IBGP POLICY AND SCALING

Tawfiq Khan
TCOM610
George Mason University

IBGP Policy

- BGP policy so far: applied to routes between customer border routers and service providers
- How to propagate BGP routes inside AS
  - ISPs: many or most routers run BGP, all routers run an IGP; usually run IBGP full-mesh or route-reflector
  - Enterprise: most routers run IGP only, a few border routers run BGP; need redistribution or other policy to propagate BGP routes
- Interactions between BGP and IGP
  - IGP cost to next-hop as the next to last BGP decision process
  - Protocol admin distance to pick eBGP > OSPF > iBGP
  - MED as decision process and export MED equals iGP cost
  - Need cooperation between IGP and BGP; otherwise potential loop, especially under failure condition

Redistributing BGP into IGP

- Redistribute BGP routes into the IGP
  - Don’t inject full BGP routes
  - Cause excessive IGP overhead: CPU, memory, convergence, IGP protocol limitations
  - Inject partial BGP routes plus default is okay
  - Usually only inject critical partial routes for performance or specific load-balancing purpose
  - Redistribution causes loss of information (AS-PATH, LOCAL_PREF, MED etc)
  - eBGP instability affects IGP stability

Example: Redistribution of full BGP routes into IGP caused loss of AS_PATH, and re-advertise whole Internet routes back to BGP caused a historic Internet meltdown

Following Default in a AS

- Inject default from AS border routers
  - Each border router injects a default into IGP
    - IGP routers might receive multiple defaults
    - Internal routes choose one default to reach border routers based on shortest IGP metrics
    - For non-Internet bound traffic, it still follows IGP paths
    - A most common and practical solution
  - Still important to run IBGP inside AS among border routers and some transit routers
  - Possible to have routing loop if multiple default and topology constraint

Case Study: default with primary/backup

- Two border routers with external links: one primary default with higher local_pref; other backup default with lower local_pref
- Remember: inside IGP routers, default has no BGP attributes, only IGP metric (cost) counts!
- Possible to have conflicting IGP default direction and BGP default direction, resulting into routing loop
- Loop Avoidance:
  - Manipulate IGP metric: backup router injects default with very high metric
  - Ensure IBGP path is shorter than IGP path
  - Run BGP on transit routers (even not border routers): bring transit routers into IBGP mesh
  - Only one default is generated and populated into IGP at one time and dynamically generate another one under failure condition

Generation of defaults

- Desired behavior
  - BGP router should stop injecting 0/0 into IGP if external link fails
    - Achieved by redistributing BGP default into IGP, RIP only, not OSPF
  - BGP router should inject 0/0 into IGP only if default is pointing to external link
    - Cisco OSPF can originate default based on (a) existence of default in routing table Or (b) default pointing to external links
- External link failures
  - redistribute 0/0 into IGP (if possible)
  - IGP default disappears when external 0/0 ceases to exist
- External and internal preferences
  - backup router generates default if exterior default preferred
  - stop generating 0/0 if interior default preferred

Cisco Command

```
router ospf 10
  default-information originate [always] [metric metric-val] [metric-type type-val] [route-map map-name]

default-information originate always

default-information originate route-map SEND_DEF_IF
access-list 1 permit 0.0.0.0
access-list 2 permit 172.16.20.1
route-map SEND_DEF_IF permit 10
  match ip address 1
  match ip next-hop 2
```

Other Policy

- BGP routing is dynamical by nature and each network’s exit point may change dynamically
- If multiple exit points exist and exit point selection changes, your IGP default path may conflict with your BGP exit points
- Prevention strategies
  - deterministic exit points selected in both BGP and IGP
  - prevent traffic from IBGP routers traveling back over IGP-only routers
  - Make IGP and BGP selection consistent

IBGP Review

- Run IBGP to pass eBGP routes from border routes to other internal routers (and other border routers)
- IBGP speaker can’t advertise IBGP learnt routes to other IBGP speaker
  - If this rule is not followed, potential routing loop can occur, since AS_PATH based loop detection does not work in iBGP
  - Therefore a full IBGP mesh is required
- IBGP default behaviors
  - Only send update to IBGP peers if (i) newly learnt external routes (ii) newly selected best route (iii) withdraw routes
  - No change on AS_PATH
  - No change on next-hop unless next-hop-self is configured
  - Actually it is recommended that you don’t apply any policy to IBGP process; otherwise you may end up partitioning your AS

Route Reflector – RFC 4456

- Break the IBGP rule: IBGP speaker will not pass iBGP-learnt routes to other iBGP peers
- Routers configured as route reflectors will conditionally pass IBGP updates to other IBGP speakers
- Each RR reflects routes on behalf of a set of clients
- The combination of a route reflector and its clients is called a *cluster*
- RR Behaviors:
  - A Route from a Non-Client IBGP peer => Reflect to all the Clients.
  - A Route from a Client peer => Reflect to all the Non-Client peers and also to the Client peers.
  - No need for full IBGP among clients

Route Reflectors

- Only Configuration on the route reflector
    - neighbor *peer-address* route-reflector-client
- Configuration on the route reflector client and other peers is normal – no changes
- Additional processing and update overhead for RRs, as they see all potential candidate routes
    - RRs will make BGP decision and only reflect best routes
- If RR clients are full-mesh among themselves, the route reflector can be configured not to do client-to-client reflection
    - [no] bgp client-to-client reflection

