BGP Scaling Techniques

Scalable Infrastructure
Workshop

AfNOG 2010

BGP Scaling Techniques

- How to scale iBGP mesh beyond a few peers?
- How to implement new policy without causing flaps and route churning?
- How to reduce the overhead on the routers?

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
  - Peer-groups
  - Route Reflectors (and Confederations)
- Deprecated Scaling Techniques
  - Soft Reconfiguration
  - Route Flap Damping

Dynamic Reconfiguration

Non-destructive policy changes

Route Refresh

- Problem:
- Hard BGP peer reset required after every policy change because the router does not store prefixes that are rejected by policy
- Hard BGP peer reset:
  - Tears down BGP peering
  - Consumes CPU
  - Severely disrupts connectivity for all networks
- Solution:

Route Refresh

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
- clear ip bgp x.x.x.x [soft] in tells peer to resend full BGP announcement
- clear ip bgp x.x.x.x [soft] out resends full BGP announcement to peer

Dynamic Reconfiguration

- Use Route Refresh capability if supported
  - Supported on virtually all routers
  - Find out from "show ip bgp neighbor"
  - Non-disruptive, "Good For the Internet"
- Otherwise use Soft Reconfiguration IOS feature
- Only hard-reset a BGP peering as a last resort

Consider the impact of a hard-reset of BGP to be equivalent to a router reboot

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

Configuring Soft reconfiguration

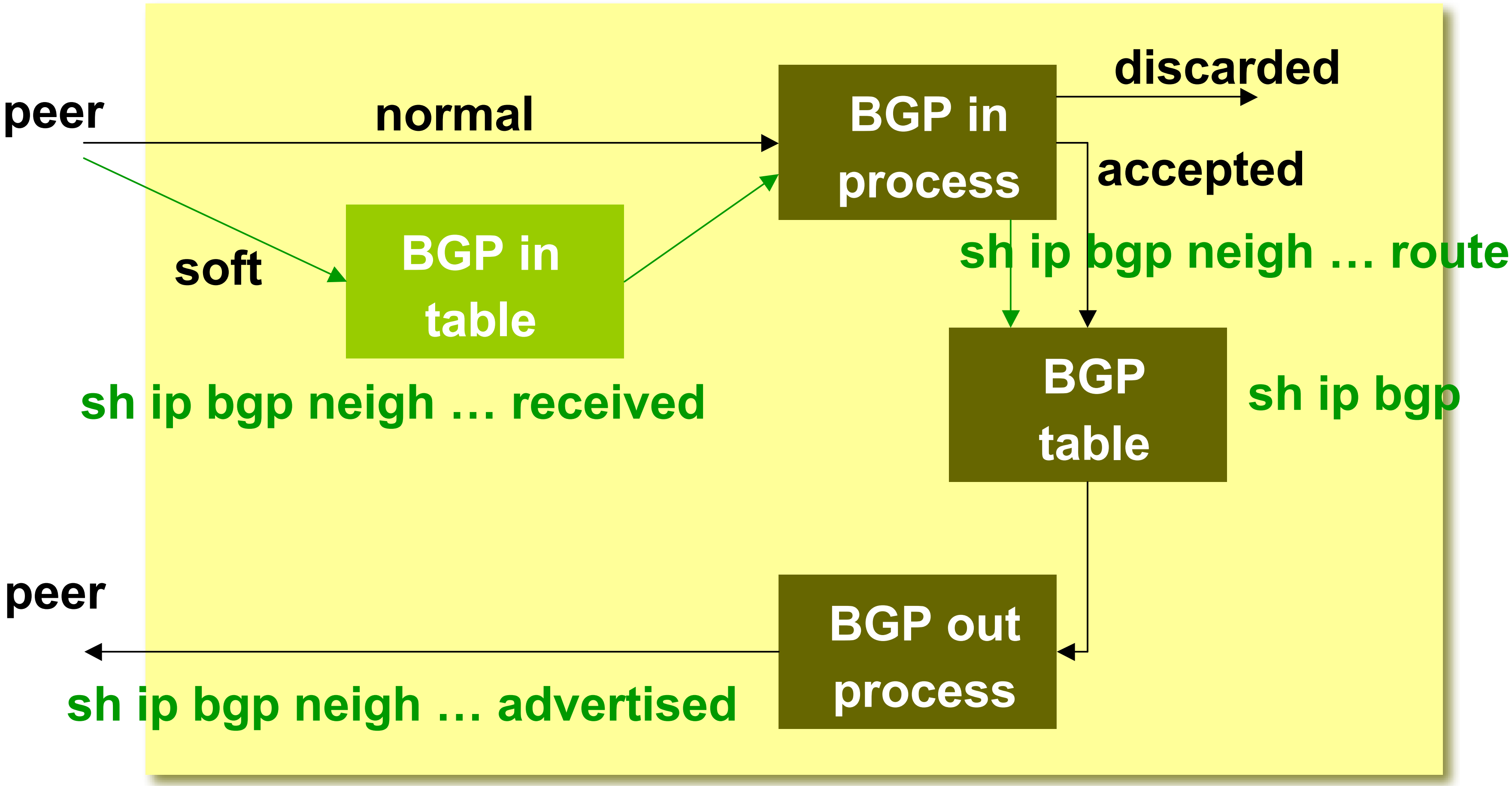
```
router bgp 100
 neighbor 1.1.1.1 remote-as 101
 neighbor 1.1.1.1 route-map infilter in
 neighbor 1.1.1.1 soft-reconfiguration inbound
! Outbound does not need to be configured
!
```

