Cisco.com

Introduction to BGP

Border Gateway Protocol

- **Routing Protocol used to exchange routing information between networks**

  exterior gateway protocol

- **RFC1771**

  work in progress to update

  `draft-ietf-idr-bgp4-18.txt`

- **Currently Version 4**
- **Runs over TCP**

BGP

- Path Vector Protocol
- Incremental Updates
- Many options for policy enforcement
- Classless Inter Domain Routing (CIDR)
- Widely used for Internet backbone
- Autonomous systems

Path Vector Protocol

- **BGP is classified as a *path vector* routing protocol** (see RFC 1322)

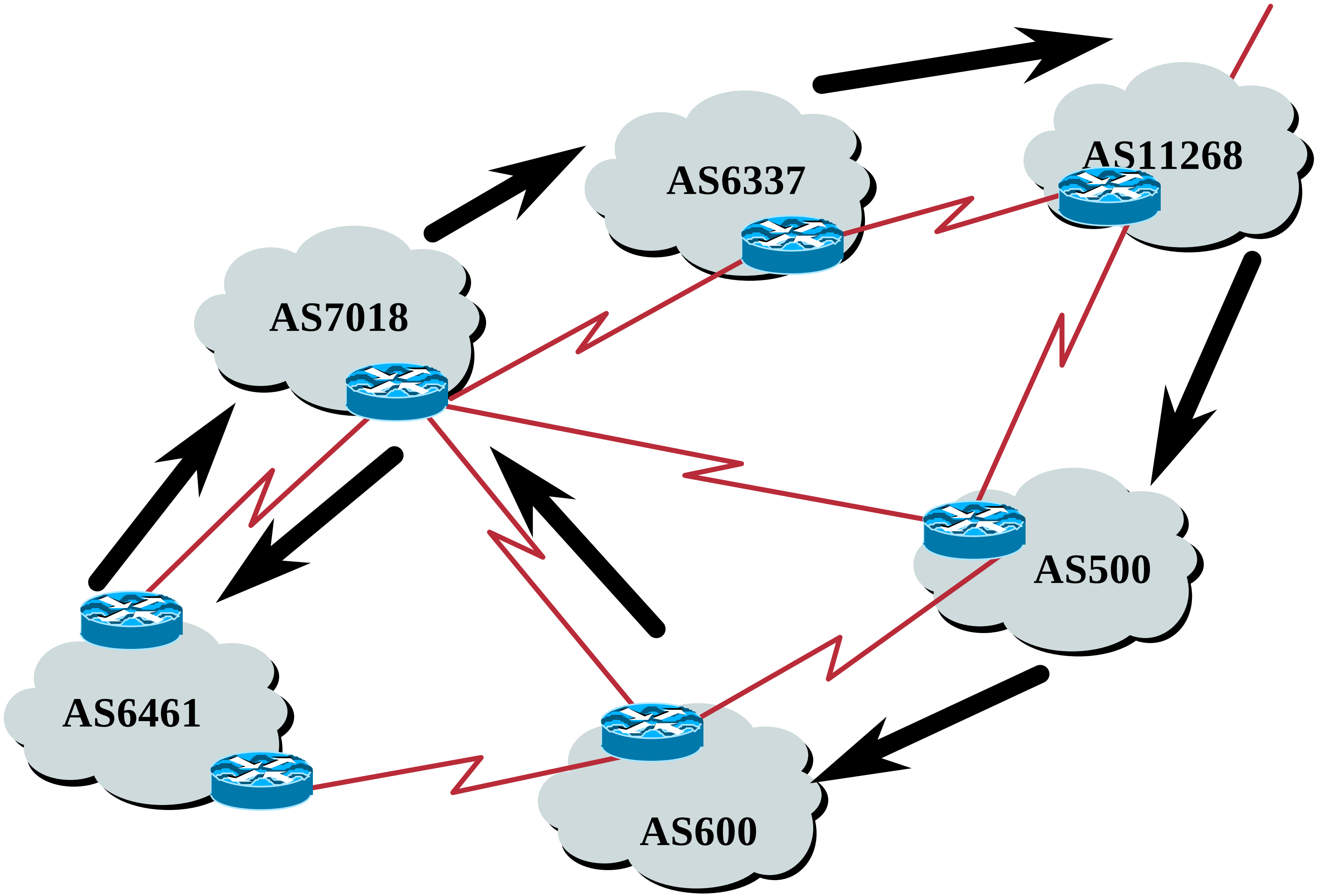
  A path vector protocol defines a route as a pairing between a destination and the attributes of the path to that destination.

