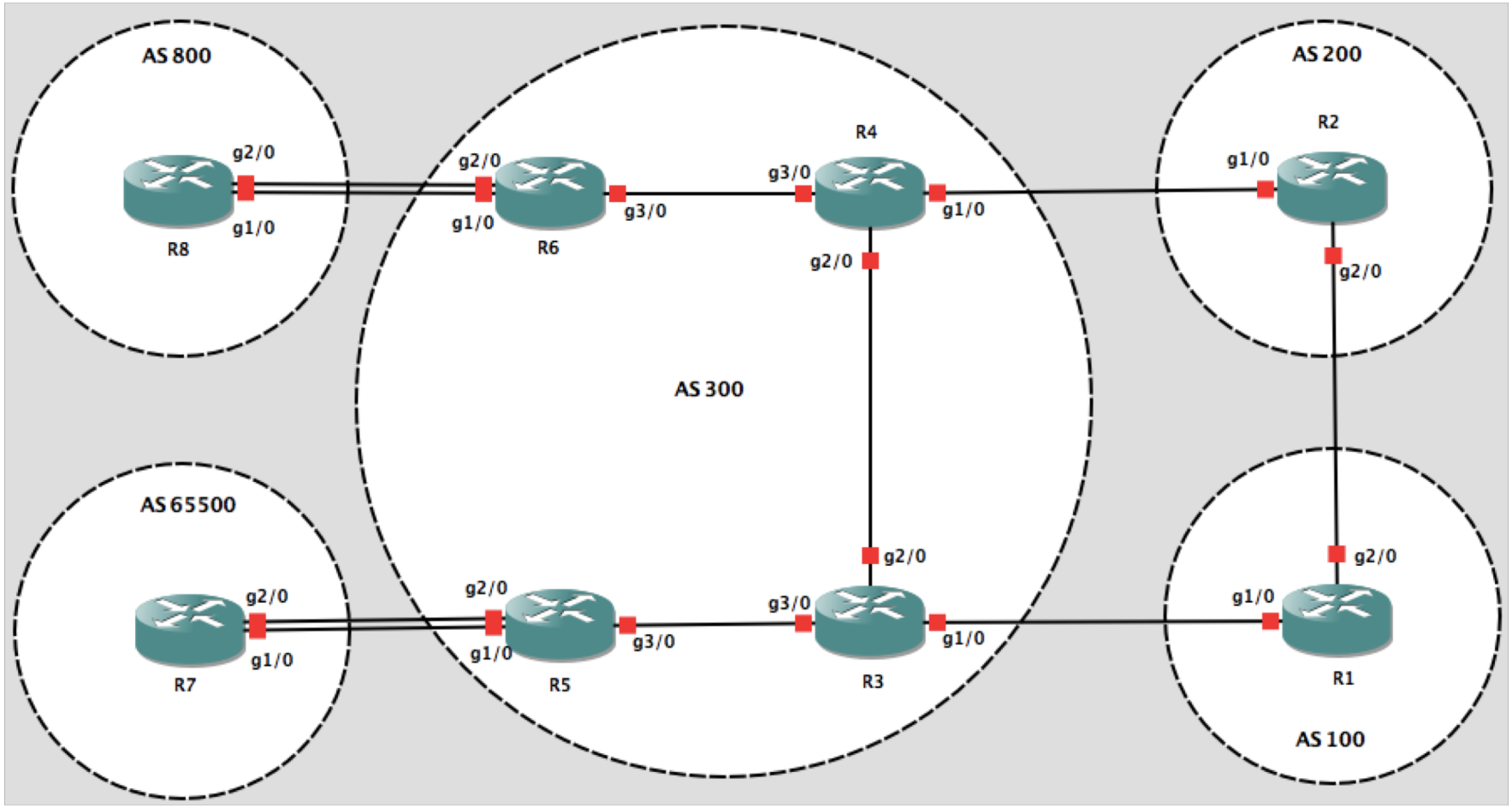
BGP

Lab Activity

Topology


AS 800
AS 200
AS 300
AS 65500
AS 100
R8
R6
R4
R2
R7
R5
R3
R1
g2/0
g1/0
g3/0

Network and IP Plan

- Loopback 10 (R3 – R6): 10.10.10.X/32
- Loopback 10 (R7): 30.30.7.1/32
- Single peering
  - Peering IP 1: 100.100.XY.X(Y)/24
- Dual peering
  - Peering IP 1: 100.100.XY.X(Y)/24
  - Peering IP 2: 100.100.YX.X(Y)/24
- R3 – R6
  - Client prefix: 20.20.X.0/24
  - OSPF Process ID: 1
  - OSPF Router ID: X.X.X.X
  - OSPF Area: 0
- R1, R2, R7 and R8
  - Client prefix: 30.30.X.0/24

Task 0: Troubleshooting Basics

BGP Concept

- Why full mesh network is required for iBGP
- Synchronization rule
- BGP table exchange behavior (best routes)
- iBGP and eBGP route propagation towards neighbor
- iBGP and eBGP next-hop behavior
- Update source for iBGP
- Interpreting BGP table
- Finding the attributes of a route
- Route Reflector
- Route-map operation
- BGP soft and hard reset

Verification

- show ip bgp summary
- show ip bgp neighbors
- show ip protocols
- show ip bgp
- show ip route bgp
- show ip bgp neighbors <neighbor IP> advertised-routes
- show ip bgp neighbors <neighbor IP> routes
- clear ip bgp <AS> / * <soft> <in/out>

Task 1: Basic Configuration

Task 1: Basic Configuration

- Configure all routers
  - Loopback
  - Interface IP

Example: R3

```
interface Loopback 10
 ip address 10.10.10.3 255.255.255.255
!
interface GigabitEthernet1/0
 description Connected to R1 Gi1/0
 ip address 100.100.13.3 255.255.255.0
