BGP Scaling Techniques

ISP Workshops

Last updated 30th December 2017

Acknowledgements

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

Philip Smith

BGP Scaling Techniques

- Original BGP specification and implementation was fine for the Internet of the early 1990s
  - But didn't scale
- Issues as the Internet grew included:
  - Scaling the iBGP mesh beyond a few peers?
  - Implement new policy without causing flaps and route churning?
  - Keep the network stable, scalable, as well as simple?

BGP Scaling Techniques

- Current Best Practice Scaling Techniques
  - Route Refresh
  - Cisco's Peer-groups
  - Route Reflectors (and Confederations)
- Deprecated Scaling Techniques
  - Soft Reconfiguration
  - Route Flap Damping

Dynamic Reconfiguration

Non-destructive policy changes

Route Refresh

- BGP peer reset required after every policy change
  - Because the router does not store prefixes which are rejected by policy
- Hard BGP peer reset:
  - Tears down BGP peering & consumes CPU
  - Severely disrupts connectivity for all networks
- Soft BGP peer reset (or Route Refresh):
  - BGP peering remains active
  - Impacts only those prefixes affected by policy change

Route Refresh Capability

- Facilitates non-disruptive policy changes
- No configuration is needed
  - Automatically negotiated at peer establishment
- No additional memory is used
- Requires peering routers to support "route refresh capability" – RFC2918
- Tell peer to resend full BGP announcement

```
clear ip bgp x.x.x.x [soft] in
```

- Resend full BGP announcement to peer

```
clear ip bgp x.x.x.x [soft] out
```

Dynamic Reconfiguration

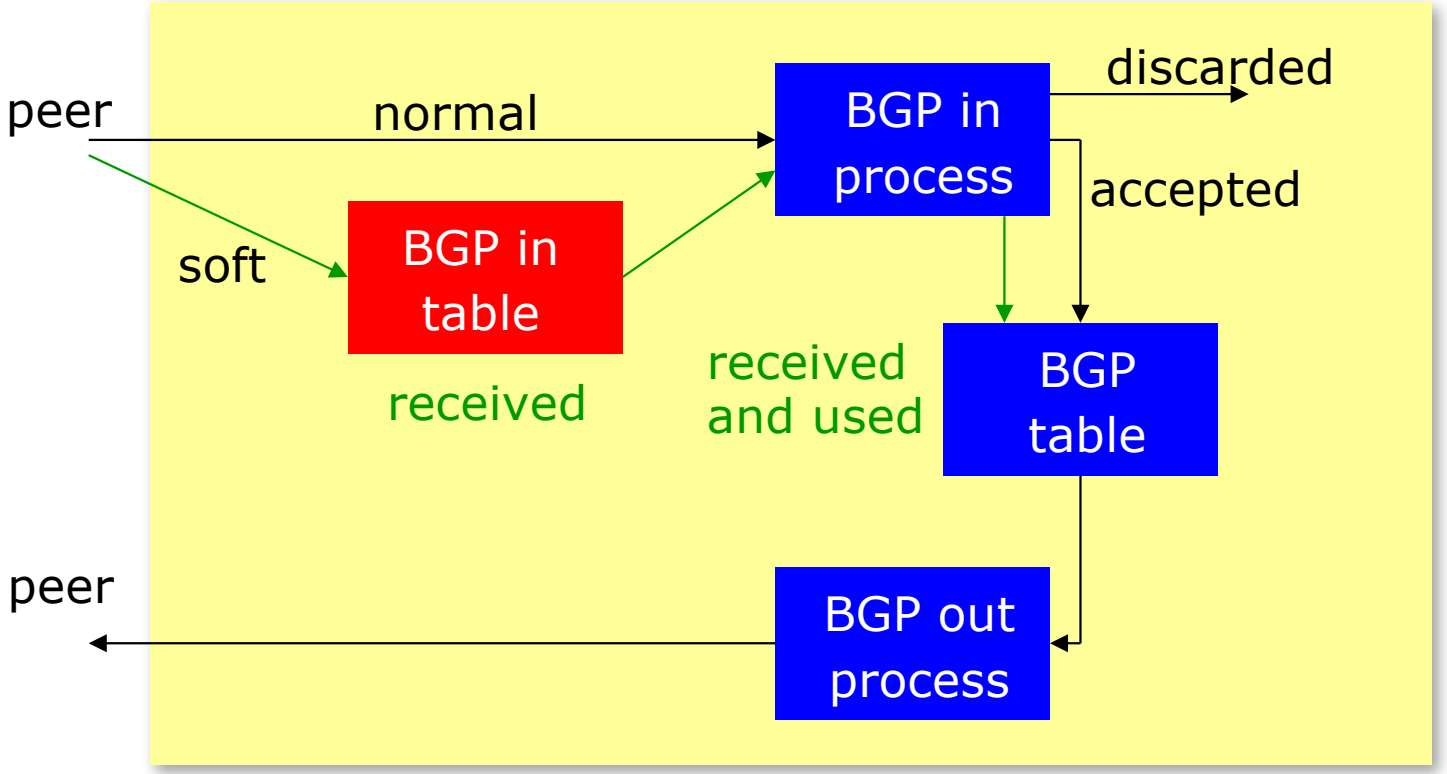
- Use Route Refresh capability
  - Supported on virtually all routers
  - Find out from “show ip bgp neighbor”
  - Non-disruptive, “Good For the Internet”
- Only hard-reset a BGP peering as a last resort

