CISCO

Cisco live!

July 10-14, 2016 • Las Vegas, NV

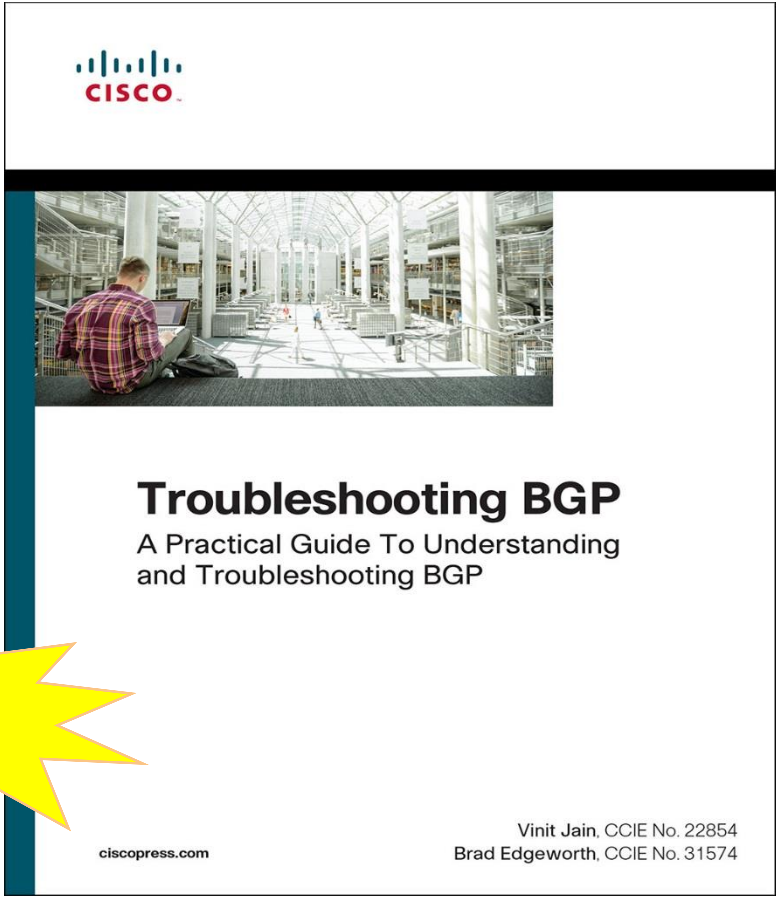
Your Time Is Now

Troubleshooting BGP

Vinit Jain (CCIE# 22854)
@vinugenie

Mani Ganesan (CCIE# 27200)
@mani_cisco

BRKRST-3320



Troubleshooting BGP

A Practical Guide To Understanding and Troubleshooting BGP

ciscopress.com

Vinit Jain, CCIE No. 22854
Brad Edgeworth, CCIE No. 31574

Coming this year

livelessons™

BGP Troubleshooting

Vinit Jain

video

Coming this year

Agenda

- BGP peering issues
  - session not coming up, dynamic peering, session Flapping
- Troubleshooting BGP Convergence Issues
  - BGP slow-peer, BGP PIC
- Troubleshooting BGP Policies
  - Communities, Missing Routes
- BGP for Service Providers
  - MPLS L3 VPNs, BGP RTC

Introduction

Housekeeping

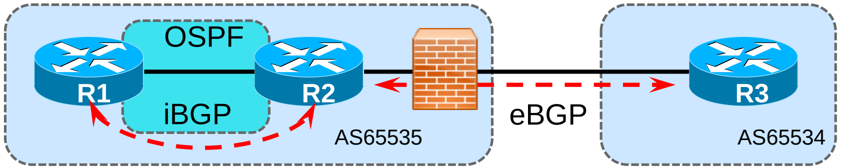
- Who we are?
- Who are you?
  - ✓ Service Provider
  - ✓ Enterprise
  - ✓ Data Center
  - ✓ Studying for CCIE
- “Advanced” Class
  - ✓ Assume BGP Operational Experience
  - ✓ Basic configuration
  - ✓ Show commands
  - ✓ Understand BGP attributes

Troubleshooting Peering Issues

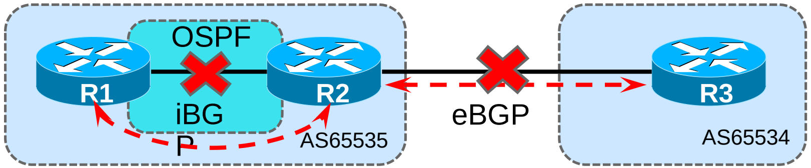
Scenario 1 - Failed BGP Peering

Problem Description

- iBGP / eBGP is not establishing
- Newly configured BGP session not coming up
- Session was up before, but not coming up now



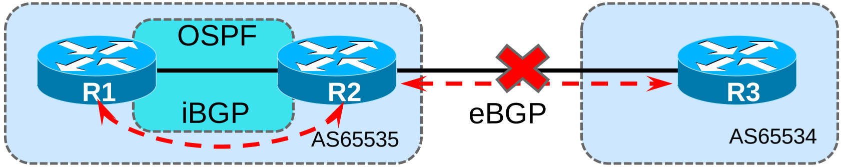
Physical Topology



Logical Topology

Failed BGP peering

Configuration



Check

- ☑ AS Numbers
- ☑ Peering IP
- □ eBGP Multihop?

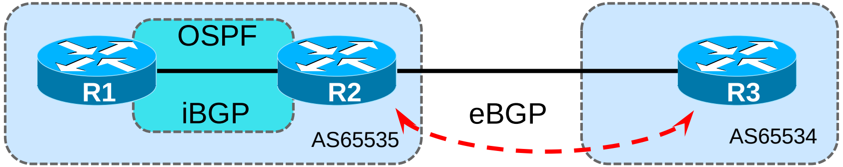
```
router bgp 65535
 bgp router-id 2.2.2.2
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 65535
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 3.3.3.3 remote-as 65534
 . . .
```

```
router bgp 65534
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
neighbor 10.23.23.2 remote-as 65535
neighbor 10.23.23.2 update-source lo0
```

Failed BGP Peering

EBGP-MultiHop – The Old Method

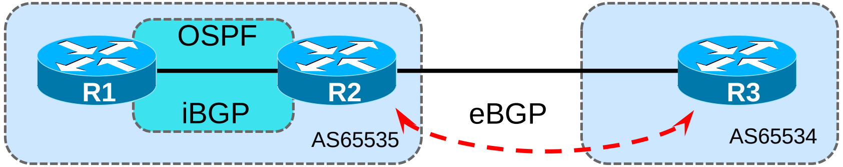
- Loopback peering for EBGP sessions typically used for load-balancing over multiple links
- Use **ebgp-multihop** *hop-count*
- Change the TTL to 2
- Disables the “is the NEXTHOP on a connected subnet” check



```
router bgp 65535
 no synchronization
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 65534
 neighbor 3.3.3.3 ebgp-multihop 2
 no auto-summary
```

Failed BGP Peering

Disable-connected-check



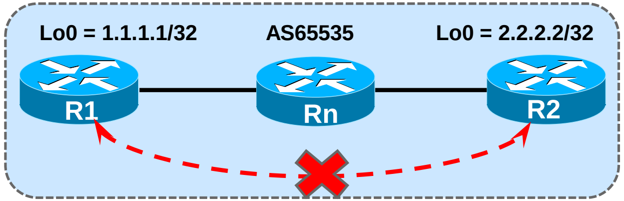
- Use **`neighbor disable-connected-check`**
- TTL remains 1
- Disables the “is the NEXTHOP on a connected subnet” check

```
router bgp 65534
 no synchronization
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 65535
 neighbor 2.2.2.2 disable-connected-check
 neighbor 2.2.2.2 up lo0
```

Failed BGP peering

Reachability – IBGP or EBGP

```
R1# ping 2.2.2.2

Sending 5, 100-byte ICMP Echos to 2.2.2.2,
timeout is 2 seconds:

Packet sent with a source address of 10.12.12.1

!!!!!

Success rate is 0 percent (0/5)
```

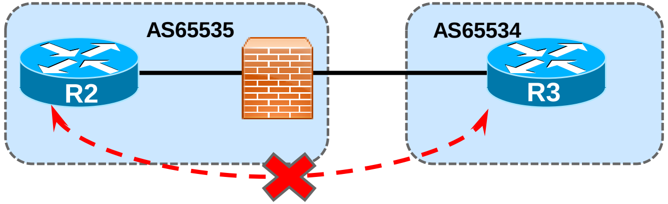


```
R1# ping 2.2.2.2 source loopback0

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

.....

Success rate is 0 percent (0/5)
```

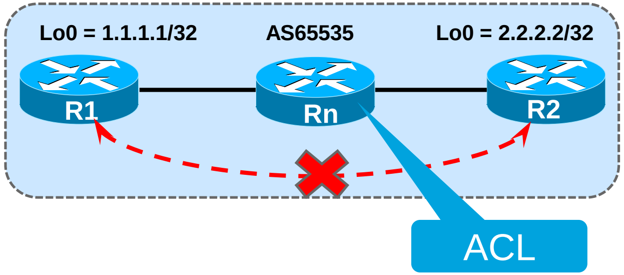
Failed BGP peering

Verify any Firewall / ACL in path for TCP port **179**

```
ASA_FW# sh run access-list
access-list OUT extended permit icmp any any
access-list OUT extended permit ospf any any
access-list OUT extended permit tcp any any eq telnet
. . . .
```



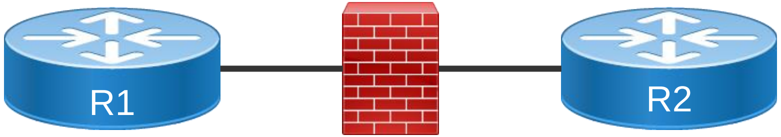
```
Rn# sh ip access-list R1_R2
permit icmp any any
permit ospf any any
permit tcp host 10.12.12.1 eq bgp 2.2.2.2
permit tcp host 10.12.12.1 2.2.2.2 eq bgp
. . . .
```



Securing BGP Connections

BGP Pass-Through

- ASA / PIX offsets TCP sequence number with a random number for every TCP session
  - Causes MD5 authentication to fail
  - ASA strips off TCP option 19



1. Create ACL to permit BGP traffic
2. Create TCP Map to allow TCP option 19
3. Create class-map to match BGP traffic
4. Disable sequence number randomization and Enable TCP option 19 in global policy

Securing BGP Connections

BGP Pass-Through – ASA FW Configuration

```
access-list OUT extended permit tcp host 10.1.12.1 host 10.1.12.2 eq bgp
access-list OUT extended permit tcp host 10.1.12.2 eq bgp host 10.1.12.2
!
access-list BGP-TRAFFIC extended permit tcp host 10.1.110.2 host 10.1.110.10 eq bgp
access-list BGP-TRAFFIC extended permit tcp host 10.1.110.2 eq bgp host 10.1.110.10
!
tcp-map TCP-OPTION-19
tcp-options range 19 19 allow
!
access-group OUT in interface Outside
!
class-map BGP_TRAFFIC
match access-list BGP-TRAFFIC
!
policy-map global_policy
  class BGP_TRAFFIC
    set connection random-sequence-number disable
    set connection advanced-options TCP-OPTION-19
```

Failed BGP peering

Verify TCP session

```
R2#sh tcp brief
TCB         Local Address            Foreign Address        (state)
65F19834    2.2.2.2.179               1.1.1.1.46523          ESTAB
```

Quick test when BGP is down

```
R1#telnet 2.2.2.2 179 /source-interface loopback 0
Trying 2.2.2.2 ...
% Destination unreachable; gateway or host down

R1#
```

- This means BGP Packets are being blocked between R1 and R2

Failed BGP peering

Blocked process in XR

- ✓ Ensure BGP process is in **Run** state.
- ✓ Check for blocked BGP or TCP process on the RP / LC using **show process blocked** command.

```
RP/0/RSP0/CPU0:ASR9010-B# show process bgp
Mon Jun  3 09:47:12.646 EST
                  Job Id: 1040
                     PID: 307494
Executable path: /disk0/iosxr-routing-
4.2.3/bin/bgp
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Tue May 28 14:35:50
2013
           Process state: Run
           Package state: Normal
       Started on config: default
               . . . .
```

Failed BGP peering

Show process bgp (contd. Output)

```
RP/0/RSP0/CPU0:ASR9010-B# show process bgp
<snip>
10    2   488K   10 Nanosleep      0:00:02:0004     0:00:00:0847 bgp
1049    13    3   488K   10 Receive          0:00:00:0811     6:36:52:0264 bgp
1049    14    3   488K   10 Condvar        14:56:55:0236     9:07:49:0890 bgp
1049    15    0   488K   10 Condvar        14:56:55:0240    25:09:49:0542 bgp
1049    16    3   488K   10 Running          0:00:00:0000    57:53:33:0110 bgp
1049    17    1   488K   10 Receive          0:00:28:0379     0:00:00:0066 bgp
1049    18    1   488K   10 Mutex           13:15:50:0870     3:31:49:0712 bgp
<snip>
```

- You can also use "`show process blocked`" to check the blocked processes

Failed BGP peering

Sniffer Capture

Use SPAN to get traffic to your sniffer

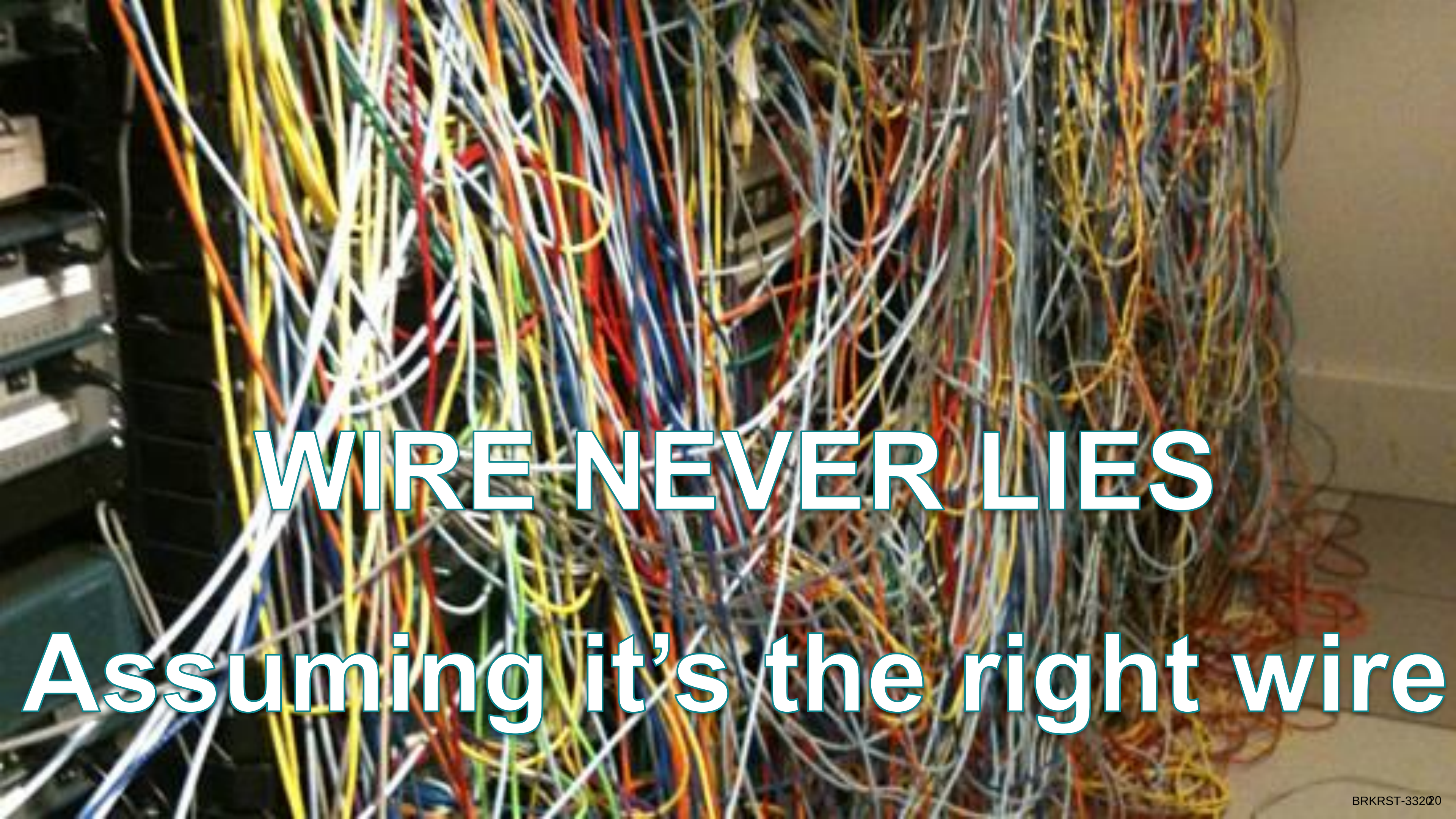
- `monitor session 1 source interface Te2/4 rx`
- `monitor session 1 destination interface Te2/2`