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

  clear ip bgp 1.1.1.1 soft [in | out]

- Notes:
  - 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

Soft Reconfiguration



Managing Policy Changes

- **`clear ip bgp <addr> [soft] [in|out]`**

  <addr> may be any of the following:

x.x.x.x	IP address of a peer
*	all peers
ASN	all peers in an AS
external	all external peers
peer-group <name>	all peers in a peer-group

Peer Groups

Saving Time!

Peer Groups

- Without peer groups
  - iBGP neighbours receive same update
  - Large iBGP mesh slow to build
  - Router CPU wasted on repeat calculations
- Solution – peer groups!
  - Group peers with same outbound policy
  - Updates are generated once per group

Peer Groups – Advantages

- Makes configuration easier
- Makes configuration less prone to error
- Makes configuration more readable
- Lower router CPU load
- iBGP mesh builds more quickly
- Members can have different inbound policy
- Can be used for eBGP neighbours too!

Configuring Peer Group

```
router bgp 100
 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 1.1.1.1 peer-group ibgp-peer
 neighbor 2.2.2.2 peer-group ibgp-peer
 neighbor 2.2.2.2 route-map  infilter in
 neighbor 3.3.3.3 peer-group ibgp-peer
```

- Note how 2.2.2.2 has different inbound filter from the peer-group

Configuring Peer Group

```
router bgp 100
 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
```

Peer Groups

- Always configure peer-groups for iBGP
  - Even if there are only a few iBGP peers
  - Easier to scale network in the future
- 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
- Peer-groups are essentially obsoleted
  - But are still widely considered best practice
  - Replaced by update-groups (internal IOS coding – not configurable)
  - Enhanced by peer-templates (allowing more complex constructs)