12.6.126.0/24 207.126.96.43 1021 0 6461 7018 6337 11268 i

AS Path

Path Vector Protocol

Cisco.com



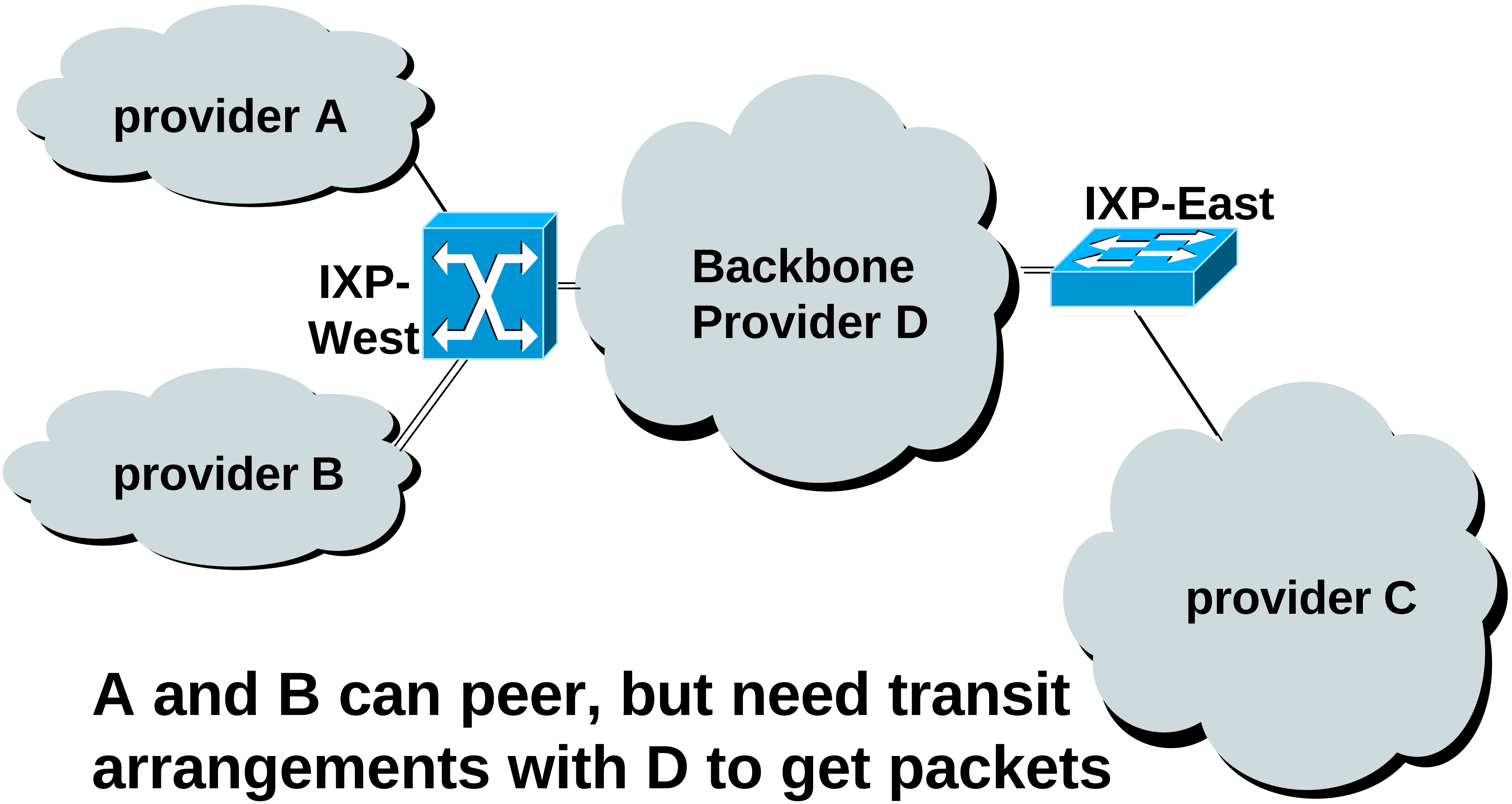
Definitions

- **Transit – carrying traffic across a network, usually for a fee**
- **Peering – exchanging routing information and traffic**
- **Default – where to send traffic when there is no explicit match in the routing table**

Default Free Zone

The default free zone is made up of Internet routers which have explicit routing information about the rest of the Internet, and therefore do not need to use a default route.