IOS-XR

- Only supported on ASR-9000
- Use ACLs to control what packets to SPAN

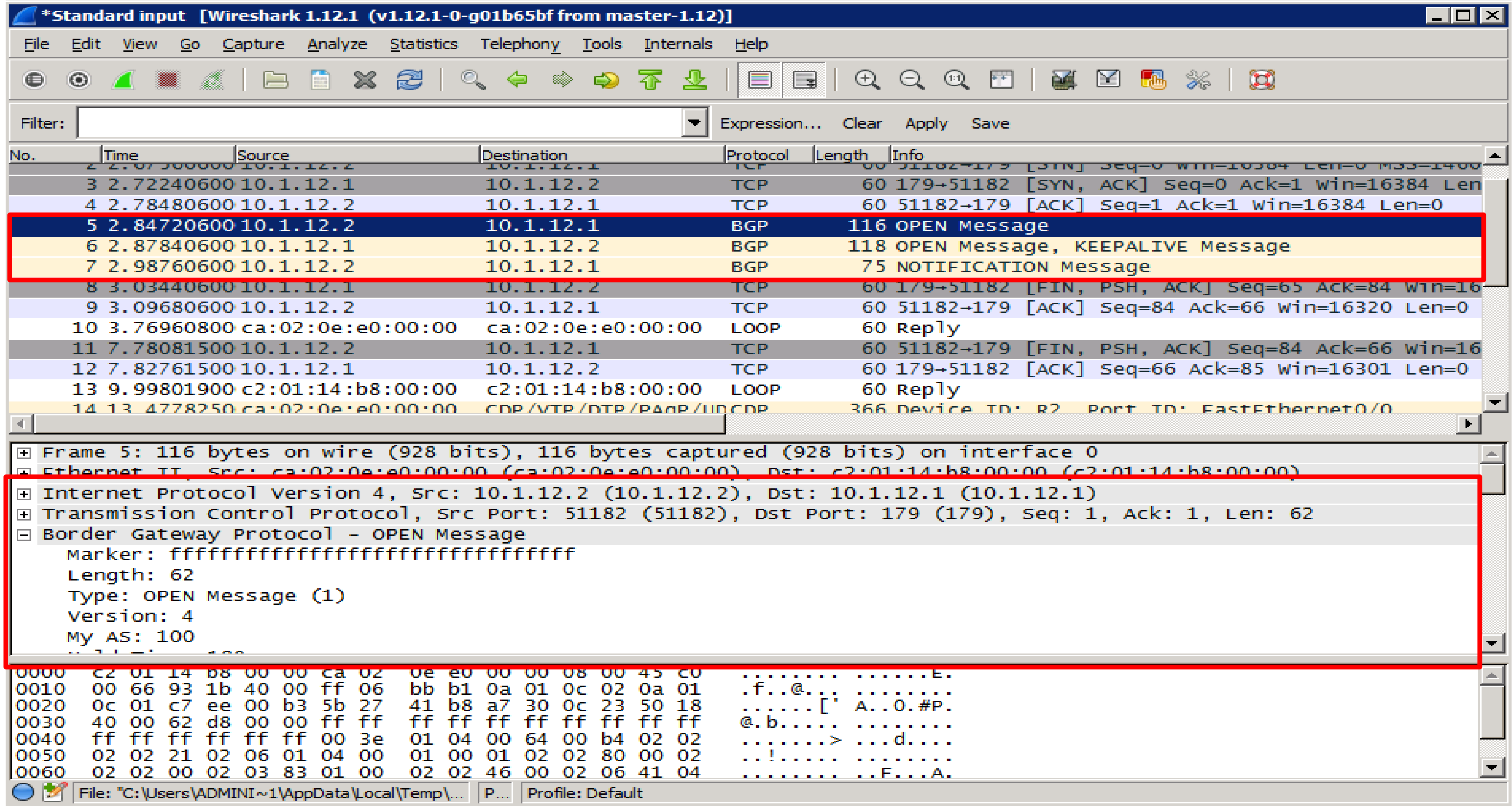
RSPAN

- *"RSPAN has all the features of SPAN, plus support for source ports and destination ports that are distributed across multiple switches, allowing one to monitor any destination port located on the RSPAN VLAN. Hence, one can monitor the traffic on one switch using a device on another switch."*



WIRE NEVER LIES

Assuming it’s the right wire



*Standard input [Wireshark 1.12.1 (v1.12.1-0-g01b65bf from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time        Source             Destination          Protocol Length Info
   2 2.67300000 10.1.12.2          10.1.12.1            TCP       60 51182→179 [SYN] Seq=0 Win=16384 Len=0 MSS=1460
   3 2.72240600 10.1.12.1          10.1.12.2            TCP       60 179→51182 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len
   4 2.78480600 10.1.12.2          10.1.12.1            TCP       60 51182→179 [ACK] Seq=1 Ack=1 Win=16384 Len=0
   5 2.84720600 10.1.12.2          10.1.12.1            BGP      116 OPEN Message
   6 2.87840600 10.1.12.1          10.1.12.2            BGP      118 OPEN Message, KEEPALIVE Message
   7 2.98760600 10.1.12.2          10.1.12.1            BGP       75 NOTIFICATION Message
   8 3.03440600 10.1.12.1          10.1.12.2            TCP       60 179→51182 [FIN, PSH, ACK] Seq=65 Ack=84 Win=16
   9 3.09680600 10.1.12.2          10.1.12.1            TCP       60 51182→179 [ACK] Seq=84 Ack=66 Win=16320 Len=0
  10 3.76960800 ca:02:0e:e0:00:00  ca:02:0e:e0:00:00    LOOP      60 Reply
  11 7.78081500 10.1.12.2          10.1.12.1            TCP       60 51182→179 [FIN, PSH, ACK] Seq=84 Ack=66 Win=16
  12 7.82761500 10.1.12.1          10.1.12.2            TCP       60 179→51182 [ACK] Seq=66 Ack=85 Win=16301 Len=0
  13 9.99801900 c2:01:14:b8:00:00  c2:01:14:b8:00:00    LOOP      60 Reply
  14 13.4778250 ca:02:0e:e0:00:00  CDP/VTP/DTP/PAgP/UD  CDP      366 Device ID: R2  Port ID: FastEthernet0/0
```

```
⊞ Frame 5: 116 bytes on wire (928 bits), 116 bytes captured (928 bits) on interface 0
⊞ Ethernet II, Src: ca:02:0e:e0:00:00 (ca:02:0e:e0:00:00), Dst: c2:01:14:b8:00:00 (c2:01:14:b8:00:00)
⊞ Internet Protocol Version 4, Src: 10.1.12.2 (10.1.12.2), Dst: 10.1.12.1 (10.1.12.1)
⊞ Transmission Control Protocol, Src Port: 51182 (51182), Dst Port: 179 (179), Seq: 1, Ack: 1, Len: 62
⊟ Border Gateway Protocol - OPEN Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 62
    Type: OPEN Message (1)
    Version: 4
    My AS: 100
    Hold Time: 180
```

```
0000  c2 01 14 b8 00 00 ca 02  0e e0 00 00 08 00 45 c0   ........ ......E.
0010  00 66 93 1b 40 00 ff 06  bb b1 0a 01 0c 02 0a 01   .f..@... ........
0020  0c 01 c7 ee 00 b3 5b 27  41 b8 a7 30 0c 23 50 18   ......['  A..0.#P.
0030  40 00 62 d8 00 00 ff ff  ff ff ff ff ff ff ff ff   @.b..... ........
0040  ff ff ff ff ff ff 00 3e  01 04 00 64 00 b4 02 02   .......> ...d....
0050  02 02 21 02 06 01 04 00  01 00 01 02 02 80 00 02   ..!..... ........
0060  02 02 00 02 03 83 01 00  02 02 46 00 02 06 41 04   ........ ..F...A.
```

File: "C:\Users\ADMINI~1\AppData\Local\Temp\... P... Profile: Default

Failed BGP peering

For your reference only

Platform Specific Packet Capture Tools

IOS

- ✓ Embedded Packet Capture

6500 / 7600

- ✓ ELAM
- ✓ NETDR Capture
- ✓ MPA (Mini Protocol Analyzer)

ASR9000

- ✓ Network Processor Capture

Nexus (7k, 5k, 3k)

- ✓ Ethanalyzer
- ✓ Elam

Failed BGP peering

7600 – Netdr Capture

```
7600-RTR#debug netdr capture rx source-ip-address 10.1.13.1
7600-RTR#show netdr captured-packets
A total of 2 packets have been captured
The capture buffer wrapped 0 times
Total capture capacity: 4096 packets
------- dump of incoming inband packet -------
interface Te1/4, routine process_rx_packet_inline, timestamp 15:20:07.111
dbus info: src_vlan 0x3F8(1016), src_indx 0x3(3), len 0x4F(79)
  bpdu 0, index_dir 0, flood 0, dont_lrn 0, dest_indx 0x380(896)
  48020400 03F80400 00030000 4F000000 00060408 0E000008 00000000 0380E753
destmac 00.1E.F7.F7.16.80, srcmac 84.78.AC.0F.76.C2, protocol 0800
protocol ip: version 0x04, hlen 0x05, tos 0xC0, totlen 61, identifier 7630
  df 1, mf 0, fo 0, ttl 1, src 10.1.13.1, dst 10.1.13.3
  tcp src 179, dst 11655, seq 788085885, ack 4134684341, win 17520 off 5
checksum 0x5F4E ack psh
```

Failed BGP peering

ASR1k – EPC Capture

```
ASR1k(config)#ip access-list extended MYACL
ASR1k(config-acl)#permit tcp any eq bgp any
ASR1k(config-acl)#permit tcp any any eq bgp
ASR1k#monitor capture CAP1 buffer circular packets 1000
ASR1k#monitor capture CAP1 buffer size 10
ASR1k#monitor capture CAP1 interface GigabitEthernet0/0/0 in
ASR1k#monitor capture CAP1 access-list MYACL
ASR1k#monitor capture CAP1 start
ASR1k#monitor capture CAP1 stop
ASR1k#monitor capture CAP1 export bootflash:cap1.pcap
```

Failed BGP peering

ASR1k – EPC Capture

```
ASR1k#show monitor capture buffer CAP1 dump
16:25:44.938 JST Aug 21 2015 : IPv4 LES CEF    : Gig0/0 None

F19495B0:                   AABBCC00 0800AABB          *;L...*;
F19495C0: CC000700 08004540 003B1C5D 4000FE06  L.....E@.;.]@.~.
F19495D0: 42020707 07070808 08084A07 00B39372  B.........J..3.r
F19495E0: FFE37CDC E3D35018 3D671161 0000FFFF  .c|\cSP.=g.a....
F19495F0: FFFFFFFF FFFFFFFF FFFFFFFF FD        ............}
```

Failed BGP peering

Nexus - Ethanalyzer

```
N7K-1#ethanalyzer local interface inbound-hi display-filter "bgp" limit-captured-frames 0
Capturing on 'eth4'
1 wireshark-cisco-mtc-dissector: ethertype=0xde09, devicetype=0x0
wireshark-broadcom-rcpu-dissector: ethertype=0xde08, devicetype=0x0
<snip>
2  81 2015-09-01 04:50:34.115833 192.168.10.2 -> 192.168.10.1 BGP 236 OPEN Message
5  86 2015-09-01 04:50:34.259108 192.168.10.1 -> 192.168.10.2 BGP 200 OPEN Message
 87 2015-09-01 04:50:34.259440 192.168.10.1 -> 192.168.10.2 BGP 149 KEEPALIVE Me
ssage
 88 2015-09-01 04:50:34.271319 192.168.10.2 -> 192.168.10.1 BGP 185 KEEPALIVE Me
ssage
6  92 2015-09-01 04:50:35.272488 192.168.10.1 -> 192.168.10.2 BGP 178 UPDATE Messa
ge, KEEPALIVE Message
8  93 2015-09-01 04:50:35.288438 192.168.10.2 -> 192.168.10.1 BGP 214 UPDATE Messa
ge, KEEPALIVE Message
 94 2015-09-01 04:50:35.288813 192.168.10.2 -> 192.168.10.1 BGP 214 UPDATE Messa
ge, KEEPALIVE Message
```

Dynamic BGP peering

BGP Dynamic Neighbors

Allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses

BGP passively listens to configured address range for incoming sessions

BGP neighbor dynamically created

- Remote address is source of TCP connection
- Config template associated with listen range is applied

Provisioning

- No manual config necessary on hub for new clients
- Significant reduction in config overhead

Dynamic BGP Peering

Configuration

```
router bgp 65535
 neighbor Test peer-group
 bgp listen limit 300
   bgp listen range 192.168.0.0/16 peer-group Test
   neighbor Test remote-as 300 alternate-as 200
  !
  address-family ipv4 unicast
    neighbor Test activate
```

Creating a global limit of BGP dynamic subnet range neighbors

Configuring subnet range and associating with a peer group

Associating Autonomous System numbers for listen range peers

- Max Listen Limit – 5000
- Alternate-as limit is 5 (Config only used with listen range peer-groups)

Dynamic BGP Peering

Verifying Dynamic BGP Peers

```
R2# show ip bgp summary

BGP router identifier 192.168.3.1, local AS number 65535
BGP table version is 1, main routing table version 1

Neighbor       V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
*192.168.3.2   4 200   2       2       1    0    0 00:00:37        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 1/(300 max), Subnet ranges: 1