 no shutdown
!
interface GigabitEthernet2/0
 description Connected to R4 Gi2/0
 ip address 100.100.34.3 255.255.255.0
 no shutdown
!
interface GigabitEthernet3/0
 description Connected to R5 Gi3/0
 ip address 100.100.35.3 255.255.255.0
 no shutdown
```

Task 2: IGP Configuration

Task 2: IGP Configuration

- Configure OSPF in R3 – R6
  - Router ID
  - Interface with OSPF 1 and area 0
- Check OSPF neighbors
- Ping loopback of R6 from R5
  - Check routing table

Example: R3

```
router ospf 1
 router-id 3.3.3.3
 passive-interface gi1/0
!
interface Loopback 10
 ip ospf 1 area 0
!
interface GigabitEthernet2/0
 ip ospf 1 area 0
!
interface GigabitEthernet3/0
 ip ospf 1 area 0
```

Task 3: iBGP Configuration

Task 3: iBGP Configuration

- Configure R3 as iBGP RR and R4-R6 as RR Client
  - Peer-group in R3
  - Update source loopback
  - Next-hop-self
  - Re-check routing table of R3 – R6
  - Check Cluster-ID and Originator-ID

Example: R3

```
router bgp 300
 bgp router-id 3.3.3.3
 address-family ipv4
  neighbor IBGP-PEER peer-group
  neighbor IBGP-PEER remote-as 300
  neighbor IBGP-PEER update-source Loopback10
  neighbor IBGP-PEER route-reflector-client
  neighbor IBGP-PEER next-hop-self
 neighbor 10.10.10.4 peer-group IBGP-PEER
 neighbor 10.10.10.4 description iBGP with R4
 neighbor 10.10.10.5 peer-group IBGP-PEER
 neighbor 10.10.10.5 description iBGP with R5
 neighbor 10.10.10.6 peer-group IBGP-PEER
 neighbor 10.10.10.6 description iBGP with R6
```