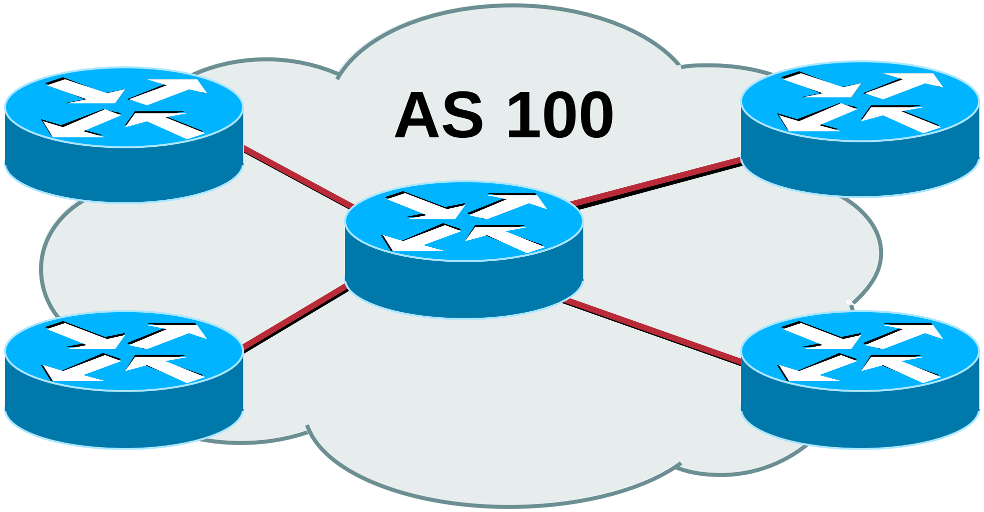
Peering and Transit example



A and B can peer, but need transit arrangements with D to get packets to/from C

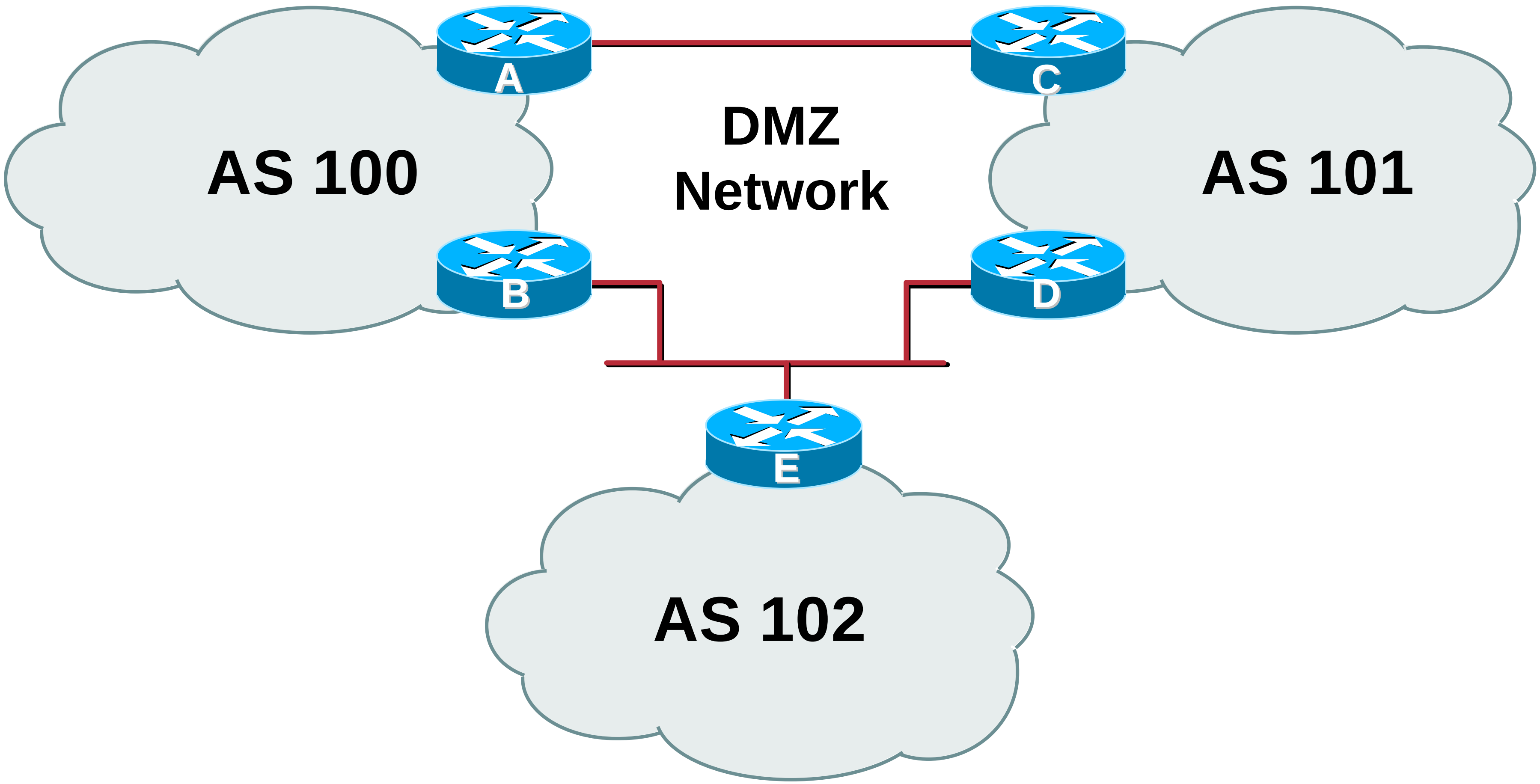
Autonomous System (AS)