BGP peergroup groupTAC listen range group members:
192.168.0.0/16
```

Dynamic BGP Peering

Most common issues

MD5 related

- Make sure MD5 password is configured at both ends if see the error such as "**`MD5 received, but NOT expected from..`**" message

Resource issues in a scaled environment

Security issues if the range is not carefully defined

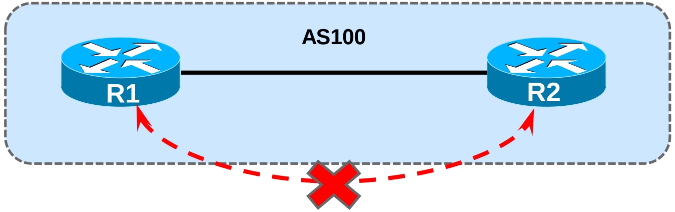
Lab TRY - Try removing '`bgp listen range`' and add it back

- Only try it in lab for testing purposes, not in live production

TAC Case Example - 1

BGP Peering down

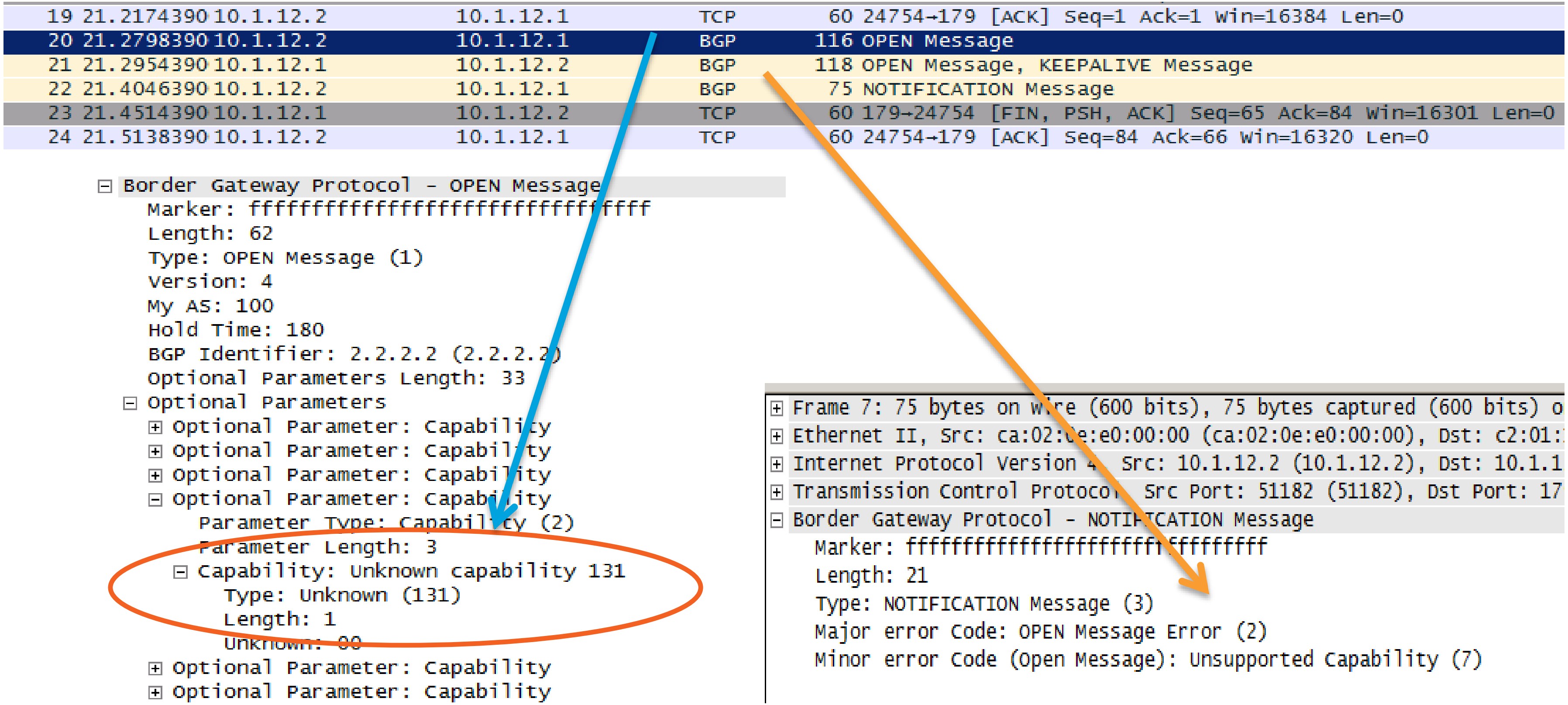
- Customer reported new iBGP peer not coming up with a different vendor device
- Configuration verified
- “**show ip bgp summary**” shows BGP state changes from **Idle** to **Active** and then to **Closing** state
- TCP session goes to **Established** but then immediately moves to **CloseWait**



TAC Case Example - 1

show log | in BGP

```
R2#

*Jun  5 18:18:04.667: %BGP-3-NOTIFICATION: sent to neighbor
10.1.12.1 active 2/7 (unsupported/disjoint capability) 0 bytes

R2#

*Jun  5 18:18:04.671: %BGP-4-MSGDUMP: unsupported or mal-
formatted message received from 10.1.12.1:

FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 002D 0104 0064 00B4
0101 0101 1002 0601 0400 0100 0102 0280 0002 0202 00
```



```
19 21.2174390 10.1.12.2      10.1.12.1      TCP     60 24754→179 [ACK] Seq=1 Ack=1 Win=16384 Len=0
20 21.2798390 10.1.12.2      10.1.12.1      BGP    116 OPEN Message
21 21.2954390 10.1.12.1      10.1.12.2      BGP    118 OPEN Message, KEEPALIVE Message
22 21.4046390 10.1.12.2      10.1.12.1      BGP     75 NOTIFICATION Message
23 21.4514390 10.1.12.1      10.1.12.2      TCP     60 179→24754 [FIN, PSH, ACK] Seq=65 Ack=84 Win=16301 Len=0
24 21.5138390 10.1.12.2      10.1.12.1      TCP     60 24754→179 [ACK] Seq=84 Ack=66 Win=16320 Len=0
```

```
⊟ Border Gateway Protocol - OPEN Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 62
    Type: OPEN Message (1)
    Version: 4
    My AS: 100
    Hold Time: 180
    BGP Identifier: 2.2.2.2 (2.2.2.2)
    Optional Parameters Length: 33
  ⊟ Optional Parameters
    ⊞ Optional Parameter: Capability
    ⊞ Optional Parameter: Capability
    ⊞ Optional Parameter: Capability
    ⊟ Optional Parameter: Capability
        Parameter Type: Capability (2)
        Parameter Length: 3
      ⊟ Capability: Unknown capability 131
          Type: Unknown (131)
          Length: 1
          Unknown: 00
    ⊞ Optional Parameter: Capability
    ⊞ Optional Parameter: Capability
```

```
⊞ Frame 7: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) o
⊞ Ethernet II, Src: ca:02:0e:e0:00:00 (ca:02:0e:e0:00:00), Dst: c2:01:
⊞ Internet Protocol Version 4, Src: 10.1.12.2 (10.1.12.2), Dst: 10.1.1
⊞ Transmission Control Protocol, Src Port: 51182 (51182), Dst Port: 17
⊟ Border Gateway Protocol - NOTIFICATION Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 21
    Type: NOTIFICATION Message (3)
    Major error Code: OPEN Message Error (2)
    Minor error Code (Open Message): Unsupported Capability (7)
```

TAC Case Example - 1

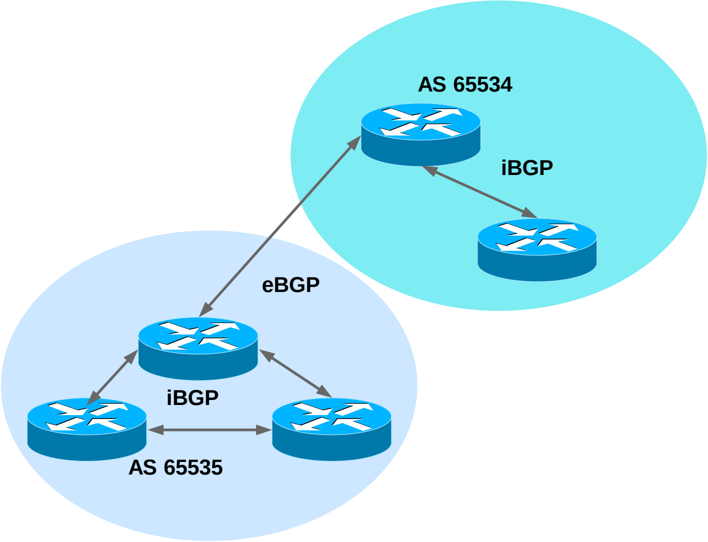
Resolution

- Analysis
  - Capability 131 is set when BGP is trying to establish multisession
  - The other side did not understand this capability i.e. Single-session
- Resolution
  - Configure both sessions to use same capability i.e. Single-session / Multi-session
    - **`neighbor <ip-addr> transport single-session | multi-session`**
  - Disable capability negotiation during session establishment process
    - **`neighbor <ip-addr> dont-capability-negotiate`**

Scenario 2 - BGP Peer Flapping

Problem Description

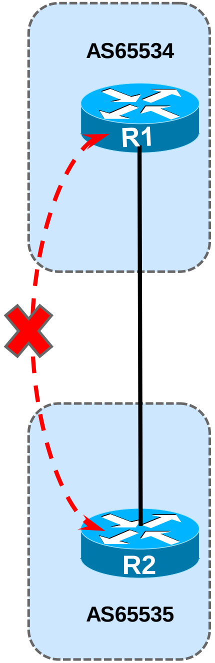
- Multiple BGP Sessions Flapping
- Keeps oscillating between two states (Idle/Established)
- Symptoms
  - Hold time expired notifications
  - High CPU
  - Interface Input-Queue Drops



BGP Peer Flapping

BGP States

- Stuck in IDLE state:
  - No connected route to the peer
- Staying in ACTIVE state:
  - No route to the peer address (IP connectivity is not there)
  - Configuration error, update-source
- Flapping IDLE/ACTIVE:
  - TCP establishes but BGP negotiation fails – Misconfigured AS
- Flapping IDLE/Established:
  - Bad update, TCP problem (MSS size in multi-hop)



BGP Peer Flapping

Notifications

```
R2#
*Mar 24 20:25:47.262: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down BGP
Notification sent
*Mar 24 20:25:47.262: %BGP-3-NOTIFICATION: sent to neighbor 1.1.1.1 4/0
(hold time expired) 0 bytes
```

- BGP NOTIFICATIONs consist of an error code, sub-code and data
  - All Error Codes and Sub-codes can be found here
    - http://www.iana.org/assignments/bgp-parameters/bgp-parameters.xml
    - http://tinyurl.com/bgp-notification-codes
  - Data portion may contain what triggered the notification
    - Example: corrupt part of the UPDATE

Notifications Contd…

```
%BGP-3-NOTIFICATION: sent to neighbor 2.2.2.2 2/2 (peer in wrong AS) 2 bytes 00C8
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 002D 0104 00C8 00B4 0202 0202 1002 0601 0400
0100 0102 0280 0002 0202 00
```

Value	Name	Reference
1	Message Header Error	RFC 4271
2	OPEN Message Error	RFC 4271
3	UPDATE Message Error	RFC 4271
4	Hold Timer Expired	RFC 4271
5	Finite State Machine Error	RFC 4271
6	Cease	RFC 4271

The first 2 in “2/2” is the Error Code….so “OPEN Message Error”

Notifications Contd…

Subcode #	Subcode Name	Subcode Description
1	Unsupported BGP version	The version of BGP the peer is running isn’t compatible with the local version of BGP
2	Bad Peer AS	The AS this peer is locally configured for doesn’t match the AS the peer is advertising
3	Bad BGP Identifier	The BGP router ID is the same as the local BGP router ID
4	Unsupported Optional Parameter	There is an option in the packet which the local BGP speaker doesn’t recognize
6	Unacceptable Hold Time	The remote BGP peer has requested a BGP hold time which is not allowed (too low)
7	Unsupported Capability	The peer has asked for support for a feature which the local router does not support

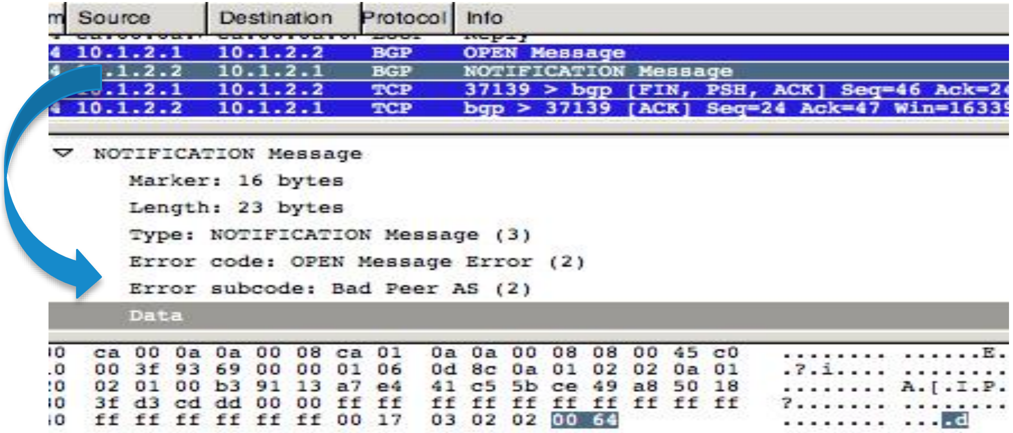
OPEN Message Subcodes shown above
The second 2 in “2/2” is the Error Subcode….so “Bad Peer AS”

BGP Peer Flapping

Notifications

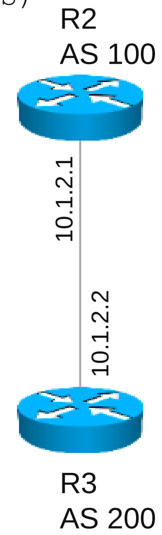
```
R2# show log | include NOTIFICATION
%BGP-3-NOTIFICATION: sent to neighbor 10.1.2.1 2/2 (peer in wrong AS)
2 bytes 0064 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 002D 0104
0064 00B4 0101 0101 1002 0601 0400 0100 0102 0280 0002 0202 00
```

x0064 = "data" of NOTIFICATION
x0064 = decimal 100



```
m Source      Destination  Protocol  Info
4 10.1.2.1    10.1.2.2     BGP       OPEN Message
4 10.1.2.2    10.1.2.1     BGP       NOTIFICATION Message
4 10.1.2.1    10.1.2.2     TCP       37139 > bgp [FIN, PSH, ACK] Seq=46 Ack=2
4 10.1.2.2    10.1.2.1     TCP       bgp > 37139 [ACK] Seq=24 Ack=47 Win=1633

▽ NOTIFICATION Message
      Marker: 16 bytes
      Length: 23 bytes
      Type: NOTIFICATION Message (3)
      Error code: OPEN Message Error (2)
      Error subcode: Bad Peer AS (2)
      Data

0   ca 00 0a 0a 00 08 ca 01  0a 0a 00 08 08 00 45 c0   ........ .......E.
0   00 3f 93 69 00 00 01 06  0d 8c 0a 01 02 02 0a 01   .?.i.... ........
0   02 01 00 b3 91 13 a7 e4  41 c5 5b ce 49 a8 50 18   ........ A.[.I.P.
0   3f d3 cd dd 00 00 ff ff  ff ff ff ff ff ff ff ff   ?....... ........
0   ff ff ff ff ff ff 00 17  03 02 02 00 64            ........ ....d
```

Sniff of BGP Notification Sent from R2 to R1



BGP Peer Flapping

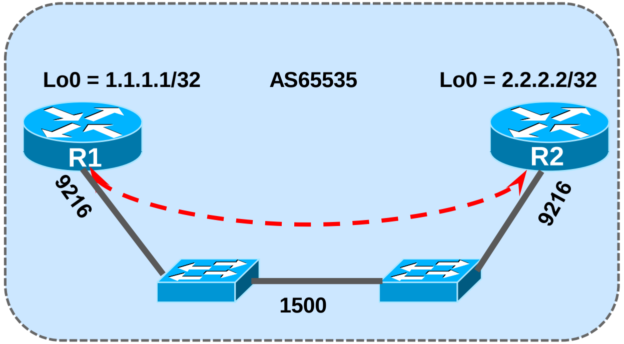
Regular Interval Flaps

```
*Jun 22 15:16:23.033: %BGP-3-NOTIFICATION: received from neighbor
192.168.2.2 4/0 (hold time expired) 0 bytes

*Jun 22 15:16:23.033: %BGP-5-ADJCHANGE: neighbor 192.168.2.2 Down
BGP Notification received

*Jun 22 15:16:55.621: %BGP-5-ADJCHANGE: neighbor 192.168.2.2 Up

*Jun 22 15:19:56.409: %BGP-3-NOTIFICATION: received from neighbor
192.168.2.2 4/0 (hold time expired) 0 bytes

*Jun 22 15:19:56.409: %BGP-5-ADJCHANGE: neighbor 192.168.2.2 Down
BGP Notification received

*Jun 22 15:20:13.361: %BGP-5-ADJCHANGE: neighbor 192.168.2.2 Up
```

BGP Peer Flapping

MTU Mismatch

```
R2#show ip bgp neighbors 1.1.1.1 | in max data segment
Datagrams (max data segment is 9176 bytes):