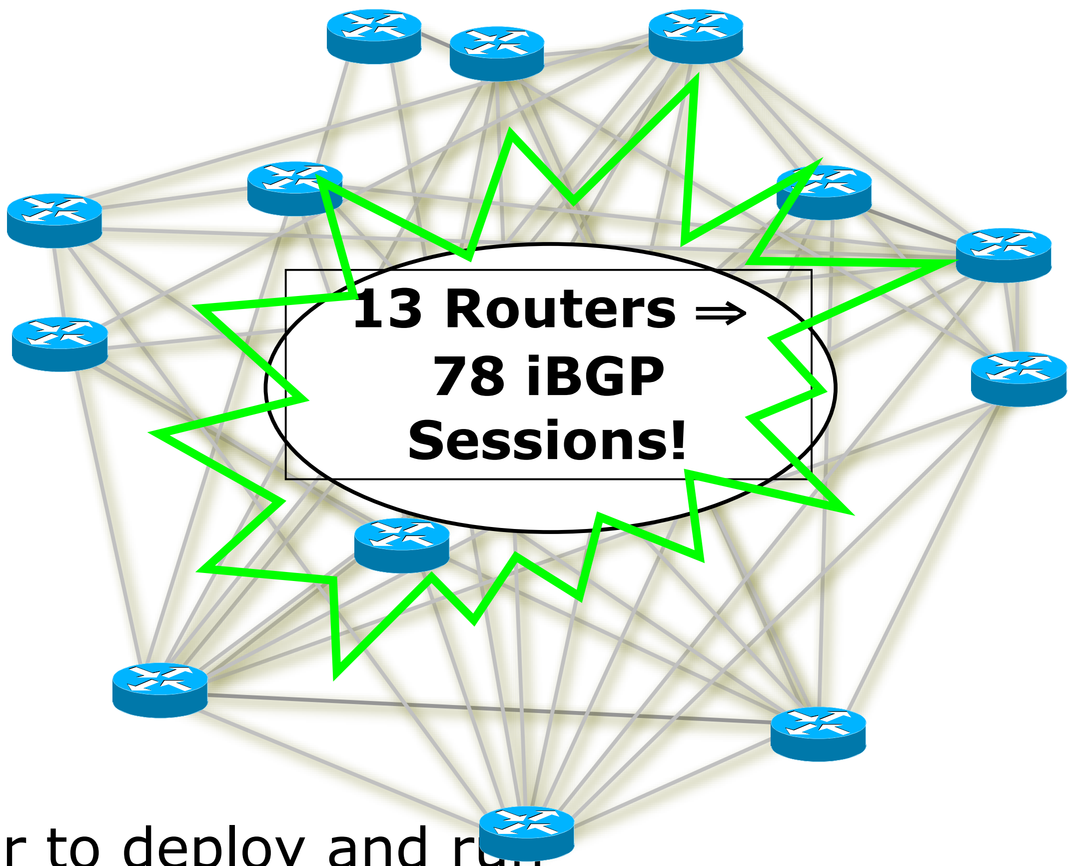
Route Reflectors

Bigger networks!

Scaling iBGP mesh

Avoid n(n-1)/2 iBGP mesh

n=1000 ⇒ nearly half a million ibgp sessions!