Consider the impact to be equivalent to a router reboot

Cisco's Soft Reconfiguration

- Now deprecated — but:
- Router normally stores prefixes which have been received from peer after policy application
  - Enabling soft-reconfiguration means router also stores prefixes/attributes received prior to any policy application
  - Uses more memory to keep prefixes whose attributes have been changed or have not been accepted
- Only useful now when operator requires to know which prefixes have been sent to a router prior to the application of any inbound policy

Cisco's Soft Reconfiguration



Configuring Soft Reconfiguration

```
router bgp 100
 address-family ipv4
  neighbor 1.1.1.1 remote-as 101
  neighbor 1.1.1.1 route-map infilter in
  neighbor 1.1.1.1 soft-reconfiguration inbound
! Outbound does not need to be configured !
```

- Then when we change the policy, we issue an exec command

```
clear ip bgp 1.1.1.1 soft [in | out]
```

- Note:
  - When “soft reconfiguration” is enabled, there is no access to the route refresh capability

    ```
    clear ip bgp 1.1.1.1 [in | out]
    ```

  - will also do a soft refresh

Cisco's Peer Groups

Cisco's Peer Groups

- Problem – how to scale iBGP
  - Large iBGP mesh slow to build
  - iBGP neighbours receive the same update
  - Router CPU wasted on repeat calculations
- Solution – peer-groups
  - Group peers with the same outbound policy
  - Updates are generated once per group

Cisco's Peer Groups

- Advantages today:
  - Makes configuration easier
  - Makes configuration less prone to error
  - Makes configuration more readable
  - Members can have different inbound policy
  - Can be used for eBGP neighbours too!
- Initial advantages:
  - Lower router CPU load
  - iBGP mesh builds more quickly
  - (Cisco's *update-groups* now provide this)

Configuring a Peer Group

```
router bgp 100
 address-family ipv4
  neighbor ibgp-peer peer-group
  neighbor ibgp-peer remote-as 100
  neighbor ibgp-peer update-source loopback 0
  neighbor ibgp-peer send-community
  neighbor ibgp-peer route-map outfilter out
  neighbor 10.0.0.1 peer-group ibgp-peer
  neighbor 10.0.0.2 peer-group ibgp-peer
  neighbor 10.0.0.2 route-map  infilter in
  neighbor 10.0.0.3 peer-group ibgp-peer
!
```

- Note how 10.0.0.2 has an additional inbound filter over the peer-group

Configuring a Peer Group

```
router bgp 100
 address-family ipv4
  neighbor external-peer peer-group
  neighbor external-peer send-community
  neighbor external-peer route-map set-metric out
  neighbor 160.89.1.2 remote-as 200
  neighbor 160.89.1.2 peer-group external-peer
  neighbor 160.89.1.4 remote-as 300
  neighbor 160.89.1.4 peer-group external-peer
  neighbor 160.89.1.6 remote-as 400
  neighbor 160.89.1.6 peer-group external-peer
  neighbor 160.89.1.6 filter-list infilter in
!
```