Cisco.com



- Collection of networks with same routing policy
- Single routing protocol
- Usually under single ownership, trust and administrative control

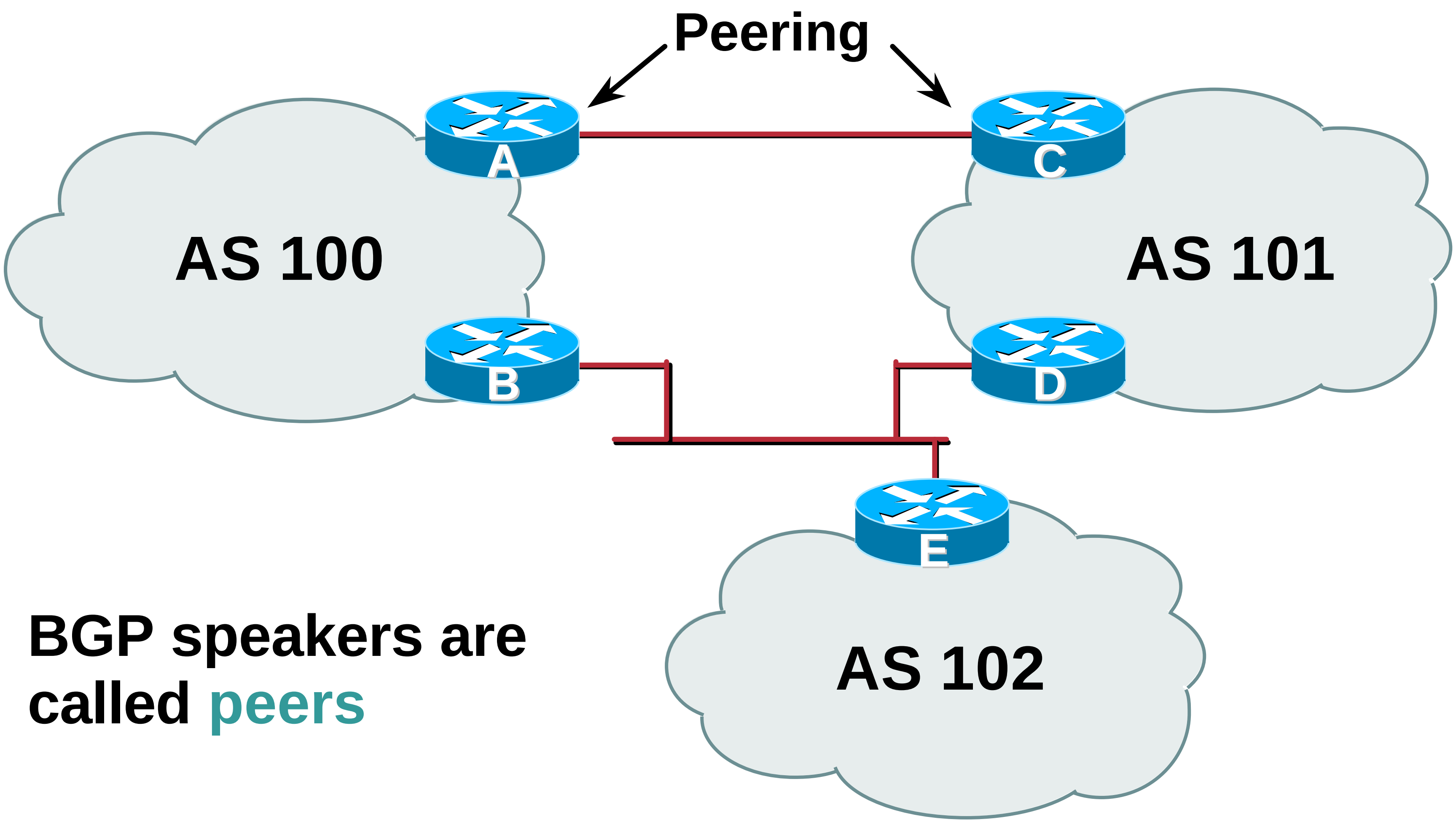
Demarcation Zone (DMZ)