Example: R4

```
router bgp 300
 bgp router-id 4.4.4.4
 neighbor 10.10.10.3 remote-as 300
 neighbor 10.10.10.3 description iBGP with R3
 neighbor 10.10.10.3 update-source Loopback10
 neighbor 10.10.10.3 next-hop-self
```

Task 4: eBGP Configuration

Task 4: eBGP Configuration

- Configure all EBGP peering
  - Announce Network
  - Check eBGP neighbor
  - Check routing table
  - Check BGP Table

Example: R1

```
router bgp 100
 bgp router-id 1.1.1.1
 neighbor 100.100.12.2 remote-as 200
 neighbor 100.100.12.2 description eBGP with R2
 neighbor 100.100.13.3 remote-as 300
 neighbor 100.100.13.3 description eBGP with R3
```

Task 5: Multiple eBGP Peering

Task 5: Multiple eBGP Peering

- Configure eBGP in loopback between R5 and R7
  - One eBGP peering using multihop option
  - Static route to reach the loopbacks
  - Update source
  - Check BGP neighbors
- Configure eBGP between R6 and R8
  - Two eBGP peering for two physical link
  - Check BGP neighbors

Example: R5

```
router bgp 300
 bgp router-id 5.5.5.5
 neighbor 30.30.7.1 remote-as 65500
 neighbor 30.30.7.1 description eBGP with R7
 neighbor 30.30.7.1 ebgp-multihop 2
 neighbor 30.30.7.1 update-source Loopback10
!
ip route 30.30.7.1 255.255.255.255 100.100.57.7
ip route 30.30.7.1 255.255.255.255 100.100.75.7
```

Example: R6

```
router bgp 300
 bgp router-id 6.6.6.6
 neighbor 100.100.68.8 remote-as 800
 neighbor 100.100.68.8 description eBGP-Pri with R8
 neighbor 100.100.86.8 remote-as 800
 neighbor 100.100.86.8 description eBGP-Sec with R8
```

Task 6: Originate Prefix

Task 6: Originate Prefix

- Originate prefix in R3-R6
  - **`network`** command
  - Pull up route

Example: R4

```
router bgp 300
 address-family ipv4
 network 20.20.4.0 mask 255.255.255.0
!
ip route 20.20.4.0 255.255.255.0 null 0
```

Task 7: BGP Authentication

Task 7: BGP Authentication

- Configure MD5 between R1 and R2
  - Configure R1 and wait for message/log
  - Configure different password in R2
  - Fix the problem with correct password
- Check bgp table

Example: R1

```
router bgp 100
 address-family ipv4
  neighbor 100.100.12.2 password BGPLAB
```

Task 8: Remove Private AS

Task 8: Remove Private AS

- Check bgp table from R1 for R7's loopback
  - Is the private AS present in the AS path?
- Remove private AS in R3 towards R1
  - Configure remove-private-as in bgp with R1
  - Soft clear bgp table
- Re-check R1's bgp table for private AS

Example: R3

```
router bgp 300
 address-family ipv4
  neighbor 100.100.13.1 remove-private-as
```

Task 9: Default Route

Task 9: Default Route

- Configure default route in R1 towards R3
  - Soft clear bgp neighbor
  - Check bgp table from R1, R2, R3 and R7

Example: R1

```
router bgp 100
 address-family ipv4
  neighbor 100.100.13.3 default-originate
```

Task 10: Summary Route

Task 10: Summary Route

- Configure summarization in R3 for 20.20.0.0/16
  - Aggregate address in R3 without and with summary-only
  - Check bgp and routing table from R1
  - Aggregate address in R3 without and with as-set
  - Check bgp and routing table from R1 and R4
- Remove the summary route configuration
- http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5441-aggregation.html

Example: R3

```
router bgp 300
 address-family ipv4
  aggregate-address 20.20.0.0 255.255.0.0 [summary-only]
  aggregate-address 30.30.0.0 255.255.0.0 as-set
```