RR

- If a given cluster has more than one route reflector for redundancy, need to set the cluster-ID
  - bgp route-reflector *cluster-ID*
- Cluster-ID can be any unique four-byte integer within AS
  - Must bear same cluster-number for RRs in the same cluster
  - For loop avoidance inside a cluster
- Redundant route reflector must be supported by physical redundancy, or it will add little value
  - In case of link failure, other clients should be able to reach alternative RR
- Route reflector selection and cluster assignment should take topology into account
  - One cluster with redundant RR in each geographical locations or POP

RR Example

```
! Router A serving as a route reflector
router bgp 100
  neighbor 1.1.1.2 remote-as 100
  neighbor 1.1.1.2 route-reflector-client
  neighbor 1.1.1.3 remote-as 100
  neighbor 1.1.1.3 route-reflector-client
  neighbor 1.1.1.4 remote-as 100
  neighbor 1.1.1.4 route-reflector-client
  neighbor 1.1.1.5 remote-as 100
! normal peer
  neighbor 1.1.1.6 remote-as 100
! route reflector peer
  neighbor 1.1.1.7 remote-as 100
```

RR Problems

- Historical problems with using peer groups and route reflectors have been cleared up
- Route reflector clients should not have IBGP peerings outside of the cluster
  - they should not have any IBGP peerings inside of the cluster except with the route reflectors unless client-to-client reflection is turned off (on by default)
- RR does limit a router's view
  - RR will only reflect best route, not other candidate routes
  - RR made route selection on behalf of clients
  - Combined with MED might cause infinite MED oscillation problem (RFC 3345)

Persistent Route Oscillations RFC 3345

- Type I Churn
  - Conditions: Single Level RR structure and accepting MED from more than two different AS and MED values are unique
  - Workaround: inter-cluster links have higher IGP cost than intra-cluster links; or don’t use MED for BGP route selection
- Type II Churn
  - Conditions: Hierarchical RR structure and accepting MED from more than two different AS and MED values are unique
  - Workaround: Don’t accept MED from peers; Always compare-MED; full-mesh RR clients

RR Loop-avoidance

- BGP update inside AS: can’t be detected by AS_PATH loop-detection
- Avoiding loops
  - Originator ID (optional, non-transitive, type 9) is added by the route reflector – it is the router ID of the originator of the route (not the route reflector necessarily) within the local AS
  - Originator ID is different from the originator of the route in the global network
  - Clients that hear a route with Router ID = Originator ID drop the announcement – this will only occur in a severely mis-configured network

RR Loop-avoidance

- Avoiding loops
  - Cluster List (optional, non-transitive, type 10) is a sequence of cluster ID that this update has traversed, similar to AS_Path
  - RR appends its own cluster ID when sending update to peers outside cluster
  - When received update with its own cluster ID, RR drops the update
  - The cluster list should not grow beyond one item for normal setups.
  - Hierarchical route reflectors have not worked historically, but newer implementation has solved this problem

RR Design BCP

- Select Two Redundant RR in each metro POP location
- All other routers in the same POP IBGP mesh with RRs in the same locations
- No IBGP full-mesh among clients in the same cluster
- Full IBGP mesh among RRs
- No hierarchical RR structure
- Choose RR topology in such a way so that RR is never forced to make BGP decision based on IGP cost much different from its clients
    - Pop-based RR design

AS Confederations (RFC 5065)

- Reduce the IBGP mesh within an AS
  - Divide the AS into multiple sub-ASs and assign the whole group a single AS that will be invisible to external peers
  - Each sub-AS will maintain its own fully meshed IBGP configuration
  - Peering with external peers use the confederation ID as the AS number
- IBGP peerings within the configured sub-AS
- EBGP peerings to confederation peers will follow IBGP rules – Next_Hop, MED and Local_Pref are preserved
- Sub-AS are shielded from the outside world
- Configure with the following commands
  - bgp confederation identifier *as-num*
  - bgp confederation peers *list-of-confederation peer-asnums*

Example

```
router bgp 100
 bgp confederation identifier 1000
 bgp confederation peers 200 300
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.2 remote-as 100
 neighbor 2.2.2.2 remote-as 200
 neighbor 3.3.3.3 remote-as 300
 neighbor 4.4.4.4 remote-as 500
```

Neighbors 1.1.1.1, 1.1.1.2, 2.2.2.2, and 3.3.3.3 would set up peerings with ASN 100, while neighbor 4.4.4.4 would set up a peering with ASN 1000

AS Confederations

- Usually you will want to use private ASNs for confederations
- Sometimes confederations result in non-optimal routing within a network
  - Limited AS_PATH view, since sub-AS do not influence overall AS_PATH length
- Pay attention to some sample paths after configuring
- BGP decision algorithm changes
- From EBGP > IBGP to EBGP > Confederation EBGP > IBGP
- Typical Design: central backbone sub-AS connecting to all other sub-ASs

RR or Confederations

- Both can be deployed anywhere in the network, and support hierarchical deployment
- Confederation has medium scalability, while RR has very high scalability
- Confederation has medium to high complexity in migration, while it is very low to migrate to RR
- RR problem: limited visibility into candidate routes cause oscillation, traffic engineering issues etc

IGP Scalability

- IGP is typical under one administrator and should not grow out of control
  - IGP hierarchical design to scale
  - OSPF multiple areas or multiple level ISIS
  - Use aggregation or summary in each area/level
- Use BGP to segment your network and scale IGP
  - Multiple regions with different IGP connected by IBGP
    - Inject default into IGP in each region and inject IGP routes into IBGP
    - All EBGP router should be part of IBGP
  - Multiple regions with different IGP and AS connected by EBGP
    - May be difficult to get public AS for this design
    - May use private ASN and one public AS to connect to Internet
  - Use Confederation to control IGP expansion
    - Migration overhead and no policies among subAS