R2# ping 1.1.1.1 source 2.2.2.2 df-bit size 1500
Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 1.1.1.1, timeout is
2 seconds:
Packet sent with a source address of 2.2.2.2
Packet sent with the DF bit set
!!!!!
R2# ping 1.1.1.1 source 2.2.2.2 df-bit size 9216
Type escape sequence to abort.
Sending 5, 9216-byte ICMP Echos to 1.1.1.1, timeout is
2 seconds:
Packet sent with a source address of 2.2.2.2
Packet sent with the DF bit set
.....
```



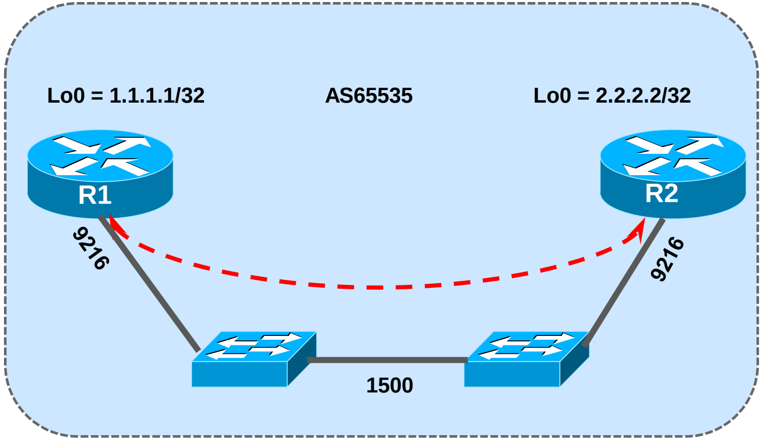
- BGP OPENs and Keepalives are small
- UPDATEs can be much larger
- Maybe small packets work but larger packets do not?

MTU – 20byte IP header – 20 byte TCP header = MSS

Failed BGP Peering

Path MTU Discovery

- R1 sends a packet with packet size of outgoing interface MTU and DF-bit set
- Intermittent device who has lower MTU has two options
  - Fragment and send the packets (if DF-bit not set)
  - Drop the packet and send ICMP error message Type 3 Code 4
- ICMP error message also have the MTU details in the Next-Hop MTU field
- Source on receiving the message, sends the packet with mentioned MTU.

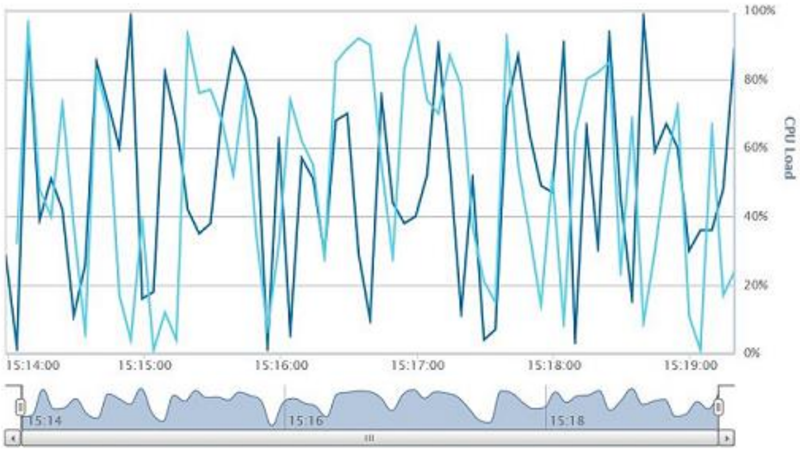


Type 3 – Destination Unreachable
Code 4 – Fragmentation needed and DF-bit set

Failed BGP peering

High CPU – Victim BGP

- High CPU can cause data and control plane packet loss
- High CPU can be caused due to process or interrupt (traffic hitting CPU)
- Example – Packets with TTL set to 1 are punted to CPU



```
PE1_RTR# sh proc cpu sorted | ex 0.00
CPU utilization for five seconds: 92%/91%; one minute: 14%; five minutes: 8%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
  82        7490       929       8062  0.29%  2.96%  1.52%   0 Exec
   4     6308484    908039       6947  0.05%  0.08%  0.06%   0 Check heaps
```

Failed BGP peering

Interface Input-Queue Drops and CoPP Drops

```
PE_RTR# sh int gi 0/1 | in drop
  Input queue: 5/75/4351/0 (size/max/drops/flushes); Total output drops: 0

PE_RTR# show policy-map control-plane
 Control Plane
. . . .
Class-map: Routing (match-any)
      1441 packets, 195499 bytes
      5 minute offered rate 5000 bps, drop rate 1000 bps
      Match: access-group name CPP-Routing
      police:
          cir 10000000 bps, bc 312500 bytes
        conformed 271 packets, 40286 bytes; actions:
          transmit
        exceeded 57 packets, 61143 bytes; actions:
          drop
        conformed 5000 bps, exceed 1000 bps
```

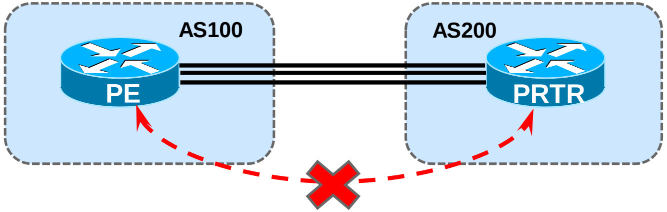
Input-Queue drops can also lead to multiple Control-Plane packet loss along with data loss

Drops in CoPP policy can cause BGP sessions to flap

TAC Case Example - 2

BGP Session Flapping

- Multiple BGP sessions flapping noticed on ATM circuit towards a partner router following an IOS upgrade on the router
- No Changes in configuration
- Previous "**show ip bgp summary**" shows all the other neighbors were stable
- Requested "`show ip bgp neighbor <nei-ip>`" command before and after upgrade



```
router bgp 100
address-family vrf ABC
neighbor 12.12.12.2 remote-as 200
 neighbor 12.12.12.2 password cisco
 neighbor 12.12.12.2 activate
 neighbor 12.12.12.2 send-community
```

TAC Case Example - 2

MTU Analysis

Before Upgrade

```
Datagrams (max data segment is 1440 bytes):

Rcvd: 517011 (out of order: 1), with data: 232834, total data bytes: 4674384

Sent: 525432 (retransmit: 6796 fastretransmit: 5),with data: 295508, total data bytes: 6886010
```

After Upgrade

```
Datagrams (max data segment is 1460 bytes):

Rcvd: 166 (out of order: 0), with data: 76, total data bytes: 2203

Sent: 168 (retransmit: 2 fastretransmit: 0),with data: 92, total data bytes: 2555
```

TAC Case Example - 2

MTU Calculation

Before Upgrade – Wrong Calculation

MSS = MTU – (IP Header + TCP Header) – Optional Bytes

After Upgrade – Correct Calculation

MSS = MTU – (IP Header + TCP Header)

The old code had a different behavior of calculating the MSS value

TAC Case Example - 2

Resolution

- Globally change the TCP MSS negotiation value

  - ip tcp mss <mss-value>

- Remove password authentication for the affected neighbors

Troubleshooting Summary – Peering Issues

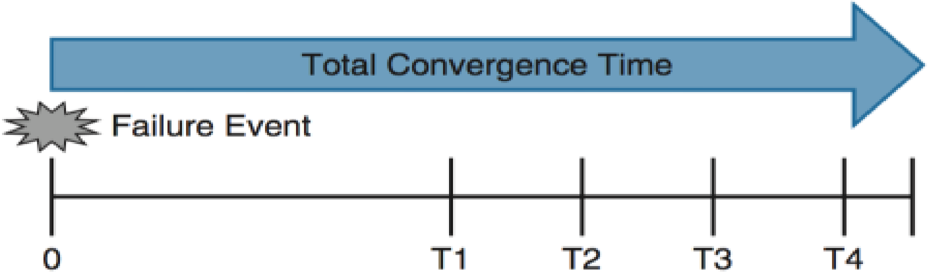
- BGP Peering Down
  - Verify BGP configuration (router-config, ACL, Firewall, etc.)
  - Verify reachability between peers sourcing the peering IP
  - Check the TCP table for the active sessions (show tcp brief)
  - Packet Captures
- Peer Flapping
  - Analyze the Notifications generated in syslogs
  - Verify MTU in the path
  - Verify CPU utilization and history
  - Drops (interface, CoPP)

Troubleshooting BGP Convergence Issues

Scenario 3 - Troubleshooting BGP Convergence

Problem Description

- BGP Table is getting updated slowly
- Traffic loss (Traffic Black-Hole) is experienced
- High CPU



Troubleshooting BGP Convergence

What is convergence in terms of BGP?

- Establish sessions with a number of peers
- Locally generate all the BGP path (either via network command, redistribute static/connected/IGP), and/or from other component for other address family
  - e.g. MVPN from multicast, L2VPN from l2vpn mgr, EVPN from evpn mgr, etc.
- Send and receive multiple BGP tables (different BGP address-families) to/from each peer
- Upon receive all the paths from peers, do the best path calculation to find the best path (and/or multi path, additional-path, backup path, etc.)

Troubleshooting BGP Convergence

What is convergence in terms of BGP?

- If import/export is involved, the import/export of all kind of variations
  - VRF import, AF import, global import, MVPN import, EVPN import, etc.
- Install the best into multiple routing table
  - Default RIB or VRF, IPv4/IPv6
- For other address family, pass the path calculation result to different lower layer components like step 2 (mvpn, evpn, l2vpn, etc.)

Troubleshooting BGP Convergence

Dimensional Factors

- Number of peers
- Number of address-families
- Number of path/prefix per address-family
- Link speed of individual interface, individual peer
- Different update group settings and topology
- Complexity of attribute creation / parsing for each address-family

Complex Routing Policy – IOS XR

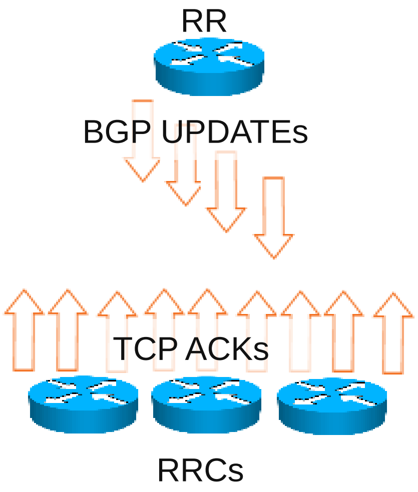
```
as-path-set match-ases
  ios-regex '^(.*65531)$',
  ios-regex '^(.*65532)$',
  ios-regex '^(.*65533)$',
     <snip>
prefix-set K1-routes
  10.170.53.0/24
end-set
prefix-set K2-routes
  10.147.4.0/24
end-set
prefix-set K3-routes
  198.168.44.0/23,
  198.168.46.0/24
end-set
```

```
route-policy Inbound-ROUTES
  if destination in K1-routes then
    pass
  elseif destination in K2-routes then
    pass
  elseif destination in K3-routes then
    pass
else
    drop
  endif
end-policy
!
router bgp 65530
neighbor-group IGW
  remote-as 65530
address-family ipv4 unicast
route-policy Inbound-ROUTES in
```

Convergence

Dropping TCP Acks

- Primarily an issue on RRs (Route Reflectors) with
  - One or two interfaces connecting to the core
  - Hundreds of RRCs (Route Reflector Clients)
- RR sends out tons of UPDATES to RRCs
- RRCs send TCP ACKs
- RR core facing interface(s) receive huge wave of TCP ACKs



Convergence

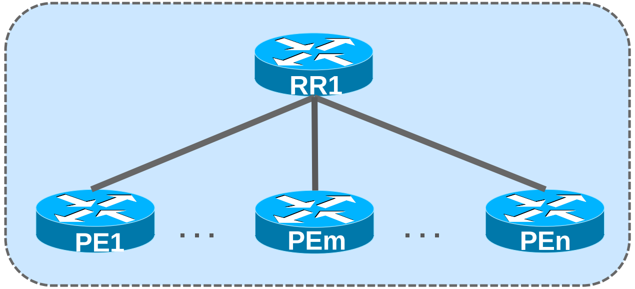
Dropping TCP Acks

- Interface input queue fills up…TCP ACKs are dropped ☹
  - Each time a TCP packet is dropped, the session goes into slow start
  - It takes a good deal of time for a TCP session to come out of slow start
- Increase the input queue
  - **hold-queue 1000 in**
- If you still see drops increase to 4096

Troubleshooting BGP Convergence

Update Groups

- Update Group is a collection of peers with identical outbound policy.
- Helps in improving IBGP convergence
  - Update messages are formatted and replicated to all the peers
- A Master is selected in the update group, which is updated first in the group
- Based on the message formatted for the master / Leader, all the peers are then replicated with the same formatted message
  - The message formatting only happens once.



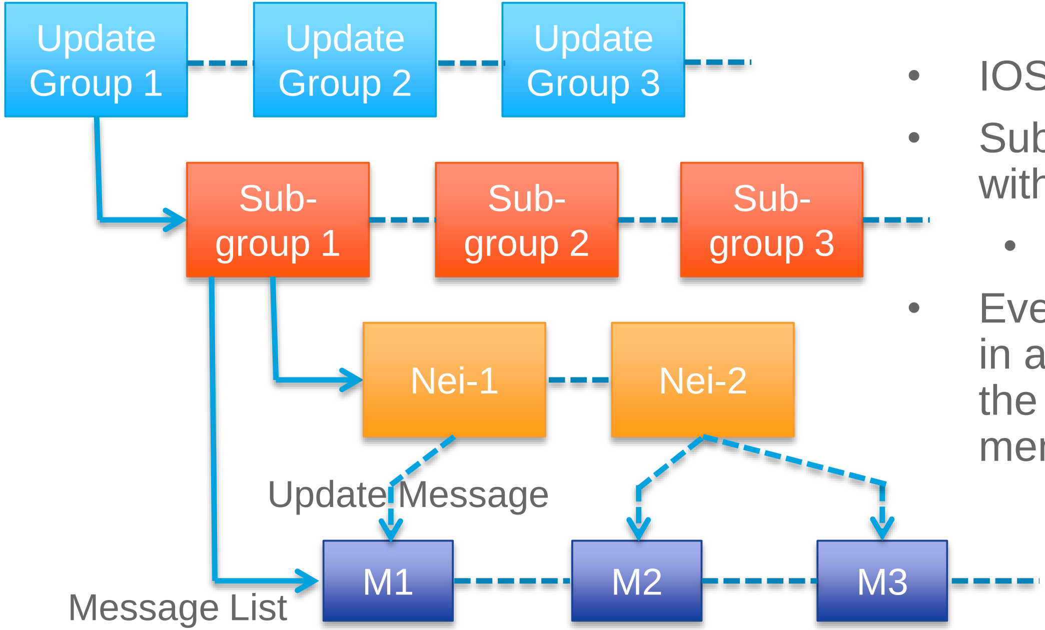
Troubleshooting BGP Convergence

Update Groups

```
R1#show bgp ipv4 unicast update-group
BGP version 4 update-group 2, internal, Address Family: IPv4 Unicast
  BGP Update version : 7/0, messages 0, active RGs: 1
  Route-Reflector Client
  Route map for outgoing advertisements is dummy
  Topology: global, highest version: 7, tail marker: 7
  Format state: Current working (OK, last not in list)
                Refresh blocked (not in list, last not in list)
  Update messages formatted 4, replicated 15, current 0, refresh 0, limit
1000
  Number of NLRIs in the update sent: max 1, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 4 members:
   10.1.12.2        10.1.13.2*        10.1.14.2        10.1.15.2
```

Troubleshooting BGP Convergence

Update Groups on IOS XR

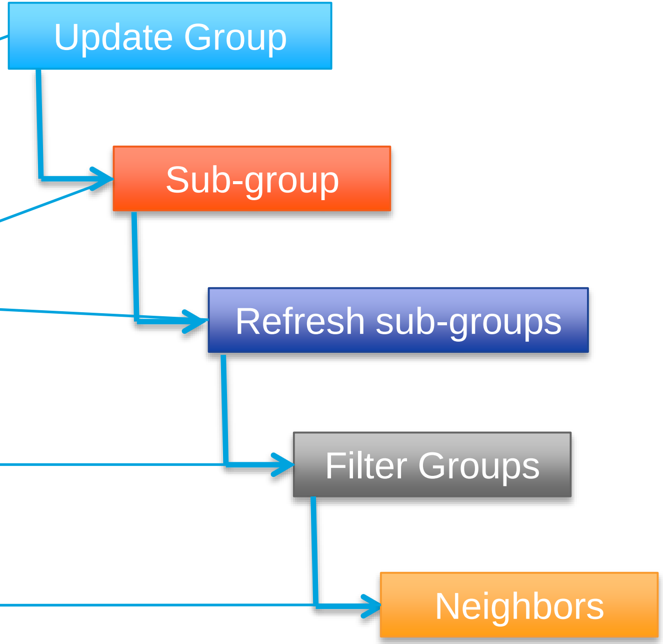


- IOS XR have hierarchical update groups
- Sub-Groups are subset of neighbors within an update Group
  - Neighbors running at same pace
- Even a newly configured neighbor is put in a separate sub-group till it reaches the same table version as other members

Troubleshooting BGP Convergence

Update Groups on IOS XR

```
RP/0/0/CPU0:R10#show bgp update-group
Update group for IPv4 Unicast, index 0.2:
<snip>
Sub-groups merged: 5
Number of refresh subgroups: 0
Messages formatted: 36, replicated: 68
All neighbors are assigned to sub-group(s)
    Neighbors in sub-group: 0.2, Filter-Groups num:3
    Neighbors in filter-group: 0.3(RT num: 3)
        10.1.100.1
    Neighbors in filter-group: 0.1(RT num: 3)
         10.1.100.2
    Neighbors in filter-group: 0.2(RT num: 3)
         10.1.100.8
```



Verify TCP Stats – IOS XR

```
RP/0/8/CPU0:R10#show tcp brief | include 10.1.102.2
0x10146a20 0x60000000  0  0  10.1.102.1:62233  10.1.102.2:179 ESTAB
```

```
RP/0/8/CPU0:R10#show tcp stat pcb 0x10146a20 location 0/8/CPU0
=================================================================
 Statistics for PCB 0x10146a20, vrfid 0x60000000
Send:    0 bytes received from application
            <snip>
         0 packets failed getting queued to network (v4/v6 IO)
         0 packets failed getting queued to network (NetIO)
Rcvd:    722 packets received from network
         380 packets queued to application
         0 packets failed queuing to application
```

Verify TCP NSR Stats – IOS XR

```
RP/0/8/CPU0:R10#show tcp nsr statistics pcb 0x10146a20
PCB 0x10146a20
Number of times NSR went up: 1
Number of times NSR went down: 0
Number of times NSR was disabled: 0
Number of times switch-over occured : 0
IACK RX Message Statistics:
  Number of iACKs dropped because SSO is not up            : 0
  Number of stale iACKs dropped                            : 0
  Number of iACKs not held because of an immediate match   : 0
TX Messsage Statistics:
    Data transfer messages:
        Sent 118347, Dropped 0, Data (Total/Avg.) 2249329/19
                <SNIP>
```

Troubleshooting BGP Convergence – IOS XR

Show bgp all all convergence

```
RP/0/0/CPU0:R10# show bgp all all convergence
Address Family: IPv4 Unicast
====================================
Converged.
All received routes in RIB, all neighbors updated.
All neighbors have empty write queues.