Policy Control: Route-map

- A route-map is like a “programme” for IOS
- Has “line” numbers, like programmes
- Each line is a separate condition/action
- Concept is basically:

  if *match* then do *expression* and exit

  else

  if *match* then do *expression* and exit

  else etc
- Route-map “continue” lets ISPs apply multiple conditions and actions in one route-map

Policy Control: Route-map

- Lines can have multiple set statements
- Lines can have multiple match statements
- Line with only a match statement
  - Only prefixes matching go through, the rest are dropped
- Line with only a set statement
  - All prefixes are matched and set
  - Any following lines are ignored
- Line with a match/set statement and no following lines
  - Only prefixes matching are set, the rest are dropped

Policy Control: Route-map

- Lines can have multiple set statements
- Lines can have multiple match statements
- Line with only a match statement
  - Only prefixes matching go through, the rest are dropped
- Line with only a set statement
  - All prefixes are matched and set
  - Any following lines are ignored
- Line with a match/set statement and no following lines
  - Only prefixes matching are set, the rest are dropped

Policy Control: Route-map

- Example
  - Omitting the third line below means that prefixes not matching list-one or list-two are dropped

```
route-map sample permit 10
 match ip address prefix-list list-one
 set local-preference 120
!
route-map sample permit 20
 match ip address prefix-list list-two
 set local-preference 80
!
route-map sample permit 30 ! Don't forget this
```

Task 11: Weight

Task 11: Weight

- Option 1: Configure weight in BGP peering
  - Check the BGP table in R6 and R8
  - Check the next-hop for R8's route from R6
  - Configure weight for both link so that secondary link becomes best route to R8 from R6
  - Soft clear bgp
  - Check BGP table and routing table for best path
  - Check weight attribute from R6
  - Check weight attribute from R8
  - Revert to previous configuration
- Option 2: Configure weight using route-map

Example: R6

```
router bgp 300
 address-family ipv4
  neighbor 100.100.68.8 weight 100
  neighbor 100.100.86.8 weight 200

clear ip bgp <neighbor IP>/* in/out/soft
```

Task 12: Local Preference

Task 12: Local Preference

- Option 1: Configure default local preference

  `bgp default local-preference` *`[value]`*
- Option 2: Configure local preference in route-map
- Configure local preference in R3 so that R1 becomes next hop for R2's prefix
  - Check BGP table for R2's prefix in R1
  - Configure prefix-list
  - Configure route-map
  - Match prefix-list
  - Set higher local-preference
  - Assign the route-map on neighbor command with R1
  - Soft clear bgp table
- Re-check the next hop
- Remove the local preference configuration

Example: R3

```
ip prefix-list LPREF seq 10 permit 30.30.2.0/24

route-map UPLOAD permit 10
 match ip address prefix-list LPREF
 set local-preference 200
!
route-map UPLOAD permit 20

router bgp 300
 address-family ipv4
  neighbor 100.100.13.1 route-map UPLOAD in
```

Task 13: Multi Exit Discriminator

Task 13: Multi Exit Discriminator

- Option 1: Configure default metric

  `default metric` *`[value]`*
- Option 2: Configure metric in route-map
- Configure MED so that secondary path wins to reach AS300's prefixes from R8 and primary path wins for other prefixes.
  - Check bgp table of R8 and R6
  - Configure prefix-list
  - Configure route-map to set MED
  - Assign route-map to neighbor
  - Soft clear bgp
  - Check metric in R8, R6, R3 and R1
- Re-check the bgp table
- Remove the MED configuration