- Can be used for eBGP as well

Peer Groups

- Peer-groups are considered obsolete by Cisco:
  - Replaced by update-groups (internal coding – not configurable)
- But are still considered best practice by many network operators
- Cisco introduced peer-templates
  - A much enhanced version of peer-groups, allowing more complex constructs

Cisco's update-groups (1)

- Update-groups is an internal IOS coding, taking over the performance gains introduce by peer-groups

```
Router1#sh ip bgp 10.0.0.0/26
BGP routing table entry for 10.0.0.0/26, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.0.15.241)
      Origin IGP, metric 0, localpref 100, weight 32768, valid...
```

- The "show" command indicates the prefix is handled by update-group #1

Cisco's update-groups (2)

- The update group itself lists all the peers which get the same (identical) update:

```
Router1#sh ip bgp update-group 1
BGP version 4 update-group 1, internal, Address Family: IPv4 Unicast
  BGP Update version : 16/0, messages 0
  Topology: global, highest version: 16, tail marker: 16
  Format state: Current working (OK, last not in list)
                Refresh blocked (not in list, last not in list)
  Update messages formatted 11, replicated 13, current 0, refresh 0, limit 1000
  Number of NLRIs in the update sent: max 2, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 13 members:
   10.0.15.242      10.0.15.243      10.0.15.244      10.0.15.245
   10.0.15.246      10.0.15.247      10.0.15.248      10.0.15.249
   10.0.15.250      10.0.15.251      10.0.15.252      10.0.15.253
   10.0.15.254
```

- And this group has 13 members

Peer Groups

- Always configure peer-groups for iBGP
  - Even if there are only a few iBGP peers
  - Easier to scale network in the future
  - Makes configuration easier to read
- Consider using peer-groups for eBGP
  - Especially useful for multiple BGP customers using same AS (RFC2270)
  - Also useful at Exchange Points:
    - Where ISP policy is generally the same to each peer
    - For Route Server where all peers receive the same routing updates