Address Family: VPNv4 Unicast
====================================
Not converged.
Received routes may not be entered in RIB.
One or more neighbors may need updating.
```

Not converged – implies that there are BGP neighbors that for which the replication has not completed yet

Troubleshooting BGP Convergence – IOS XR

Verifying Performance Statistics

```
0/0/CPU0:R10#sh bgp ipv4 uni update-gr 0.2 performance-statistics
Update group for IPv4 Unicast, index 0.2:
  <snip>
  Messages formatted: 0, replicated: 0
  All neighbors are assigned to sub-group(s)
    Neighbors in sub-group: 0.1, Filter-Groups       1
     Neighbors in filter-group: 0.1(RT num: 0
      10.1.102.2    10.1.103.2    10.1.104      10.1.105.2
  Updates generated for 0 prefixes in 10 calls(best-external:0)
             (time spent: 10.000 secs)
 <snip>
```

Verify the time spent in generating and replicated the updates

BGP Convergence – NX-OS

Show bgp convergence detail

```
R20# show bgp convergence detail vrf all
Global settings:
BGP start time 5 day(s), 13:55:45 ago
Config processing completed 0.119865 after start
BGP out of wait mode 0.119888 after start
LDP convergence not required
Convergence to ULIB not required
Information for VRF default
Initial-bestpath timeout: 300 sec, configured 0 sec
BGP update-delay-always is not enabled
First peer up 00:09:18 after start
Bestpath timer not running
 Contd…
```

Troubleshooting BGP Convergence – NX-OS

Show bgp convergence detail

```
Contd…
IPv4 Unicast:
   First bestpath signalled 00:00:27 after start
   First bestpath completed 00:00:27 after start
   Convergence to URIB sent 00:00:27 after start
   Peer convergence after start:
    10.1.202.2          (EOR after bestpath)
    10.1.203.2          (EOR after bestpath)
    10.1.204.2          (EOR after bestpath)
    10.1.205.2          (EOR after bestpath)
```

If bestpath is received before EOR or peer fails to send EOR marker, it can lead to traffic loss

Troubleshooting BGP Convergence – NX-OS

Enable Debugging using Filters

```
debug bgp events updates rib brib import
debug logfile bgp
debug-filter bgp vrf vpn1
debug-filter bgp address-family ipv4 unicast
debug-filter bgp neighbor 10.1.202.2
debug-filter bgp prefix 192.168.2.2/32
```

Troubleshooting BGP Convergence – NX-OS

When Route is not downloaded into URIB

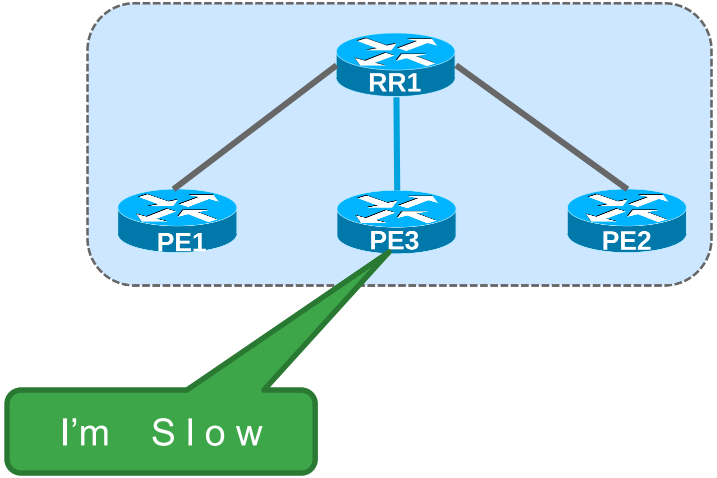
When the route is not download into URIB, it may not be a problem with BGP.

- Show routing internal event-history ufdm
- Show routing internal event-history ufdm-summary
- Show routing internal event-history recursive

Scenario 4 – BGP Slow Peer

Problem Description

- Customer reports updates not getting across all PE routers
- Caused due to:
  - RR's sending updates with high speed
  - Slow processing peers
- Symptoms
  - High CPU due to BGP
  - Updates not replicated to all peers
  - Router reloads



BGP OutQ & Cache Size

- OutQ column should show very high OutQ value
- Should be reaching the maximum cache size for that update-group

```
Router# show ip bgp vpnv4 all summary
..
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
12.123.67.97    4   109      42   87065        0    0 1000 00:10:00        0
12.122.78.19    4   109      42   87391        0    0  674 00:10:00        0

Router# show ip bgp vpnv4 all replication
                                                                    Current Next
Index  Members          Leader       MsgFmt    MsgRepl     Csize    Version Version
    1      348    12.123.67.97   1726595727 1938155978  999/1000 1012333000/1012351142
    2        2    12.122.78.19     79434677   79398843    0/200  1012351503/1012351503
    3        1   199.37.187.24            0          0    0/100           0/0
    4        2   12.122.78.249     79219618   97412908    0/200  1012351504/1012351504
```

TCP sndwnd

```
Router#show neighbor 10.1.0.1
..
iss: 3662804973  snduna: 3668039487  sndnxt: 3668039487     sndwnd:      0
irs: 1935123434  rcvnxt: 1935222998  rcvwnd:      16003  delrcvwnd:    381

SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 512 ms, ACK hold: 200 ms
Status Flags: passive open, gen tcbs
Option Flags: nagle, path mtu capable
```

- Check for send window (sndwnd) and receive window (rcvwnd) using “show ip bgp neighbor <x.x.x.x>”
- For the TCP session for which outQ is high, we might notice that sndwd is very low or zero.
- On the remote end, we should see the rcvwnd value is very low or zero.

Solution - Static Slow peer

- The manual knob to flag a peer as slow will create a separate update group for the peer.
- The advantage - there is a limit to the overhead that this feature will create.
- The drawback - slow member update group will have to progress at the pace of the slowest of the slow peers.

```
neighbor {<nbr-addr>/<peer-grp-name>} slow-peer split-update-group static
```

- This command will manually mark a neighbor as slow peer.
- The peer will be part of slow update group.

Solution - Dynamic Slow peer

- IOS BGP will monitor the transmission speeds of the peers.
- A peer will have to be exhibiting slowness for several minutes to be flagged.
- Log message for when a slow peer is detected/recovered

```
bgp slow-peer detection [threshold <seconds>]

neighbor {<nbr-addr>/<peer-grp-name>} slow-peer detection [threshold < seconds >]
```

- The threshold defines “the threshold time in seconds” to detect a peer as slow peer.
- The range is 120 seconds to 3600 seconds. Default is 300 seconds.

Solution - Slow peer protection

- Depends on Dynamic Slow Peer feature
- When a slow peer recovery is detected (the peer has converged), the peer will be moved back to its original group

```
bgp slow-peer split-update-group dynamic [permanent]

neighbor {<nbr-addr>/<peer-grp-name>} slow-peer split-update-group dynamic [permanent]
```

- When “permanent” is not configured, the “slow peer” will be moved to its regular original update group, after it becomes regular peer (converges).
- If “permanent” is configured, the peer will not be moved to its original update group automatically

Syslog Messages

- The below log message will be generated when a peer is detected as dynamic slow peer.

```
"bgp neighbor %s in af %d is detected as slow-peer"
```

- The below log message will be generated when a “slow-peer” recovers.

```
"slow bgp peer %s in af %d has recovered"
```

BGP Slow Peer - Commands

- Show Commands

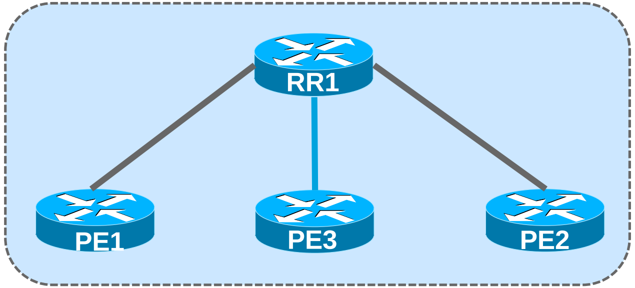
```
show ip bgp [AF/scope/topo] update-group summary slow

show ip bgp [AF/scope/topo] summary slow

show ip bgp [AF/scope/topo] neighbor slow
```

- Clear Commands

```
Clear [ip] bgp <nbr-addr> slow

Clear [ip] bgp peer-group <group-name> slow

Clear [ip] bgp af * slow

Clear ip bgp * slow
```

TAC Case Example - 3

BGP Slow Peer



- Customer reported routes were stuck in BGP RR.
- Their end-customer removed service from one of their locations but the routes are still seen on their RR and other locations
- Soft clearing the neighborship temporarily resolved the problem but reoccurred again after sometime

```
RR1# show ip bgp vpnv4 all replication
                                                                          Current    Next
Index  Members          Leader      MsgFmt     MsgRepl     Csize     Version Version
    1      150    216.156.3.10   274950548   650809652 2000/2000   421492656/421493582
    2        5     65.106.7.100    41049479   204232170     0/500    421493582/0
    5        1  66.239.189.212    16143960    16143960     0/100    421491282/421493582
```

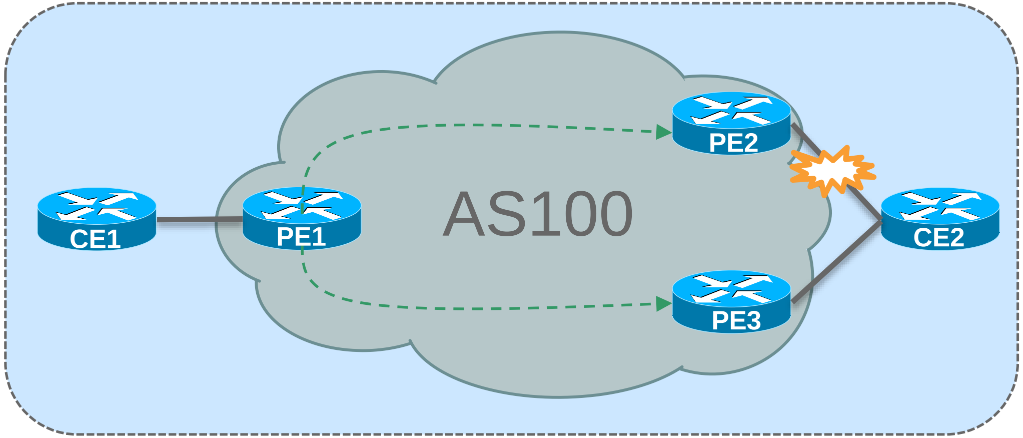
Resolution

- Two neighbors were identified to be showing slow peer symptoms
- Customer’s RR router didn’t had the slow peer capability in the IOS they were running
- Two workarounds / solutions:
  - Create a separate outbound policy for slow peers.
  - Use the "neighbor <ip> advertisement-interval <interval>".
    - Default for internal neighbors is 5 sec and for external is 30 seconds.

Scenario 5 – BGP PIC

Problem Description

- Customer reported they have a multi-homed Customer. But when the primary BGP session goes down, it takes time to converge and the customer experiences a traffic loss
- Caused due to:
  - Convergence issues
- Symptoms
  - Traffic loss

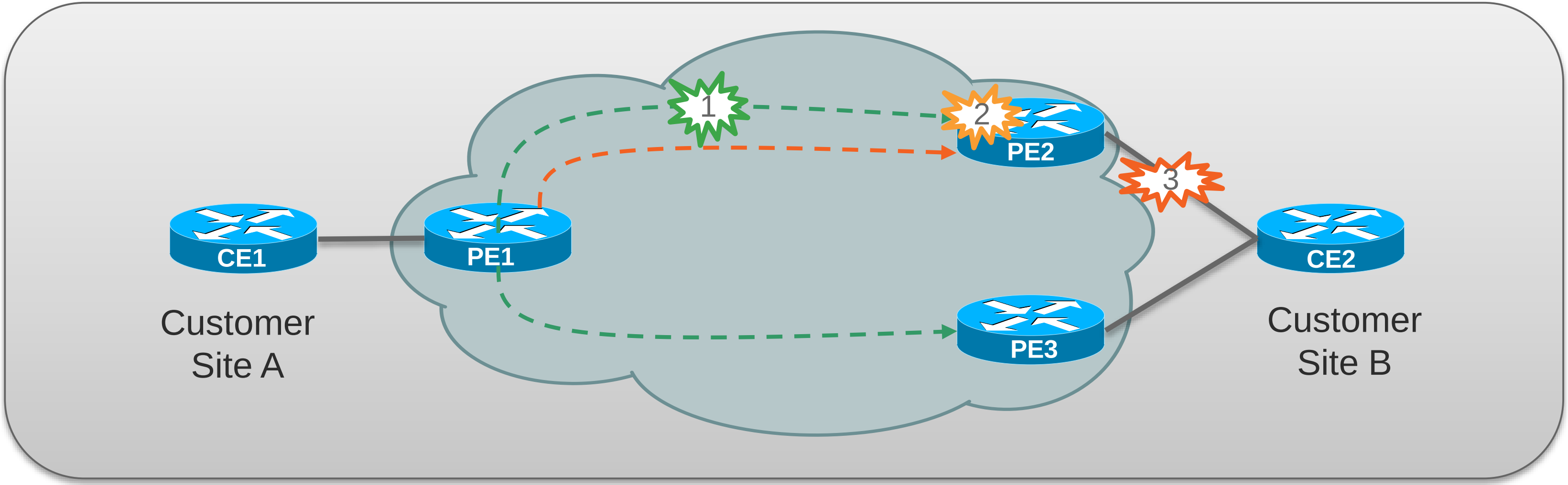


Ciscolive!

What is PIC or BGP FRR?

- **Prefix Independent Convergence** (PIC) in CEF and platform whereby cutover to any backup path happens within sub-seconds and independent of the number of prefixes.
- **BGP Fast Re-Route** (BGP FRR) – enables BGP to use alternate paths within sub-seconds after a failure of the primary or active paths.
- To achieve PIC Edge we require that routing protocols (currently BGP) install backup paths also.
- Without backup paths available to CEF/MFI convergence is driven from the routing protocols updating the RIB and CEF/MFI one prefix at a time, leading to convergence times directly proportional to the number of affected prefixes.
- When backup paths are available, CEF/MFI can use these to provide constant time and prefix independent convergence when a failure affecting a shared path-list occurs.

PIC edge vs. PIC core

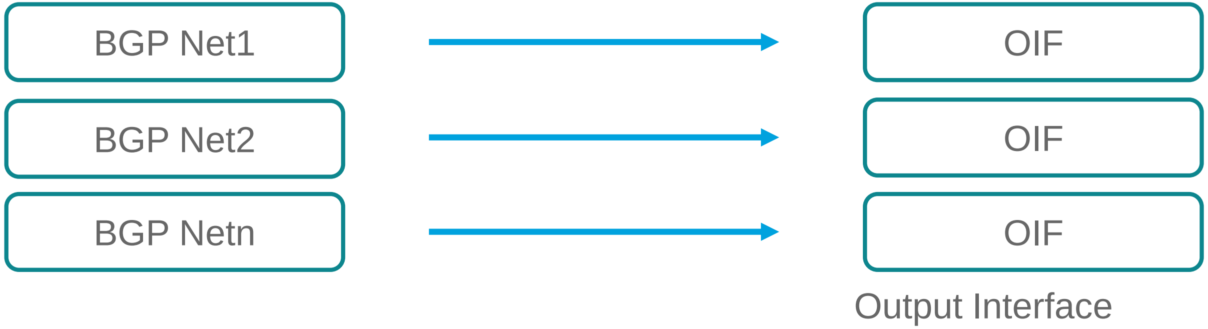


1. **PIC core** – when IGP path changes.
2. **PIC edge** – when remote PE node or it’s reach ability fails.
3. **PIC edge** – when PE-CE link fails.

BGP PIC

Flattened FIB

- With flat FIB, each prefix has its own forwarding information directly associated with an outgoing interface as one-to-one mapping

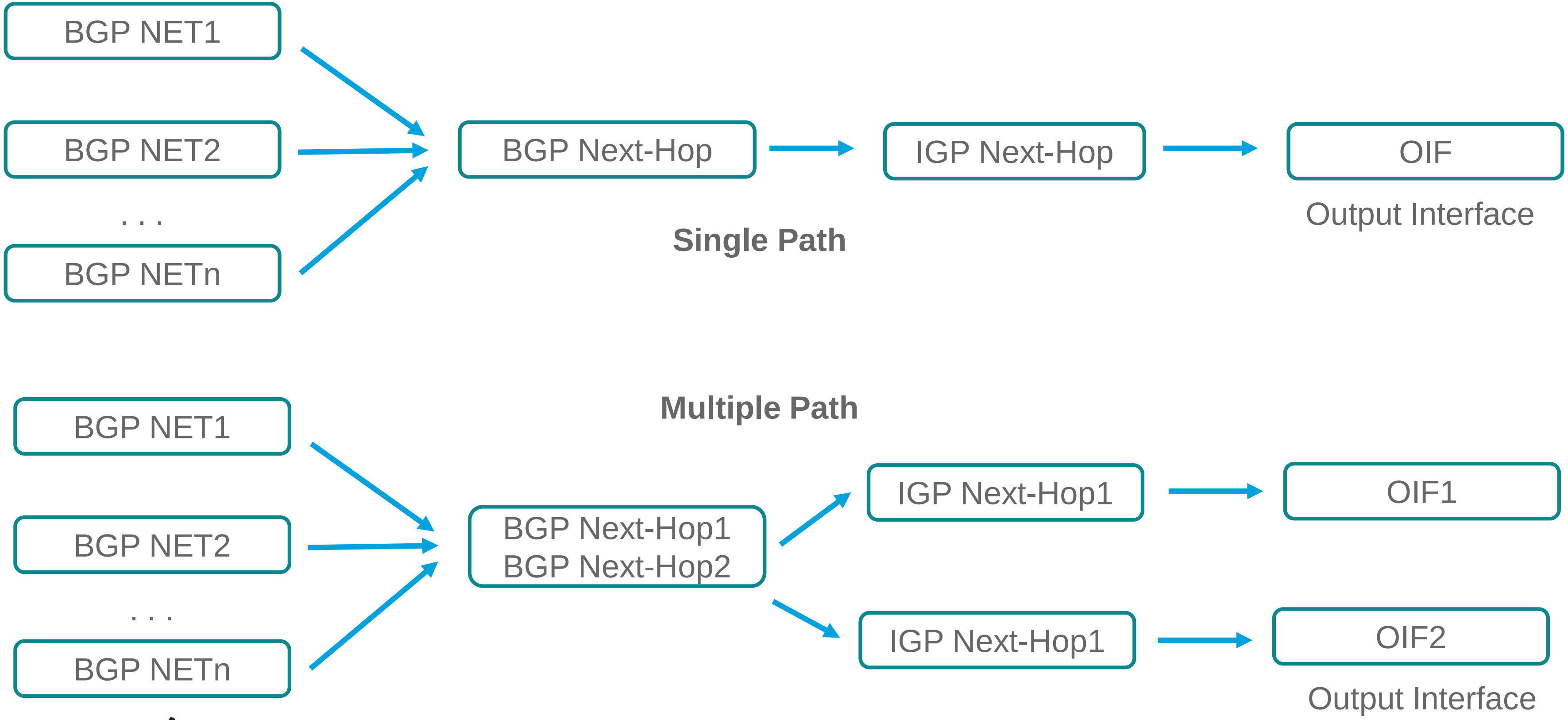


BGP PIC

Hierarchical FIB

- In hierarchical FIB a path-list is assigned to all IGP or BGP prefixes
- IGP prefixes gets path-list of type next-hop, which mean all information is available to select the outgoing interface
- BGP prefixes on the other hand, gets a path-list of type recursive, which points to another path-list type of next-hop
- BGP Core uses hierarchical FIB

Hierarchical FIB



Single Path

Multiple Path

BGP PIC

BGP PIC Core

- BGP PIC core completely depends on how quick the IGP can converge
- Enabled by default no most platforms
- If disabled, use the below command on Cisco IOS to enable BGP PIC Core