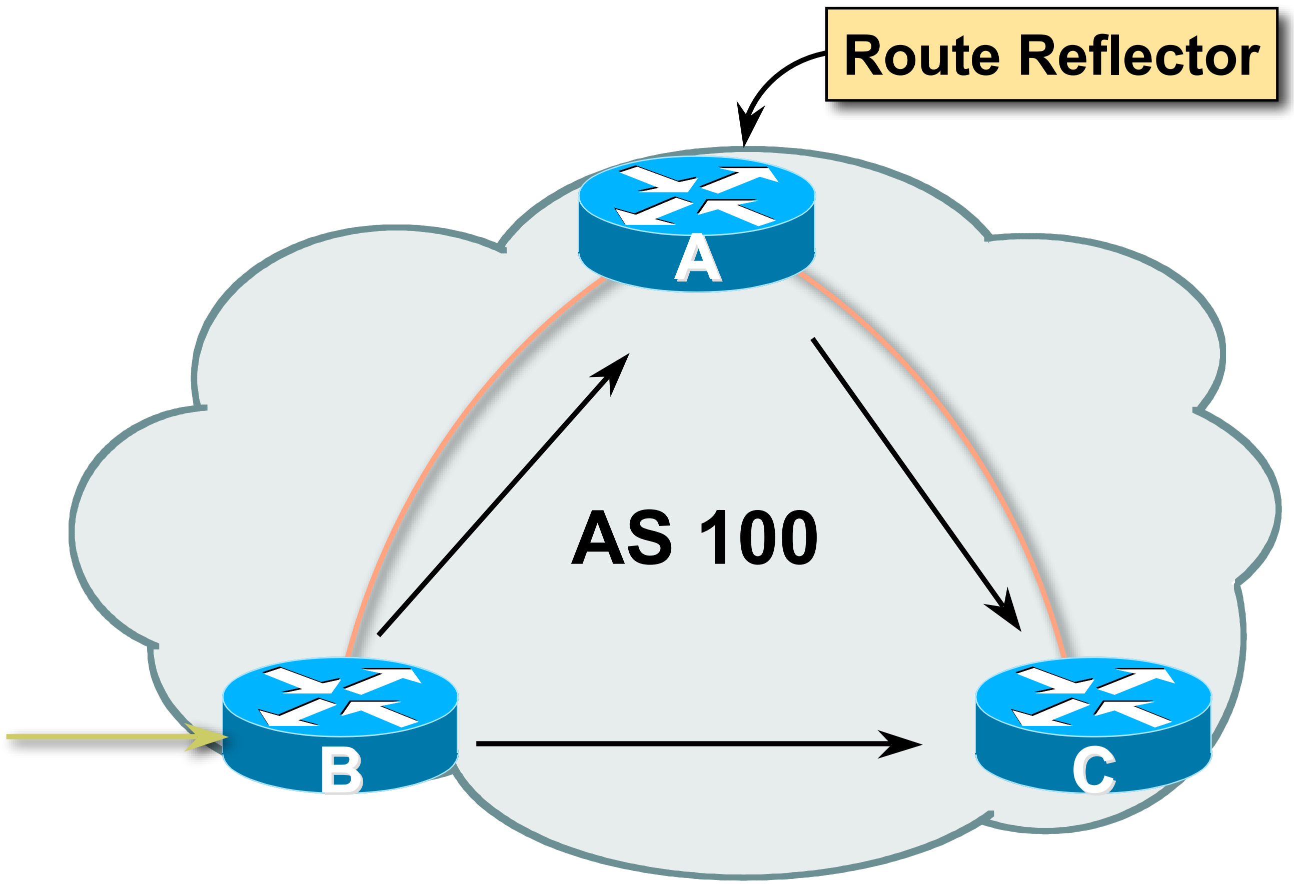


Two solutions

Route reflector – simpler to deploy and run

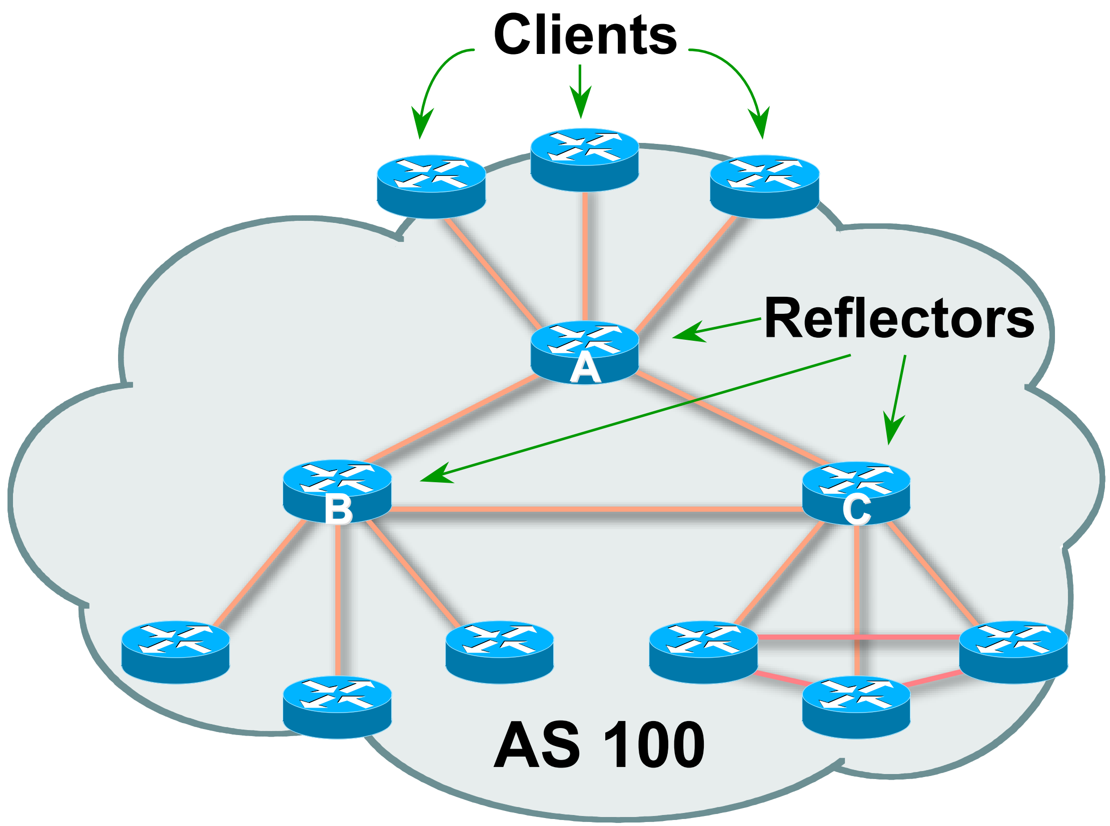
Confederation – more complex, corner case benefits