Example: R8

```
ip prefix-list MED seq 10 permit 20.20.3.0/24
ip prefix-list MED seq 15 permit 20.20.4.0/24
ip prefix-list MED seq 20 permit 20.20.5.0/24
ip prefix-list MED seq 25 permit 20.20.6.0/24

route-map MED-PRI permit 10
 match ip address prefix-list MED
 set metric 100
route-map MED-PRI permit 20
 set metric 200

DO SIMILAR CONFIGS FOR THE BACKUP PATH

router bgp 800
 address-family ipv4
  neighbor 100.100.86.6 route-map MED-PRI out
  neighbor 100.100.68.6 route-map MED-SEC out
```

Task 14: AS Path Prepend

Task 14: AS Path Prepend

- Check the bgp table in R1 for R5's prefix
- Configure as path prepend to make R2 as the next hop from R1 to reach R5
  - Configure prefix list
  - Configure route-map
  - Match prefix list
  - Set as path prepending
  - Assign in bgp neighbor
  - Soft clear bgp
- Recheck bgp table of R1
- Remove the AS path prepend configuration

Example: R3

```
ip prefix-list AS-PATH seq 10 permit 20.20.5.0/24

route-map AS-PREP permit 10
 match ip address prefix-list AS-PATH
 set as-path prepend 300 300
!
route-map AS-PREP permit 20

router bgp 300
 address-family ipv4
  neighbor 100.100.13.1 route-map AS-PREP out
```

Task 15: Prefix List

Prefix-list: Command Syntax

- Syntax:

  `[no] ip prefix-list list-name [seq seq-value] permit|deny network/len [ge ge-value] [le le-value]`

  `network/len:` The prefix and its length

  `ge ge-value:` "greater than or equal to"

  `le le-value:` "less than or equal to"

- Both "ge" and "le" are optional
  - Used to specify the range of the prefix length to be matched for prefixes that are more specific than network/len
- Sequence number is also optional
  - **`no ip prefix-list sequence-number`** to disable display of sequence numbers

Prefix-list: Examples

- Deny default route

  `ip prefix-list EG deny 0.0.0.0/0`

- Permit the prefix 35.0.0.0/8

  `ip prefix-list EG permit 35.0.0.0/8`

- Deny the prefix 172.16.0.0/12

  `ip prefix-list EG deny 172.16.0.0/12`

- In 192/8 allow up to /24

  `ip prefix-list EG permit 192.0.0.0/8 le 24`

  - This allows all prefix sizes in the 192.0.0.0/8 address block, apart from /25, /26, /27, /28, /29, /30, /31 and /32.

Prefix-list: Examples

- In 192/8 deny /25 and above

  `ip prefix-list EG deny 192.0.0.0/8 ge 25`

  - This denies all prefix sizes /25, /26, /27, /28, /29, /30, /31 and /32 in the address block 192.0.0.0/8.
  - It has the same effect as the previous example

- In 193/8 permit prefixes between /12 and /20

  `ip prefix-list EG permit 193.0.0.0/8 ge 12 le 20`

  - This denies all prefix sizes /8, /9, /10, /11, /21, /22, … and higher in the address block 193.0.0.0/8.

- Permit all prefixes

  `ip prefix-list EG permit 0.0.0.0/0 le 32`

  - 0.0.0.0 matches all possible addresses, "0 le 32" matches all possible prefix lengths

Task 15: Prefix List

- Check R4's advertised routes towards R2
- Check R2's received routes from R4
- Configure prefix-list in R4 to send only prefixes of AS 300 to R2
  - Create prefix-list and permit the prefix
  - Use no le/ge logic i.e 20.20.0.0/16
  - Re-check advertise (from R4)/received (from R2) routes
  - Permit 20.20.0.0/16 le 24
  - Re-check advertise (from R4)/received (from R2) routes
- Remove the prefix list configuration

Example: R4

```
ip prefix-list PREFIX seq 10 permit 20.20.0.0/16 le 24

router bgp 300
 address-family ipv4
  neighbor 100.100.24.2 prefix-list PREFIX out
```

