to the rescue
    - Allows AS64501 to see what the origins are, and with the LG, helps AS64501 to find where the prefix filtering culprit might be

Troubleshooting Connectivity Example IV



- Symptom: AS64501 is loadsharing between its upstreams, but the traffic load swings randomly between AS64502 and AS64503

Troubleshooting Connectivity Example IV

- Checklist:
  - Assume AS64501 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

Troubleshooting Connectivity Example IV

- Checklist:
  - Assume AS64501 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
  - RIPE-378 describes impact of route flap damping on Internet
    - www.ripe.net/ripe/docs/ripe-378
    - Strongly discouraged in its current form – RIPE 378 is obsolete!
  - RIPE-580 & RFC7196 suggest improvements
    - www.ripe.net/ripe/docs/ripe-580
    - www.rfc-editor.org/rfc/rfc7196.txt
  - Many network operators still implement route flap damping
  - Many network operators simply use the vendor defaults
    - Vendor defaults are too severe

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 backoff 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

- Tutorial has covered the most common troubleshooting techniques used by network operators 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

ISP Workshops