Route Reflector: Principle



Route Reflector

- 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
- At least one route reflector and few clients per cluster
- Route reflectors are fully meshed
- Clients in a cluster could be fully meshed
- Single IGP to carry next hop and local routes

Route Reflectors: 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 (address of loopback)
  - Do NOT use *bgp cluster-id x.x.x.x*

Route Reflectors: Redundancy

- Multiple RRs can be configured in the same cluster – not advised!
  - All RRs are 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 clusters – redundancy achieved that way
  - Each client has two RRs = redundancy

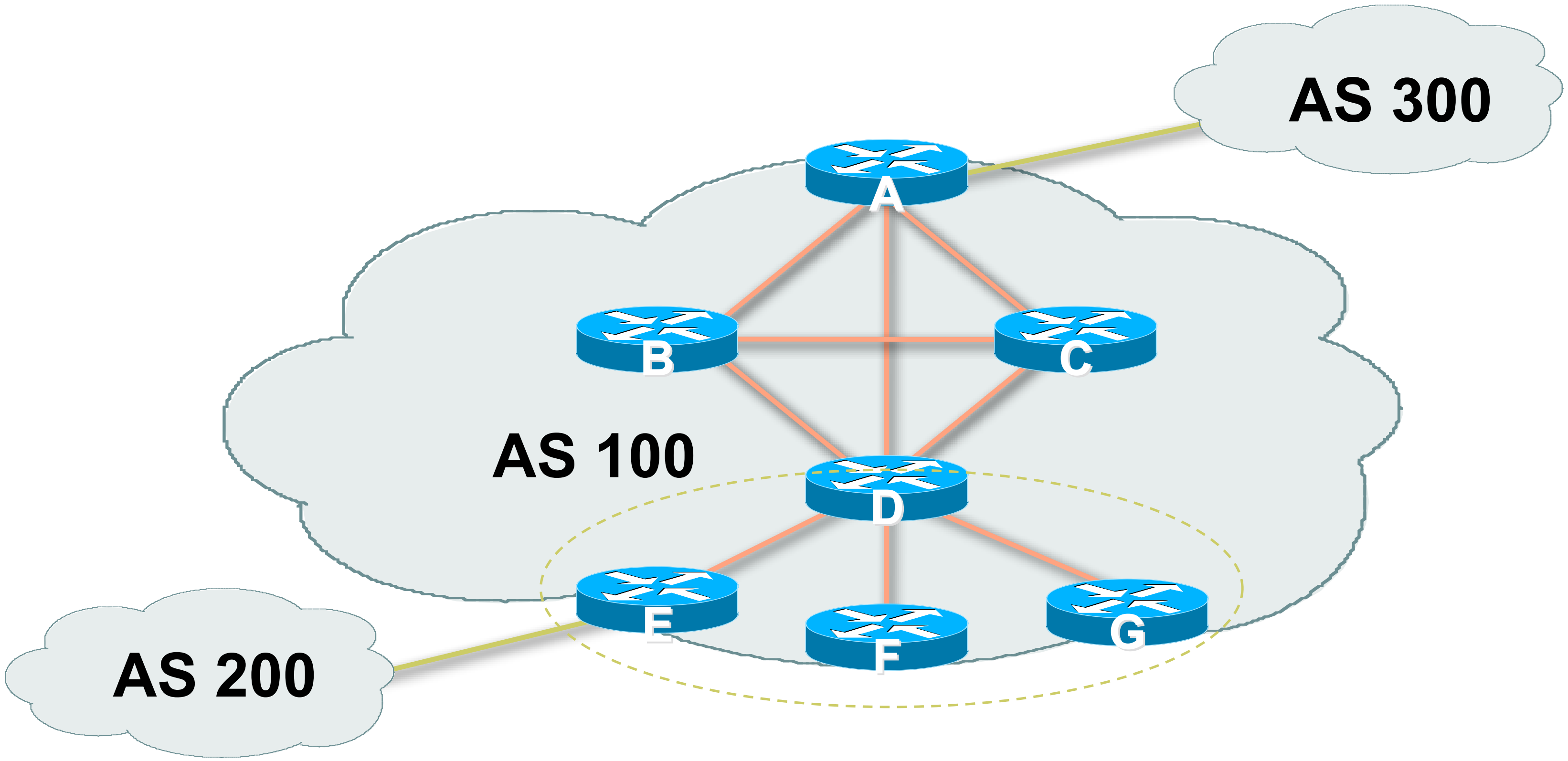
Route Reflectors: Benefits

- 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 Reflectors: Migration

- Where to place the route reflectors?
  - Follow the physical topology!
  - This will guarantee that the packet forwarding won’t be affected
- Configure one RR at a time
  - Eliminate redundant iBGP sessions
  - Place one RR per cluster

Route Reflector: Migration



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

Configuring a Route Reflector

```
router bgp 100
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 route-reflector-client
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 route-reflector-client
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 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

Route Flap Damping

Network Stability for the 1990s

Network Instability for the 21st Century!

Route Flap Damping