Task 16: Filter List

Filter-list: Regular Expression

- Like Unix regular expressions
  - . Match one character
  - * Match any number of preceding expression
  - + Match at least one of preceding expression
  - ^ Beginning of line
  - $ End of line
  - \ Escape a regular expression character
  - _ Beginning, end, white-space, brace
  - | Or
  - () brackets to contain expression
  - [] brackets to contain number ranges

Regular Expression: Examples

- Simple Examples

.*	match anything
.+	match at least one character
^$	match routes local to this AS
_1800$	originated by AS1800
^1800_	received from AS1800
_1800_	via AS1800
_790_1800_	via AS1800 and AS790
_(1800_)+	multiple AS1800 in sequence (used to match AS-PATH prepends)
_\(65530\)_	via AS65530 (confederations)

Regular Expression: Examples

- Not so simple Examples

^[0-9]+$	Match AS_PATH length of one
^[0-9]+_[0-9]+$	Match AS_PATH length of two
^[0-9]*_[0-9]+$	Match AS_PATH length of one or two
^[0-9]*_[0-9]*$	Match AS_PATH length of one or two (will also match zero)
^[0-9]+_[0-9]+_[0-9]+$	Match AS_PATH length of three
_(701|1800)_	Match anything which has gone through AS701 or AS1800
_1849(_.+_)12163$	Match anything of origin AS12163 and passed through AS1849

Task 16: Filter List

- Check R7's received routes from AS 300
- Create filter list in R7
  - Permit prefixes generated from AS 300 only
  - Apply the filter list to bgp neighbor
  - Soft clear bgp
  - Re-check R5's advertised routes and R7's received routes
  - Remove the filter-list command
- Create route map and match the filter list
  - Apply the route map to bgp neighbor
  - Soft clear bgp
  - Re-check R5's advertised routes and R7's received routes

Example: R7

```
ip as-path access-list 10 permit ^300$

router bgp 65500
 address-family ipv4
  neighbor 10.10.10.5 filter-list 10 in
```

OR

```
route-map AS-SET permit 10
 match as-path 10
route-map AS-SET permit 20

router bgp 65500
 address-family ipv4
  neighbor 10.10.10.5 route-map AS-SET in
```

Task 17: BGP Community

Task 17: BGP Community

- In R3
  - Set BGP Community 300:11 for default route received from R1
  - Set BGP Community 300:12 for the rest of the routes
- In R4
  - Set BGP Community 300:2 for the routes received from R2
- In R5
  - Set BGP Community 300:7 for the routes received from R7
- In R6
  - Permit default route towards R8 via secondary path using community
  - Permit full routes towards R8 via primary path using community

Example: R3

```
ip bgp-community new-format

ip prefix-list DEFAULT seq 10 permit 0.0.0.0/0

route-map BGP-COMM-IN permit 10
 match ip address prefix-list DEFAULT
 set community 300:11 additive
route-map BGP-COMM-IN permit 20
 set community 300:12 additive

router bgp 300
 address-family ipv4
  neighbor 100.100.13.1 route-map BGP-COMM-IN in
  neighbor IBGP-PEER send-community
```

Example: R4

```
ip bgp-community new-format

route-map BGP-COMM-IN permit 10
 set community 300:2 additive

router bgp 300
 address-family ipv4
  neighbor 100.100.24.2 route-map BGP-COMM-IN in
  neighbor 10.10.10.3 send-community
```

Example: R6

```
ip bgp-community new-format

ip community-list 1 permit 300:11

ip community-list 2 permit 300:11
ip community-list 2 permit 300:12
ip community-list 2 permit 300:2
ip community-list 2 permit 300:7

route-map BGP-COMM-PRI-OUT permit 10
 match community 2

route-map BGP-COMM-SEC-OUT permit 10
 match community 1

router bgp 300
 address-family ipv4
  neighbor 100.100.68.8 route-map BGP-COMM-PRI-OUT out
  neighbor 100.100.86.8 route-map BGP-COMM-SEC-OUT out
```

Question?