```
R1(config)cef table output-chain build favor convergence-speed
```

- No command required on IOS XR or NX-OS, as these platforms work on hierarchical FIB architecture

BGP PIC

BGP PIC Edge

- Can be implemented for PE node protection and for PE-CE Link protection
- To overcome the convergence issues, BGP installs the backup path in the RIB, FIB and LFIB (in case of MPLS VPNs).

Backup path calculation and installation:
bgp additional-paths install on Cisco IOS,

additional-paths selection route-policy *route-policy-name* on IOS XR and NX-OS
additional-paths install on NX-OS.

Best-External knob:
bgp advertise-best-external

BGP Output

```
ASR-1K# sh ip bgp vpnv4 vrf site-111111 2.0.0.0
BGP routing table entry for 65300:111111:2.0.0.0/24, version 150035
Paths: (3 available, best #1, table sie-111111)
   Additional-path-install
   Advertised to update-groups:
    105
Refresh Epoch 1
    20570 20570
    10.200.1.2 from 10.200.1.2 (10.200.1.2)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:64300:111111 , recursive-via-connected
      rx pathid: 0, tx pathid: 0x0
<snip>
  Refresh Epoch 1
  20570 20570
    10.10.10.2 from 10.10.10.2 (5.5.5.5)
      Origin incomplete, localpref 100, valid, internal, backup/repair
      Extended Community: RT:64300:111111 , recursive-via-host
      rx pathid: 0, tx pathid: 0
```

CEF Output

```
ASR-1K# sh ip cef vrf site-111111 2.0.0.0 255.255.255.0 detail

2.0.0.0/24, epoch 0, flags rib only nolabel, rib defined all labels
  recursive via 10.200.1.2
    attached to GigabitEthernet0/1/1.200
  recursive via 10.10.10.2, repair
    attached to GigabitEthernet0/1/0.451
```

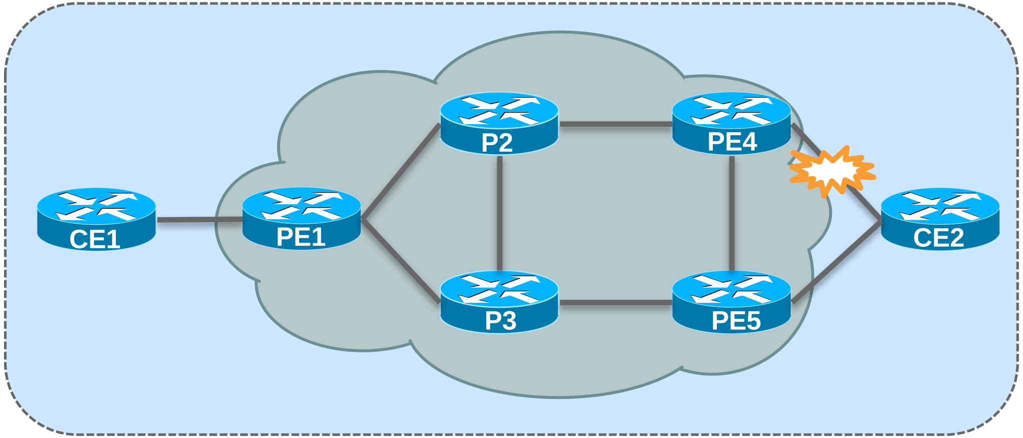
Show Commands

- **BGP**
  - ✓ `sh ip bgp ipv4 unicast <x.x.x.x>`
  - ✓ `sh ip bgp vpnv4 unicast vrf FOO <x.x.x.x>` (To check if backup/repair is set on a prefix)
  - ✓ `sh ip bgp vpnv4 vrf FOO neighbor <neighbor_ip>` (shows if PIC is enabled)
- **RIB**
  - ✓ `Show ip route vrf FOO <x.x.x.x>`
- **CEF**
  - ✓ `show (ip | ipv6) cef [vrf XX] prefix/mask internal`
  - ✓ `show monitor event cef (ipv4 | ipv6 | bfd) all`
  - ✓ `show cef bfd`
  - ✓ `debug cef bfd`
  - ✓ `debug cef loadinfo map`
  - ✓ `debug cef path`
  - ✓ `debug cef filter fib (ipv4 | ipv6) prefix/mask`

TAC Case Example

BGP PIC

- Customer has implemented BGP PIC Edge feature and is also using BFD for faster failover.
- When the best path fails, the customer does not see the fast convergence happening



CEF Output

```
ASR-1K#sh ip cef vrf site-111111 2.0.0.0 internal
2.0.0.0/30, epoch 0, flags rib only nolabel, rib defined all labels, RIB[B], refcount 6, per-
destination sharing
  sources: RIB
<snip>
   GigabitEthernet0/1/1.350(21): 10.200.1.2
  path 7FA7EB87AB00, path list 7FA7EB611F90, share 1/1, type recursive, for IPv4, flags recursive-
via-connected
  recursive via 10.200.1.2[IPv4:sie-111111], fib 7FA7ECBE2558, 1 terminal fib, v4:sie-
111111:10.200.1.2/32
    path 7FA7EB87A630, path list 7FA7EB612C10, share 1/1, type adjacency prefix, for IPv4
    attached to GigabitEthernet0/1/1.350, adjacency IP adj out of GigabitEthernet0/1/1.350, addr
10.200.1.2 7FA7EB643568
  path 7FA7EB879DF0, path list 7FA7EB611F90, share 1/1, type recursive, for IPv4, flags repair,
recursive-via-host, unuseable
  recursive via 10.10.10.2[IPv4:sie-111111], repair, fib 7FA7E2E01068, 1 terminal fib, v4:sie-
111111:10.10.10.2/32
    path 7FA7EB87A000, path list 7FA7EB612490, share 1/1, type adjacency prefix, for IPv4
    attached to GigabitEthernet0/1/0.351, adjacency IP adj out of GigabitEthernet0/1/0.351, addr
10.10.10.2 7FA7EB643B20
<snip>
```

Resolution

```
router bgp 65300
!
address-family ipv4 vrf site-111111
  bgp additional-paths install
  bgp recursion host
  network 3.3.3.0 mask 255.255.255.0
<snip>
  neighbor 10.200.1.2 remote-as 20570
  neighbor 10.200.1.2 version 4
  neighbor 10.200.1.2 fall-over bfd
  neighbor 10.200.1.2 activate
  neighbor 10.200.1.2 send-community
  neighbor 10.200.1.2 next-hop-self
  neighbor 10.200.1.2 soft inbound
exit-address-family
```

Removing this configuration, resolved the issue. BGP Recursion host is useful when performing node protection

Troubleshooting Missing Routes

Scenario 6 – Missing Routes

Problem Description

- Routes advertised were not learnt on peer router
- Symptoms
  - Traffic Loss / No Traffic for the prefix

Missing Routes

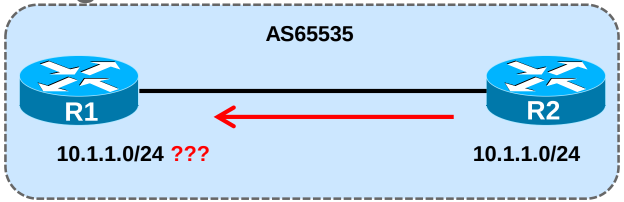
Update Filtering

- Types of filters
  - Prefix filters
  - AS_PATH filters
  - Community filters
  - Route-maps
- Applied in/out direction

Missing Routes

Update Filters

- Determine which filters are applied to the BGP session
  - Show ip bgp nei x.x.x.x
  - Show run | include neighbor x.x.x.x
- Examine the route and pick out the relevant attributes
  - Show ip bgp y.y.y.y
- Compare the attributes against the filters



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

Missing Routes

Community Problems

```
R2#show run | begin bgp
router bgp 2
network 10.1.1.0 mask 255.255.255.0 route-map set-community
...
route-map set-community permit 10
 set community 2:2 1:50
R2#show ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24, 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
```

R1 filtering routes based on community, doesn’t see anything in their BGP table

Missing Routes

Community Problems

```
R2#show run | begin bgp
router bgp 2
network 10.1.1.0 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
```

- Configuration looks Okay – filters okay, route-map okay
- But forgotten “`neighbor 1.1.1.1 send-community`”

Missing Routes

Community Problems

- R2 now advertises prefix with community to R1
- But R1 still doesn’t see the prefix
  - Since nothing is wrong on R2, 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
```

Missing Routes

Community Problems

- Community match on R1 expects 1:150 to be set on prefix
- But R2 is sending 1:50
  Typo or miscommunication between operations?
- R1 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

Debugging with ACL

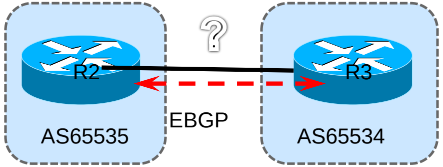
- If unable to find any config issues, try enabling debugs (conditional / filtered debugs)

```
R1#show access-list 99
Standard IP access list 99
    permit 10.1.1.0 0.0.0.255

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

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 10.1.1.0/24 -- DENIED due to: route-map;
```

Troubleshooting Missing Routes

For your reference only

Troubleshooting route advertisement problems



Troubleshooting steps:

- Determine how the prefix is injected into BGP on the advertising router
- Is the prefix in the BGP table? Is the prefix in the routing table?
- Is the prefix being advertised to “ANY” neighbor? Is it the “best” path?
- What update-group is the neighbor in?
- Is the prefix present in the “advertised-routes” output? Any output policy that can block the prefix?
- On the receive side, is there any input policy that can block the route?
- Is the prefix present in the output of “***show ip bgp neighbor <> routes***”?
- Is the prefix present in the BGP table?