- For many years, Route Flap Damping was a strongly recommended practice
- Now it is strongly discouraged as it causes far greater network instability than it cures
- But first, the theory…

Route Flap Damping

- Route flap
  - Going up and down of path or change in attribute
    - BGP WITHDRAW followed by UPDATE = 1 flap
    - eBGP neighbour going down/up is NOT a flap
  - Ripples through the entire Internet
  - Wastes CPU
- Damping aimed to reduce scope of route flap propagation

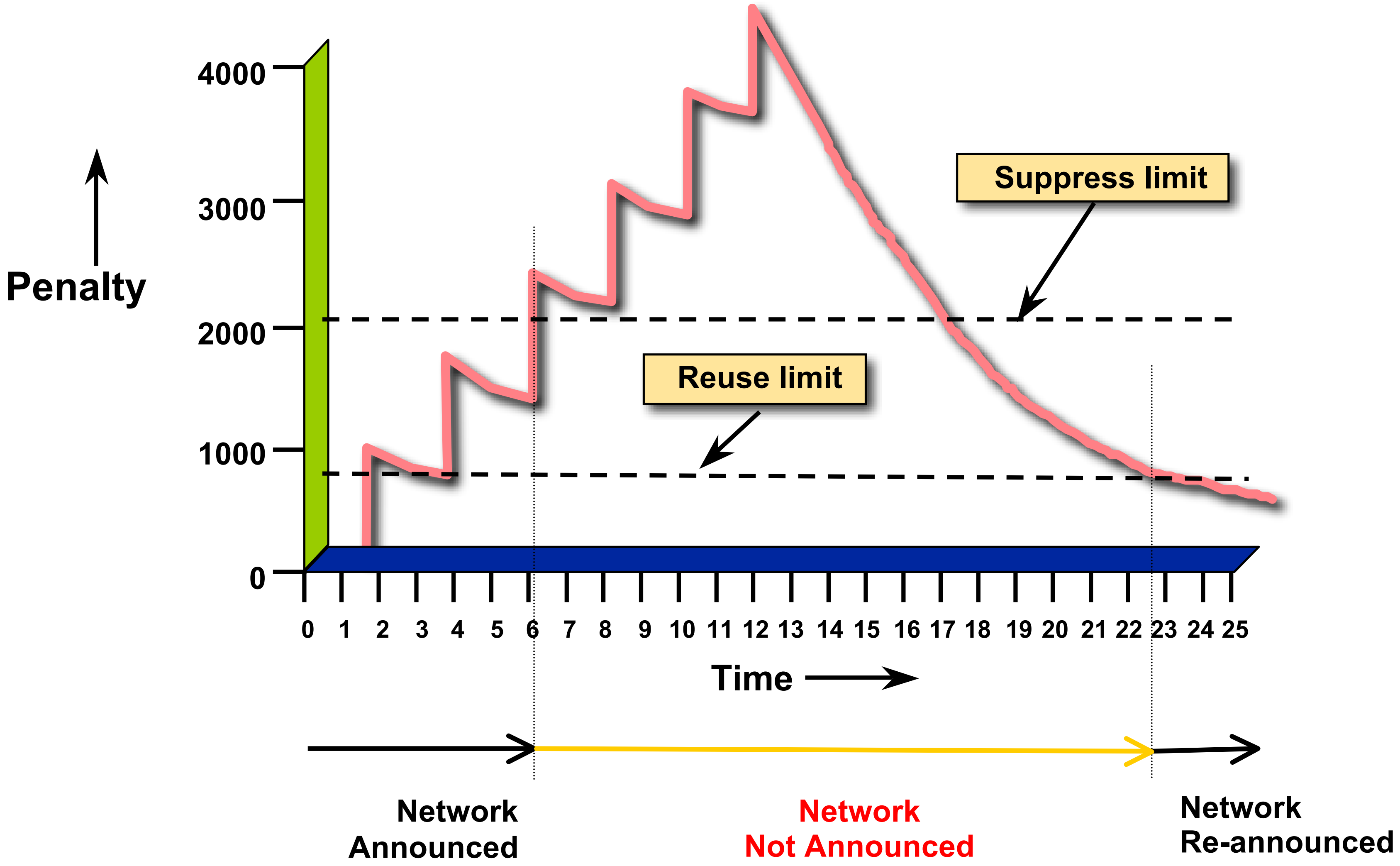
Route Flap Damping (Continued)

- Requirements
  - Fast convergence for normal route changes
  - History predicts future behaviour
  - Suppress oscillating routes
  - Advertise stable routes
- Implementation described in RFC2439

Operation

- Add penalty (1000) for each flap
  - Change in attribute gets penalty of 500
- Exponentially decay penalty
  - Half life determines decay rate
- Penalty above suppress-limit
  - Do not advertise route to BGP peers
- Penalty decayed below reuse-limit
  - Re-advertise route to BGP peers
  - Penalty reset to zero when it is half of reuse-limit

Operation



Penalty
4000
3000
2000
1000
0
Suppress limit
Reuse limit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Time
Network Announced
Network Not Announced
Network Re-announced

Operation

- Only applied to inbound announcements from eBGP peers
- Alternate paths still usable
- Controlled by:
  - Half-life (default 15 minutes)
  - reuse-limit (default 750)
  - suppress-limit (default 2000)
  - maximum suppress time (default 60 minutes)

Configuration

- Fixed damping