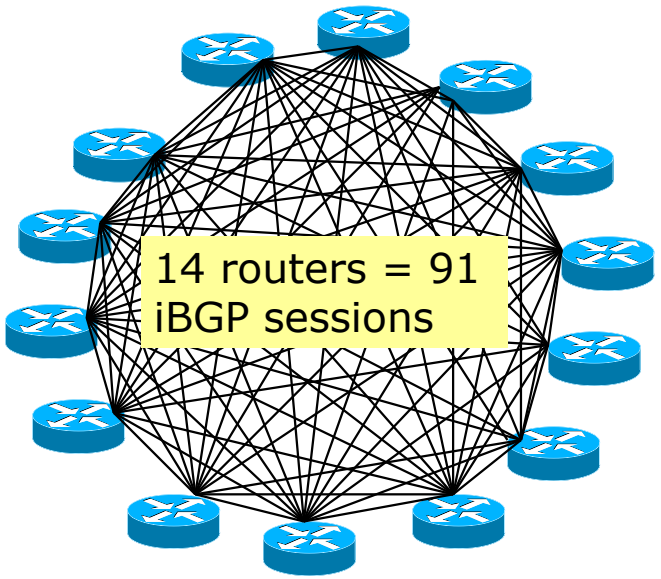
Route Reflectors

Scaling the iBGP mesh

Scaling the iBGP mesh

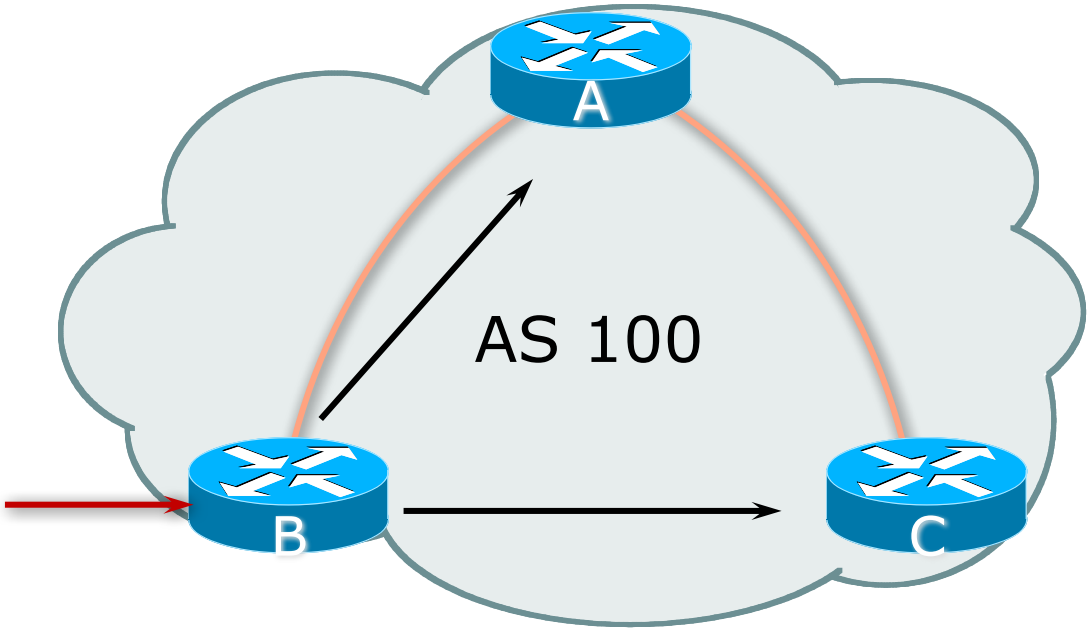
- Avoid ½n(n-1) iBGP mesh

n=1000 ⇒ nearly half a million ibgp sessions!

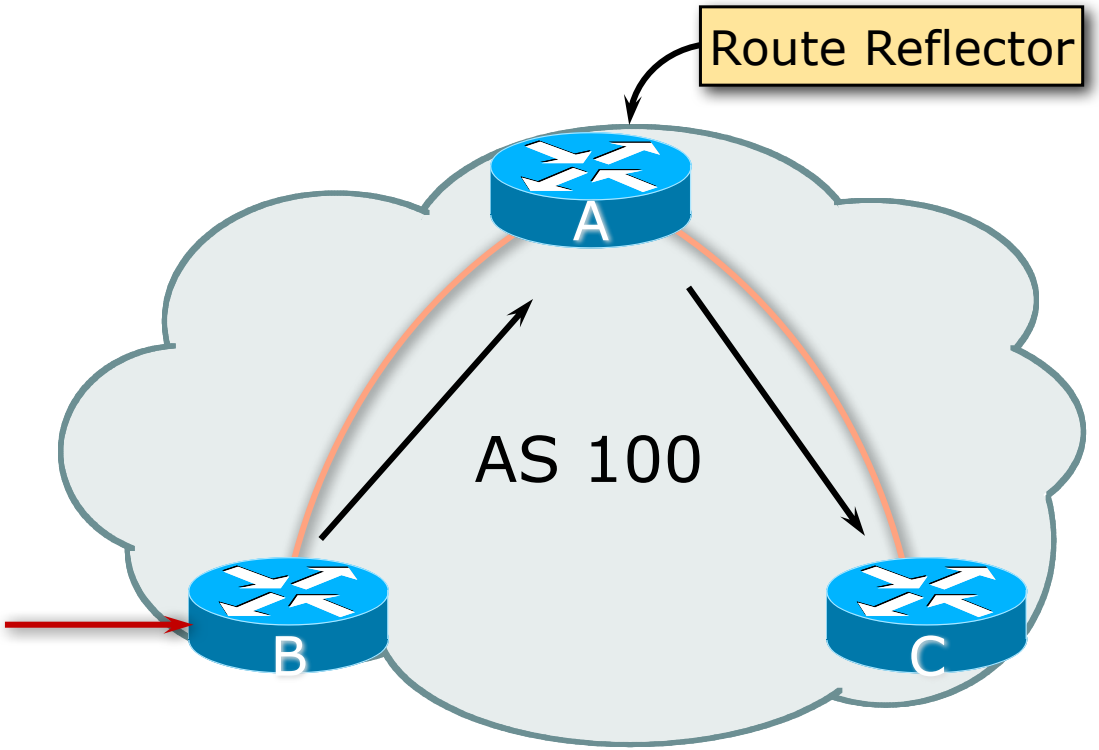


- Two solutions
  - Route reflector – simpler to deploy and run
  - Confederation – more complex, has corner case advantages