What to collect if the issue persists:

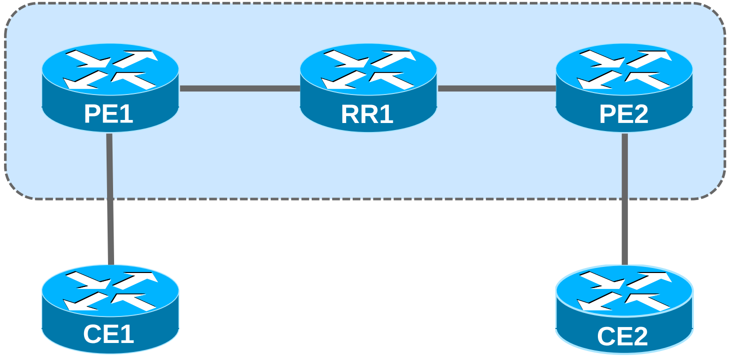
- ***Show tech*** and ***show log*** from both peers
- ***Show ip bgp summary*** from both peers
- ***Show ip bgp w.x.y.z*** for the prefix that is not being advertised/received
- ***Show ip bgp neighbor w.x.y.z advertised-routes***
- ***Show ip bgp neighbor w.x.y.z routes***
- ***Show ip bgp update-group***

BGP for Service Providers

Scenario 7 – MPLS VPN

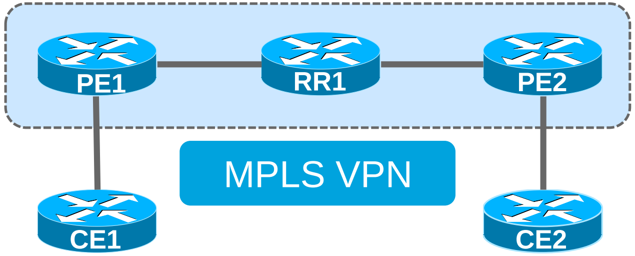
Problem Description

- Customer reports reachability issues for a Customer VRF – Customer A connected to PE1 is unable to reach its other site connected to PE2
- Caused due to:
  - Wrong IGP/VPN Label propagation
- Symptoms
  - No reachability
  - Packet loss

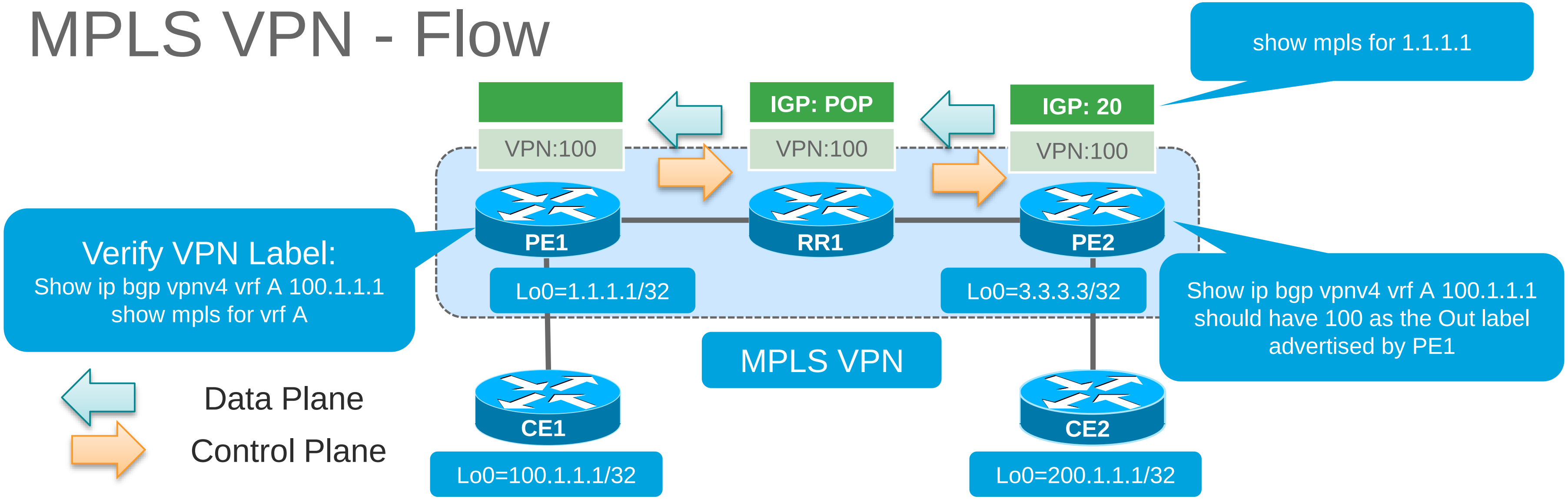


Verifying Configuration & Reachability

- Verify VRF configuration on both PE routers
  - IOS - show run vrf A
- Verify local PE-CE Reachability (PE1 – CE1) & (PE2 – CE2).
- Verify PE to PE loopback reachability
  - PE1# ping <PE2_loopback> source loopback 0
- Verify LSP path between PE routers
  - ping mpls ipv4 <dst> <subnet>
  - ping mpls traffic-eng tunnel <tunnel_num>
- Verify PE to PE reachability for VRF
  - PE1# ping vrf A <vrf_ip_on_PE2>



MPLS VPN - Flow



show mpls for 1.1.1.1

Verify VPN Label:
Show ip bgp vpnv4 vrf A 100.1.1.1
show mpls for vrf A

Show ip bgp vpnv4 vrf A 100.1.1.1 should have 100 as the Out label advertised by PE1

```
PE1#sh ip bgp vpnv4 vrf A 100.1.1.1
BGP routing table entry for 1:1:100.1.1.1/32
192.168.10.2 from 0.0.0.0 (1.1.1.1)
<snip>
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER
ID:192.168.10.1:0
      mpls labels in/out 100/nolabel
```

```
PE2#sh ip bgp vpnv4 vrf A 100.1.1.1
BGP routing table entry for 1:1:100.1.1.1/32
1.1.1.1 (metric 21) from 2.2.2.2 (2.2.2.2)
<snip>
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER
ID:192.168.10.1:0
      Originator: 1.1.1.1, Cluster list: 2.2.2.2
      mpls labels in/out nolabel/100
```

MPLS VPN

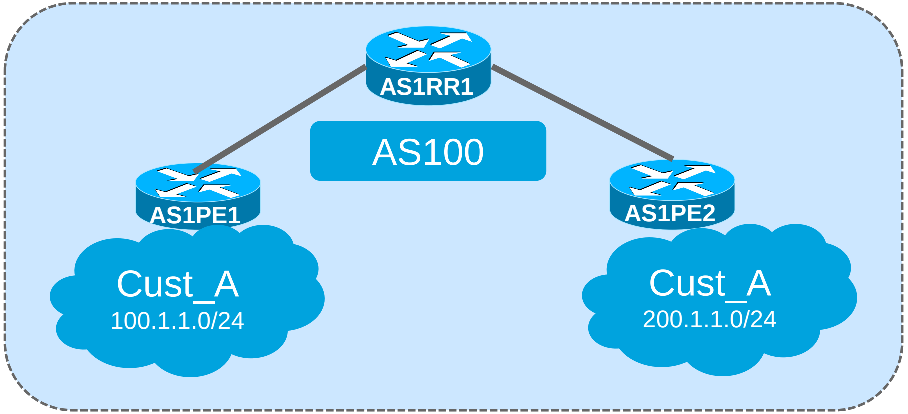
Checking All VPN Labels

```
PE1#sh ip bgp vpnv4 all labels
   Network          Next Hop        In label/Out label
Route Distinguisher: 1:1 (A)
   100.1.1.1/32     192.168.10.2      100/nolabel
   192.168.10.0/30  0.0.0.0           19/nolabel(A)
   192.168.20.0/30  3.3.3.3           nolabel/19
   200.1.1.1/32     3.3.3.3           nolabel/20
PE1#
```

TAC Case Example - 4

Controlled Debugging

- Customer reported outage for their end-customer after the link flap between PE and CE.
- The customer route is not being learnt on the remote PE router, causing loss of reachability between two sites for servers in the subnet



TAC Case Example

Troubleshooting Done

- Debugging done to see if AS1PE1 was sending the update or if AS1RR1 receiving the update

```
debug ip bgp vpnv4 unicast update <neighbor> <acl> in
Access-list 10 per 100.1.1.0 0.0.0.255
```

AS1PE1 (IOS)

```
route-policy DEBUG_BGP
if destination in BGP_PREFIX then
pass
else
drop
endif
end-policy

prefix-set BGP_PREFIX
100.1.1.0/24
end-set

debug bgp update vpnv4 unicast [in | out] route-policy DEBUG_BGP
```

AS1RR1 (XR)

TAC Case Example

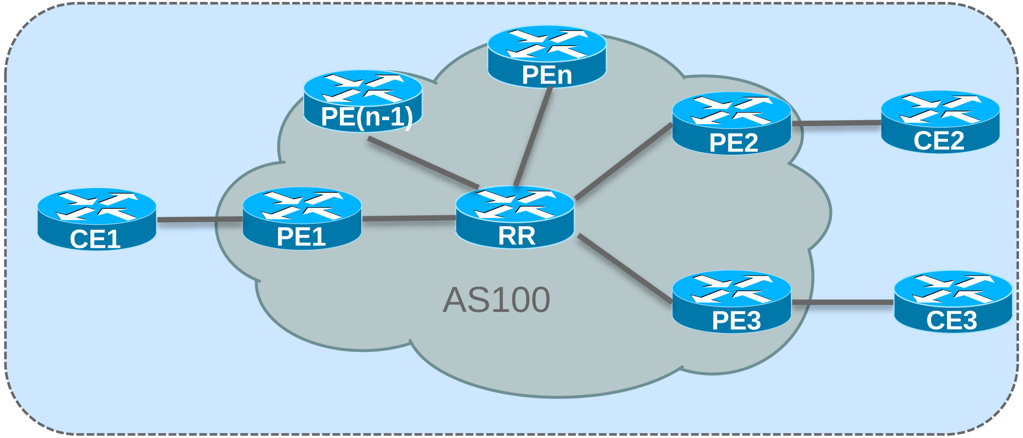
Other Imp Debugs / Traces / show techs

- IOS XR
  - Show bgp trace [update] [error]
  - Show cef trace
  - Show tech routing bgp
- IOS
  - Debug bgp <af> update
  - Debug ip bgp <af> trace
- NXOS
  - Show tech bgp
  - Show tech netstack

Scenario 8 – BGP RT Constraint Filtering

Problem Description

- Customer reported that their resource utilization on edge routers have increased with the growth of their SP network where as no extra services has been deployed on those edge devices
- Symptoms
  - Unwanted resource utilization
  - Performance issues



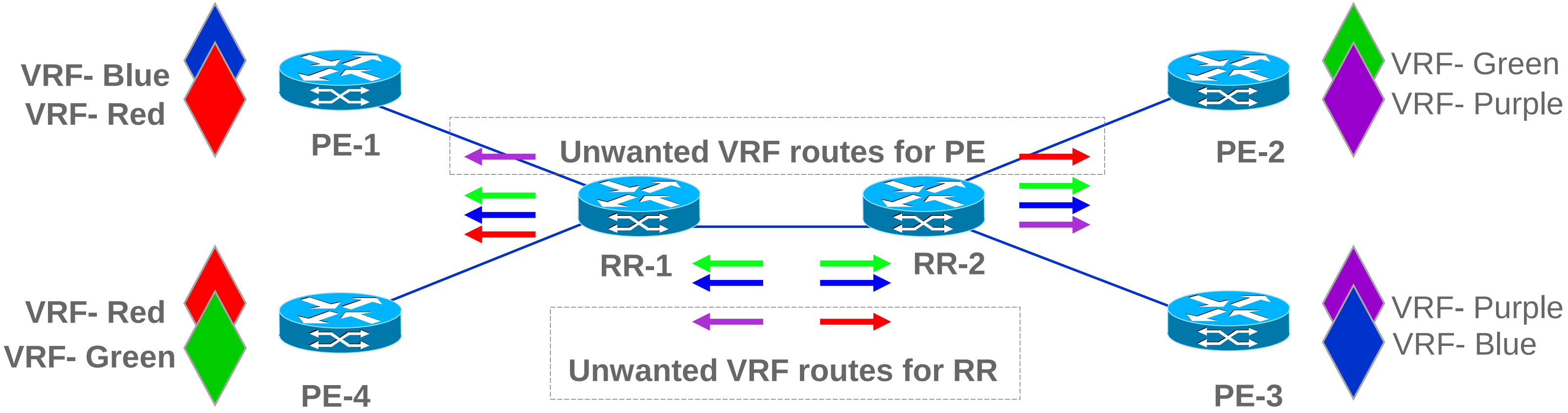
BGP RT Constraint Filtering

Overview

- PE routers use Route Target (RT) extended communities to control the distribution of routes into VRFs.
- Ideally PE routers need to hold only routes marked with Route Targets pertaining to VRFs that have local CE attachments.
- When the distribution of VPNs is sparse, there is wastage of resources in maintaining the unwanted routes at the PE routers and unnecessary distribution of routes by the route-reflectors.
- Route Target (RT)-constraint is a mechanism to prevent the propagation of VPN NLRI to a PE that is not interested in the VPN.

BGP RT Constraint Filtering

Un-wanted Routes at RR & PE



- PE-3 and PE-4 advertise VRF Blue, red and green VPN routes to RR-1
- RR-1 send ALL its VPN routes to RR-2.
- VRF-Red routes are really ‘unwanted’ on RR-2 since PE-1 and PE-2 does not have VRF Purple.

BGP RT Constraint Filtering

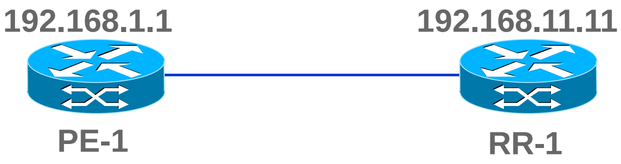
Concept

- By having BGP speakers exchanging the ‘wanted’ Route Targets, this allows BGP speaker to eliminate advertising ‘unwanted’ VPN routes to its peer.
- The ‘wanted’ Route Targets are called RT membership.
- MP-BGP UPDATE message to propagate RT membership information.
  - Peers to advertise their RT membership.
  - Restrict advertisement of VPN route based on received RT membership information.
- Perform Constraint Route Distribution on VPN v4 and v6 Route advertisements only

BGP RT Constraint Filtering

Enable RT Constraint on RR

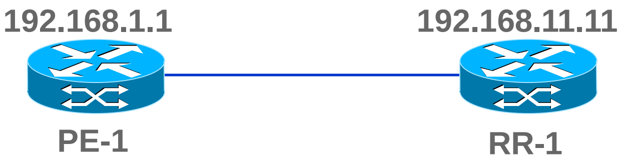
```
RR-1#sh run | b router bgp
router bgp 65000
 address-family vpnv4
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 send-community both
  neighbor 192.168.1.1 route-reflector-client
!
 address-family rtfilter
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 send-community both
  neighbor 192.168.1.1 route-reflector-client
 exit-address-family
```



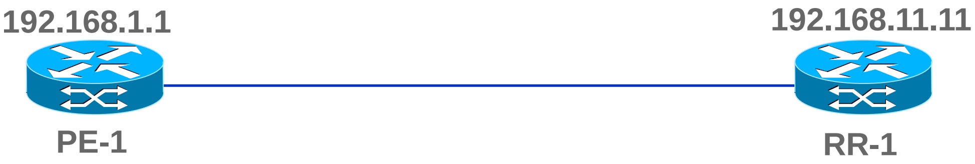
BGP RT Constraint Filtering

Enable RT Constraint on PE

```
PE-2#sh run | b router bgp
router bgp 65000
 address-family vpnv4
  neighbor 192.168.11.11 activate
  neighbor 192.168.11.11 send-community both
exit-address-family
 !
 address-family rtfilter
  neighbor 192.168.11.11 activate
  neighbor 192.168.11.11 send-community both
exit-address-family
```



RT Constraint Capability Exchange

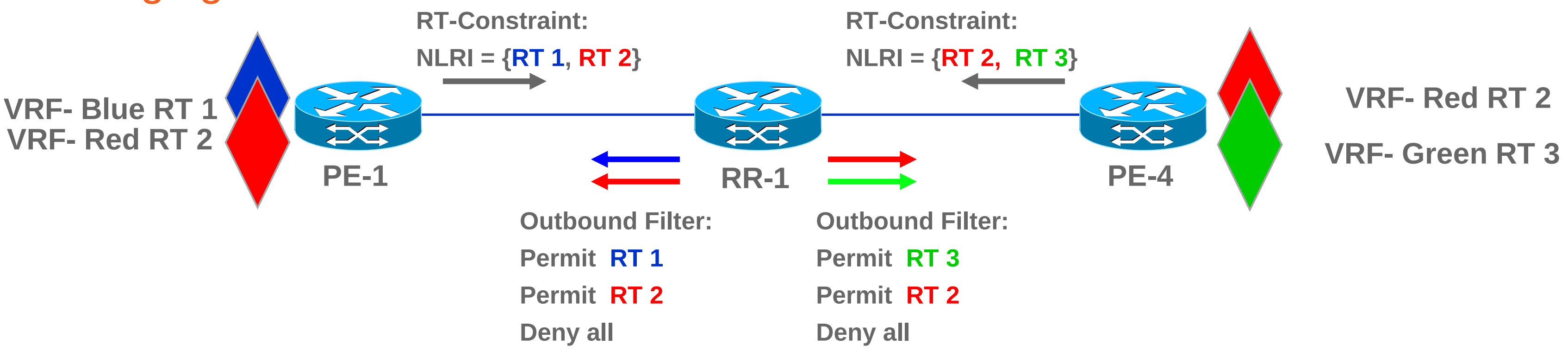


```
01:18:34.658: BGP: 192.168.1.1
   active OPEN has CAPABILITY code:
   1, length 4

01:18:34.658: BGP: 192.168.11.11
   active OPEN has MP_EXT CAP for
   afi/safi: 1/132

01:18:34.658: BGP: 192.168.11.11
   accept RTC SAFI
```

```
01:16:24.896: BGP: 192.168.1.1
  passive OPEN has CAPABILITY code:
  1, length 4

01:16:24.896: BGP: 192.168.1.1
  passive OPEN has MP_EXT CAP for
  afi/safi: 1/132

01:16:24.897: BGP: 192.168.1.1
  accept RTC SAFI
```