```
router bgp 100
 bgp dampening [<half-life> <reuse-value>
  <suppress-penalty> <maximum suppress time>]
```

- Selective and variable damping

```
bgp dampening [route-map <name>]
 route-map <name> permit 10
  match ip address prefix-list FLAP-LIST
  set dampening [<half-life> <reuse-value>
 <suppress-penalty> <maximum suppress time>]
ip prefix-list FLAP-LIST permit 192.0.2.0/24 le 32
```

Route Flap Damping History

- First implementations on the Internet by 1995
- Vendor defaults too severe
  - RIPE Routing Working Group recommendations in ripe-178, ripe-210, and most recently ripe-229
  - But many ISPs simply switched on the vendors' default values without thinking

Serious Problems:

- "Route Flap Damping Exacerbates Internet Routing Convergence"
  - Zhuoqing Morley Mao, Ramesh Govindan, George Varghese & Randy H. Katz, August 2002
- “What is the sound of one route flapping?”
  - Tim Griffin, June 2002
- Various work on routing convergence by Craig Labovitz and Abha Ahuja a few years ago
- “Happy Packets”
  - Closely related work by Randy Bush et al

Problem 1:

- One path flaps:
  - BGP speakers pick next best path, announce to all peers, flap counter incremented
  - Those peers see change in best path, flap counter incremented
  - After a few hops, peers see multiple changes simply caused by a single flap → prefix is suppressed

Problem 2:

- Different BGP implementations have different transit time for prefixes
  - Some hold onto prefix for some time before advertising
  - Others advertise immediately
- Race to the finish line causes appearance of flapping, caused by a simple announcement or path change → prefix is suppressed

Solution:

- Do NOT use Route Flap Damping whatever you do!
- RFD will unnecessarily impair access
  - to your network and
  - to the Internet
- More information contained in RIPE Routing Working Group recommendations:
  - www.ripe.net/ripe/docs/ripe-378.[pdf,html,txt]