Route Reflector: Principle

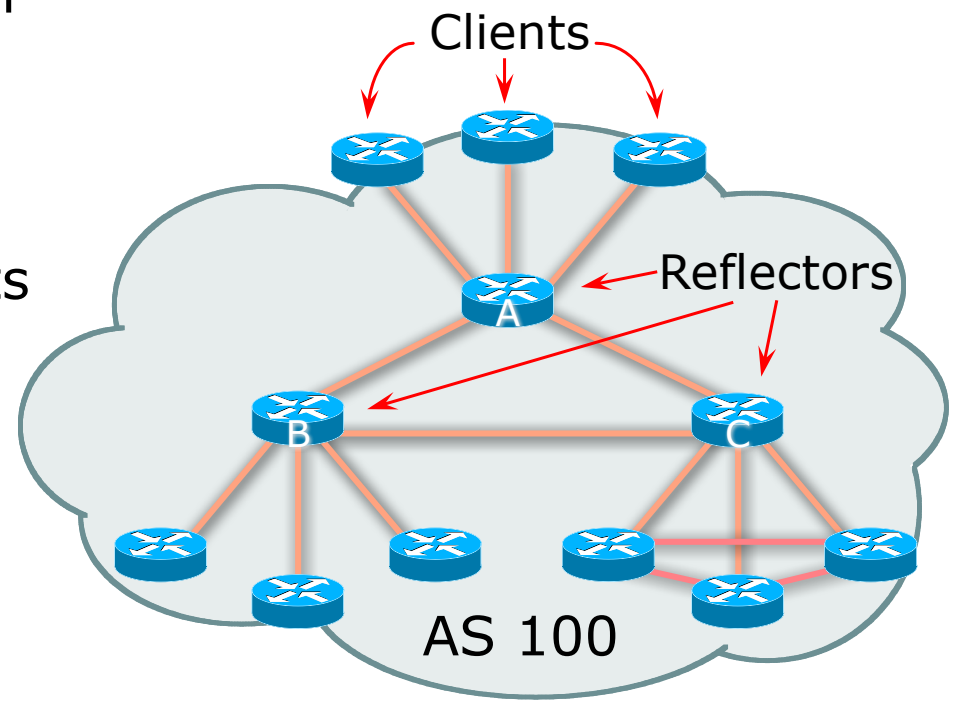


Route Reflector: Principle



Route Reflector: Rules

- Reflector receives path from clients and non-clients
- Selects best path
- If best path is from client, reflect to other clients and non-clients
- If best path is from non-client, reflect to clients only
- Non-meshed clients
- Described in RFC4456



Route Reflector: Topology

- Divide the backbone into multiple clusters
- Provision at least one Route Reflector (RR) and few clients per cluster
- Route reflectors are fully meshed
- Clients in a cluster could be fully meshed
- Single IGP still carries next-hop and any local routes