BGP RT Constraint Filtering

Exchanging RTC between PE and RR



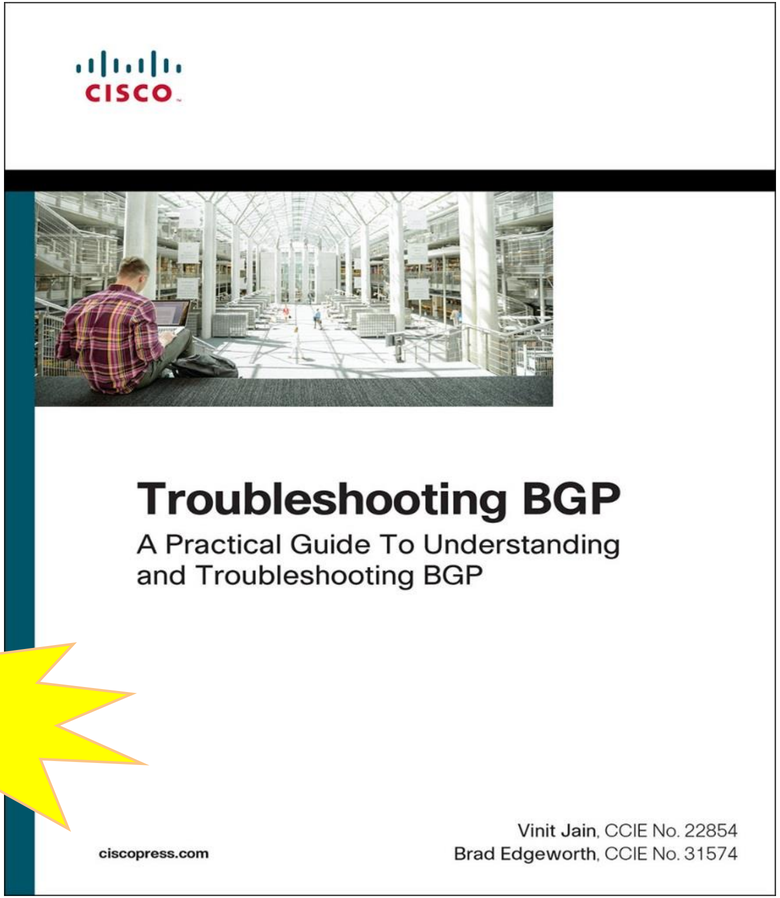
1. PE-1 sends RTC NLRI {RT 1, RT 2} to RR-1
2. PE-4 sends RTC NLRI {RT 2, RT 3} to RR-1
3. RR-1 install an outbound Filter (Permit RT 1, RT 2) for PE-1
4. RR-1 installs an outbound Filter (Permit RT 2, RT 3) for PE-4

Key Takeaways – What have we learned ?

- Key challenges while troubleshooting BGP
- Various tools and techniques
- Real TAC examples
- Best practices in BGP
- Understanding Convergence and troubleshooting Convergence on various Cisco platforms.
- Learnt what impact can a BGP slow peer have
- Troubleshooting BGP in MPLS VPN deployment

Call to Action

- Explore most common BGP problems that you faced in your network
- Baseline your network resources (CPU, Memory, BGP Prefixes, TCAM,…)
- What changed in BGP? (New prefixes, route-maps, filters, peers…)
- Try to narrow down the problem with techniques we discussed
- Collect as much information with available show commands during problematic condition – Helps faster resolution
- Leverage scripting tools for sporadic problems (EEM, TCL,…)
- Enable event tracing – helps in forensic investigation for RCA
- Debug is a last resort. Be cautious and be specific (with filters)



Troubleshooting BGP

A Practical Guide To Understanding and Troubleshooting BGP

Vinit Jain, CCIE No. 22854
Brad Edgeworth, CCIE No. 31574

ciscopress.com

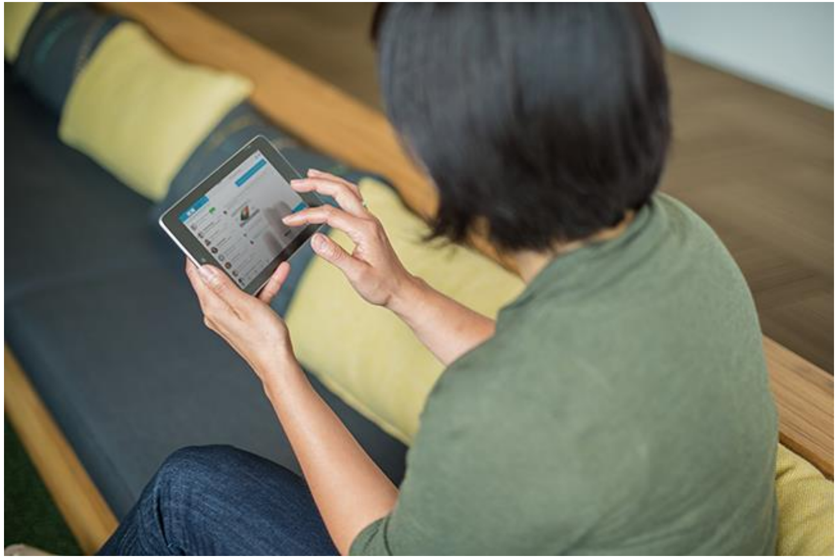
Coming
this year

Recommended Sessions

- Troubleshooting End-to-End MPLS (BRKMPL-3124)
- Troubleshooting VxLAN BGP EVPN (BRKDCN-3040)
- Securing BGP (BRKRST-3179)

Complete Your Online Session Evaluation

- Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 Amazon gift card.
- Complete your session surveys through the Cisco Live mobile app or from the Session Catalog on CiscoLive.com/us.



Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online

Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Table Topics
- Meet the Engineer 1:1 meetings
- Related sessions

Please join us for the Service Provider Innovation Talk featuring:

Yvette Kanouff | Senior Vice President and General Manager, SP Business

Joe Cozzolino | Senior Vice President, Cisco Services

Thursday, July 14th, 2016

11:30 am - 12:30pm, In the Oceanside A room

What to expect from this innovation talk

- Insights on market trends and forecasts
- Preview of key technologies and capabilities
- Innovative demonstrations of the latest and greatest products
- Better understanding of how Cisco can help you succeed

Register to attend the session live now or watch the broadcast on cisco.com

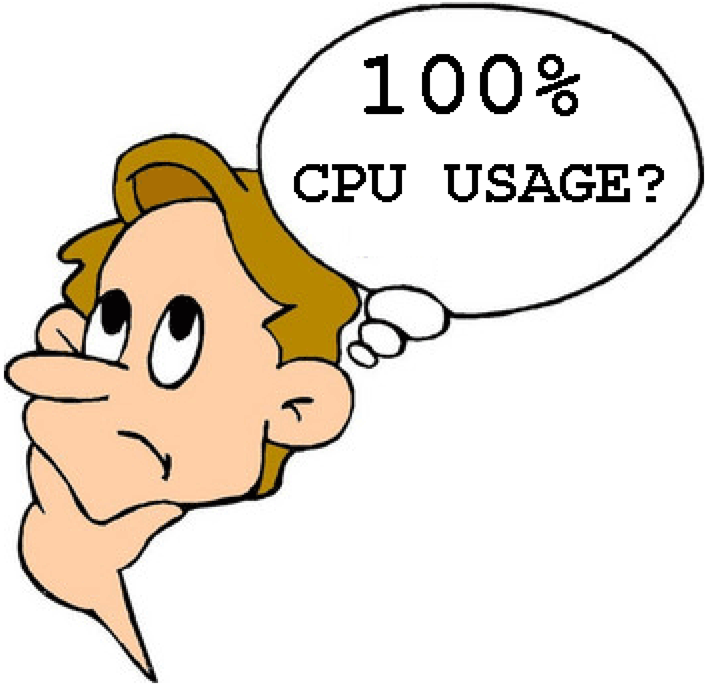
Q & A

Backup Slides

Scenario – High CPU due to BGP

Problem Description

- High CPU noticed due to BGP
  - BGP Scanner
  - BGP Router
- Symptoms
  - Router unstable
  - Traffic loss
  - Loss of Manageability



High CPU – BGP Scanner

- Can be expected for short durations carrying large Internet routing table
- High cpu condition varies with the number of neighbors and the number of routes learned per neighbor
- Verify from platform data-sheet on the scalability of the router
  - E.g. Sup720-3BXL on 6500/7600 series router has default IPv4 TCAM size of 512,000 routes and can be expanded to maximum of 1,000,000 routes
- Make use of **maximum-prefix** knob where required

```
------------------ show process cpu ------------------

CPU utilization for five seconds: 99%/0%; one minute: 28%; five minutes: 17%

 143   393745836   2319493     169755 71.49% 15.34% 12.38%   0 BGP Scanner

. . . .
```

High CPU – BGP Router

```
Router#show process cpu
  CPU utilization for five seconds: 100%/0%; one minute: 99%; five minutes: 81%
....
139     6795740   1020252       6660 88.34% 91.63% 74.01%   0 BGP Router
```

- Look at the scenario
  - Is BGP going through “Initial Convergence”?
- Are there any route churns?
- The high cpu on the device could also be due to the instability of the BGP table. (Receiving two copies of routing table – one from iBGP and one from eBGP)
  - Insufficient Memory

Route Churn (Flapping Routes)

- How to identify route churn?
  - Do “sh ip bgp summary | in table”, note the table version
  - Wait 4-5 seconds
  - Do “sh ip bgp summary | in table”, compare the table version from 4-5 seconds ago
- You have 150k routes and see the table version increase by 300
  - This is probably normal route churn
  - Know how many bestpath changes you normally see per minute
- You have 150k routes and see the table version increase by 150k
  - This is bad and is the cause of your high CPU

Route Churn

```
Router#show ip route | in 00:00:0
B       187.164.0.0 [200/0] via 218.185.80.140, 00:00:00
B       187.52.0.0 [200/0] via 218.185.80.140, 00:00:00
B       187.24.0.0 [200/0] via 218.185.80.140, 00:00:00
B       187.68.0.0 [200/0] via 218.185.80.140, 00:00:00
B       186.136.0.0 [200/0] via 218.185.80.140, 00:00:00
. . . .

Router#Show ip bgp all sum | in tab
BGP table version is 936574954, main routing table version 936574954
BGP table version is 429591477, main routing table version 429591477
Router#


Router#Show ip bgp all sum | in tab
BGP table version is 936576768, main routing table version 936575068
BGP table version is 429591526, main routing table version 429591526
Router#
```

Over 1800 prefixes flapped

Embedded Event Manager (EEM)

- Serves as a powerful tool for high CPU troubleshooting
- Triggered based on event and thresholds
- Multiple actions can be set based on events

```
event manager applet HIGHCPU
event snmp oid "1.3.6.1.4.1.9.9.109.1.1.1.1.3.1" get-type exact entry-op gt entry-val "90"
exit-op lt exit-val "70" poll-interval 5 maxrun 200
action 1.0 syslog msg "START of TAC-EEM: High CPU"
action 1.1 cli command "enable"
action 1.3 cli command "debug netdr clear-capture"
action 1.4 cli command "debug netdr capture“
action 2.0 cli command "sh clock | append disk0:proc_CPU"
action 2.1 cli command "show process cpu sorted | append disk0:proc_CPU“
action 2.2 cli command "show proc cpu history | append disk0:proc_CPU"
action 2.3 cli command "show netdr | append disk0:proc_CPU"
action 3.1 cli command "show log | append disk0:proc_CPU"
action 4.0 syslog msg "END of TAC-EEM: High CPU"
```

Scenario – High Memory / Memory Leak

Problem Description

- High Memory consumption by BGP
- Caused due to:
  - Insufficient Memory
  - Memory Leak
- Symptoms
  - Slow performance
  - Malloc Failures
  - Router reloads

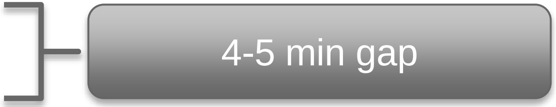
Malloc Failures

```
Sep 20 22:43:01.831 UTC: %SYS-2-MALLOCFAIL: Memory allocation of 65556 bytes
failed from 0x400E04EC, alignment 16
Pool: Processor  Free: 8952  Cause: Not enough free memory
Alternate Pool: Reclaimed  Free: 25520  Cause: Not enough free memory
-Process= "BGP Router", ipl= 0, pid= 156
-Traceback= 40348B24 403FB928 403FDFA0 403F7238 40F5AC5C 4026B690 406B794C 40682DB0
406833FC 40884688 40884DF4 40885BB4 40F9B160 40885C68 40843E68
```

- If Malloc error show the process as BGP – doesn’t mean BGP is the culprit
- This error log can be the consequence of insufficient memory or a memory leak condition
- Get memory base line from your NMS tool
- Run “**show memory debug leak [chunk]**” to identify a memory leak

Memory Leak in IOS XR

- Use IOS XR memory comparator tool to track any incremental memory leak
- Simple 3 step process
  - Show memory compare start
  - Show memory compare end
  - Show memory compare report



```
RP/0/RP0/CPU0:XR_RTR# show memory compare report
Sun Apr 12 22:28:21.715 PDT
JID   name                    mem before   mem after    difference mallocs
restart/exit/new
---   ----                    ----------   ---------    ---------- -------
1088  mibd_interface          232432236    232957764    525528     33753
1086  mibd_entity             1046476      1528332      481856     11
1044  bgp                     1037562644   1037827636   264992     425
0     malloc_dump             0            22144        22144      355
<snip>
```

Memory Baseline

- Memory usage baseline (Use of polling servers)
  - Understand since when the memory started increasing
  - What changes were made around the time the memory increased?
- Understanding if the platform is having sufficient memory based on the services its running
  - Also based on the amount of information learnt through BGP

TAC Case Example

Memory Leak

- Customer reported “show memory” loses about 3-10 Mb of Free memory per day
- It was noticed that BGP Router process accumulated most of the holding memory
- Same behavior was noticed even after the reload.

Allocated	Freed	Holding	Process
458001556	61617656	240877764	BGP Router
459588300	61623540	241919840	BGP Router
463207104	61699296	244022952	BGP Router
467475524	61706168	246534736	BGP Router

The holding memory keeps on increasing

TAC Case Example

Memory Leak

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
80.81.192.33    4  1273       0       0        0    0    0 never    Idle (Admin)
80.81.192.45    4 20646       0       0        0    0    0 never    Idle (Admin)
80.81.192.117   4  3209       0       0        0    0    0 never    Idle (Admin)
80.81.193.61    4  8220       0       0        0    0    0 never    Idle (Admin)
80.81.193.217   4  3209       0       0        0    0    0 never    Idle (Admin)
195.30.0.18     4  5539       0       0        0    0    0 never    Idle (Admin)
. . . . .
```

- Resolution
  - Removing the Idle / Admin down neighbors helped overcome the memory leak

Troubleshooting Summary – Platform issues

- High CPU
  - Verify what is the cause of the high CPU
    - Is it due interrupt or process?
  - Verify if there are route churns happening in your network / partner network
  - Verify if there was any recent changes made in your network (addition of new customer, addition of routes) which triggered the impact in your network
  - Configure EEM with appropriate outputs to gather relevant information in case the high CPU condition is random
- High Memory
  - Verify Memory baseline, changes done
  - How fast is the memory increasing?
  - Trigger of memory increase

Thank you

Cisco live!

July 10-14, 2016 · Las Vegas, NV