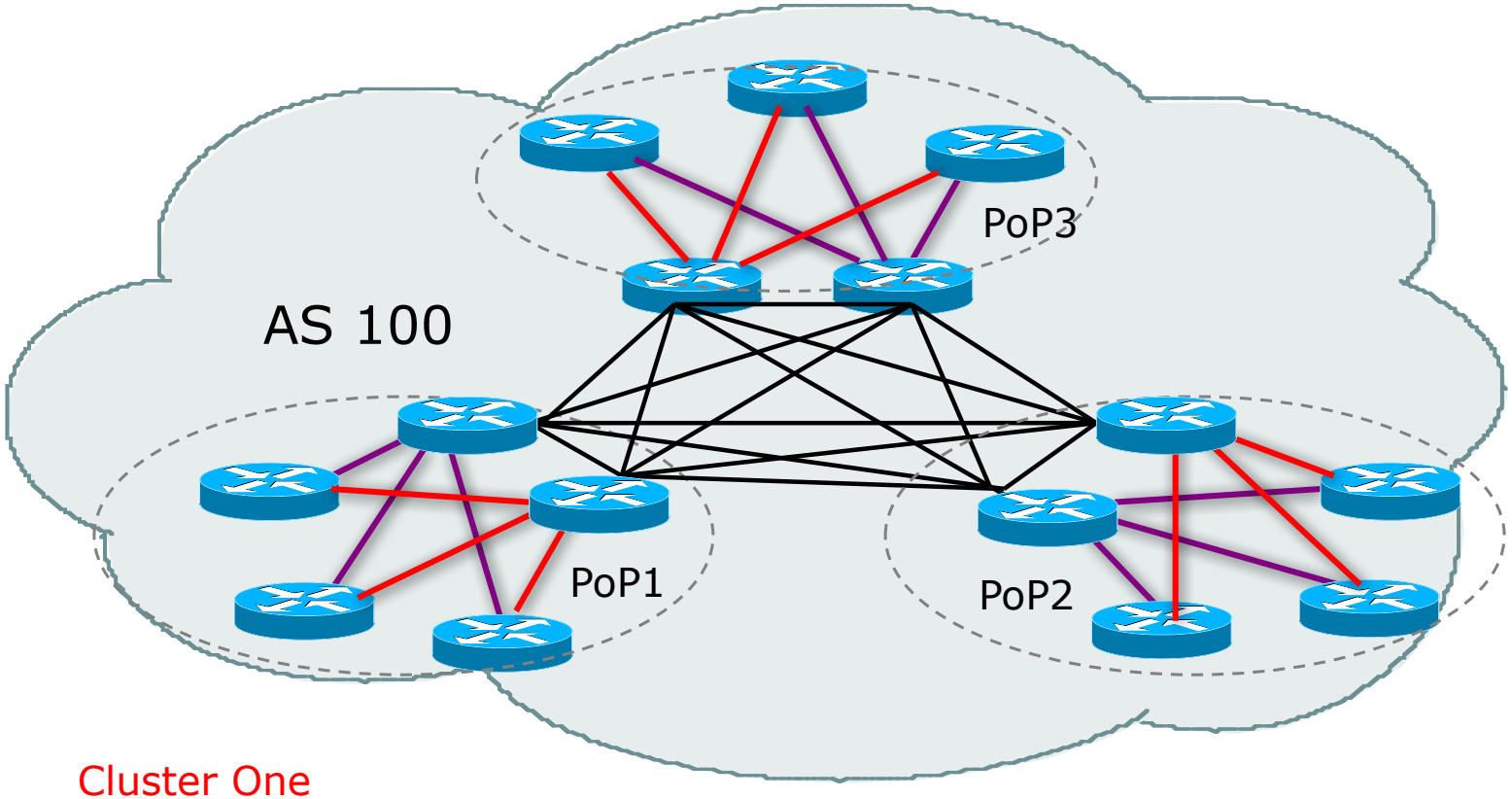
Route Reflector: Loop Avoidance

- Originator_ID attribute
  - Carries the RID of the originator of the route in the local AS (created by the RR)
- Cluster_list attribute
  - The local cluster-id is added when the update is sent by the RR
  - Cluster-id is router-id by default (usually the address of loopback interface)
  - Do NOT use **`bgp cluster-id x.x.x.x`** unless the two route reflectors are physically/directly connected

Route Reflector: Redundancy

- Multiple RRs can be configured in the same cluster – not advised!
  - All RRs in the cluster must have the same cluster-id (otherwise it is a different cluster)
- A router may be a client of RRs in different clusters
  - Common today in ISP networks to overlay two clusters – redundancy achieved that way
  - → Each client has two RRs = redundancy

Route Reflector: Redundancy



Cluster One

Cluster Two

Route Reflector: Benefits

- Solves iBGP mesh problem
- Packet forwarding is not affected
- Normal BGP speakers co-exist
- Multiple reflectors for redundancy
- Easy migration
- Multiple levels of route reflectors

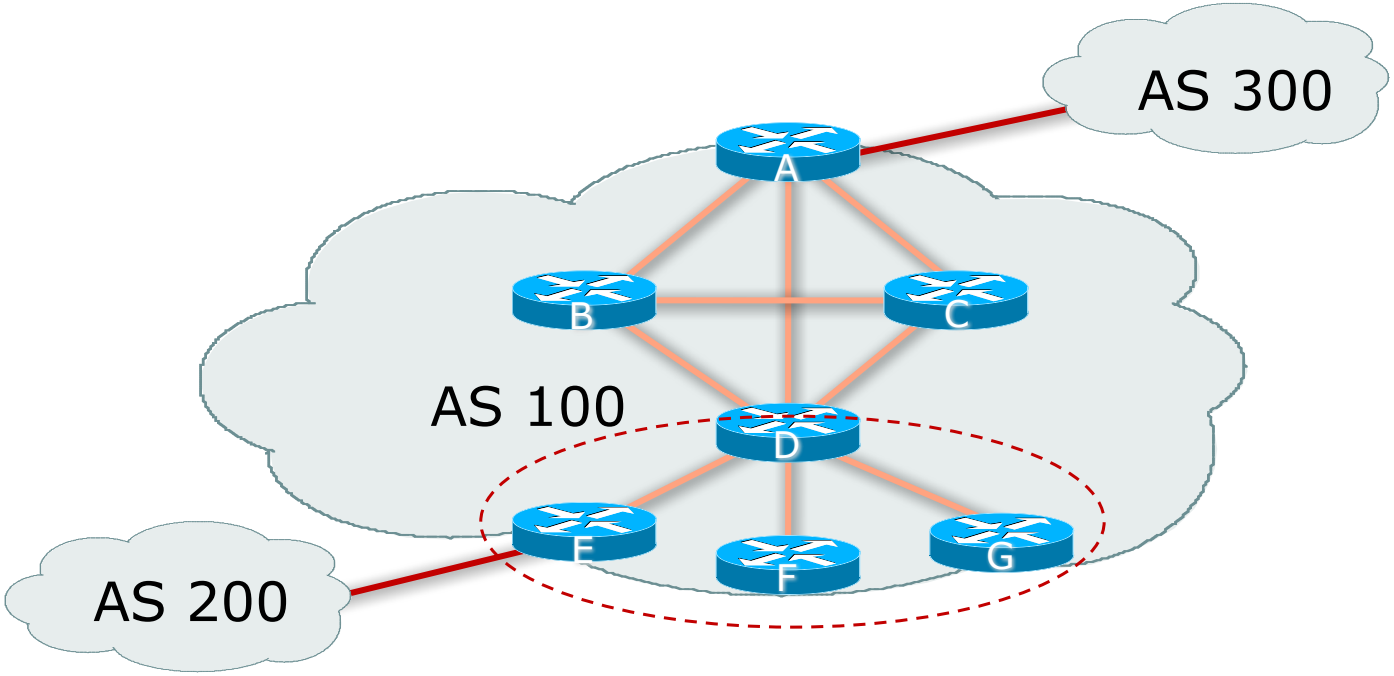
Route Reflector: Deployment

- Where to place the route reflectors?
  - Always follow the physical topology!
  - This will guarantee that the packet forwarding won't be affected
- Typical Service Provider network:
  - PoP has two core routers
  - Core routers are RR for the PoP
  - Two overlaid clusters

Route Reflector: Migration

- Typical ISP network:
  - Core routers have fully meshed iBGP
  - Create further hierarchy if core mesh too big
    - Split backbone into regions
- Configure one cluster pair at a time
  - Eliminate redundant iBGP sessions
  - Place maximum one RR per cluster
  - Easy migration, multiple levels

Route Reflector: Migration



- Migrate small parts of the network, one part at a time.

Route Reflector: Cisco IOS Configuration

- Router D configuration:

```
router bgp 100
 address-family ipv4
 ...
  neighbor 1.2.3.4 remote-as 100
  neighbor 1.2.3.4 route-reflector-client
  neighbor 1.2.3.5 remote-as 100
  neighbor 1.2.3.5 route-reflector-client
  neighbor 1.2.3.6 remote-as 100
  neighbor 1.2.3.6 route-reflector-client
 ...
```

BGP Scaling Techniques

- These 3 techniques should be core requirements on all ISP networks
  - Route Refresh (or Soft Reconfiguration)
  - Peer groups
  - Route Reflectors