- **Shared network between ASes**

BGP Basics

Cisco.com



BGP speakers are called peers

BGP General Operation

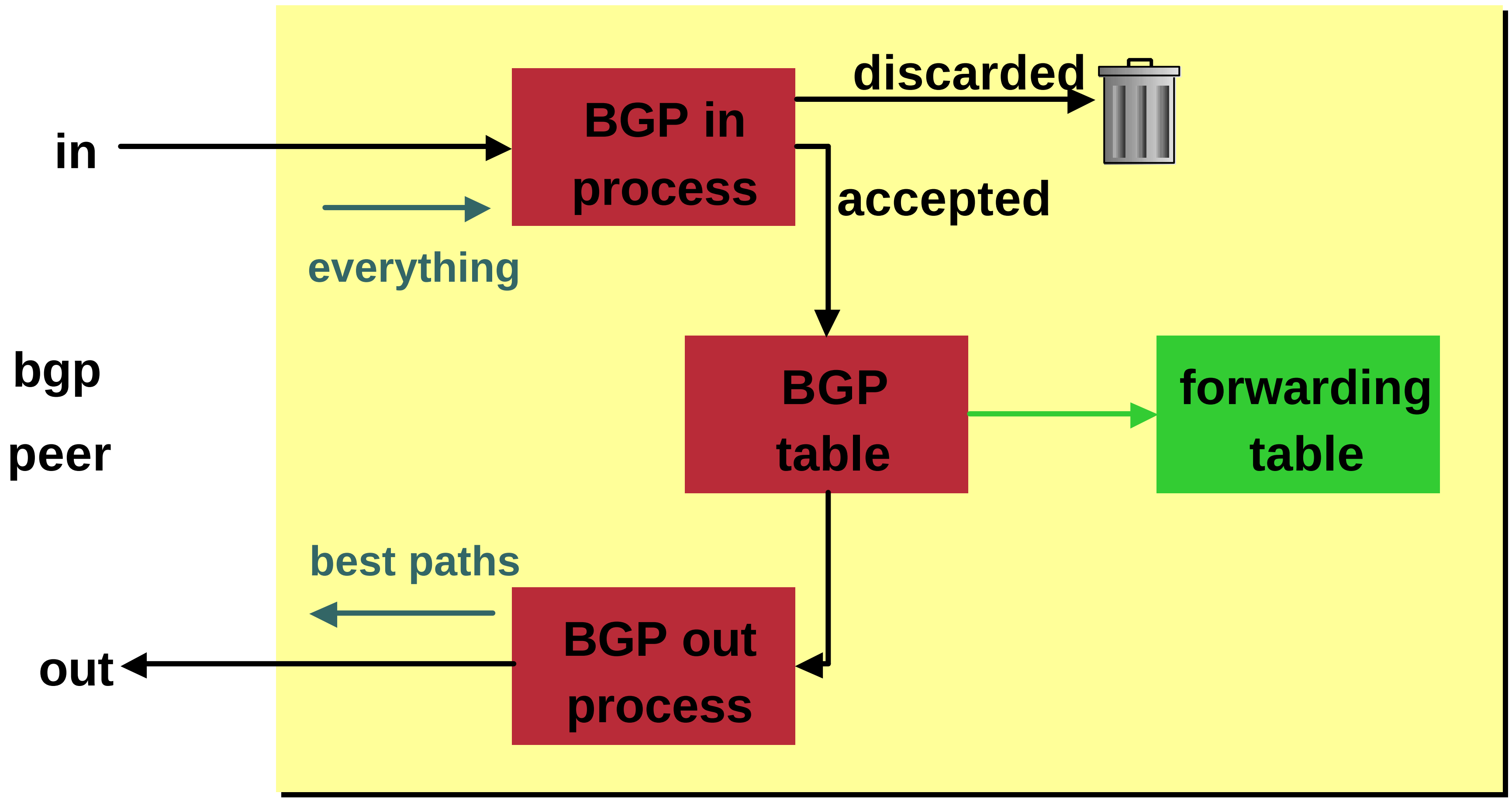
- Learns multiple paths via internal and external BGP speakers
- Picks the best path and installs in the forwarding table
- Policies applied by influencing the best path selection

Constructing the Forwarding Table

Cisco.com

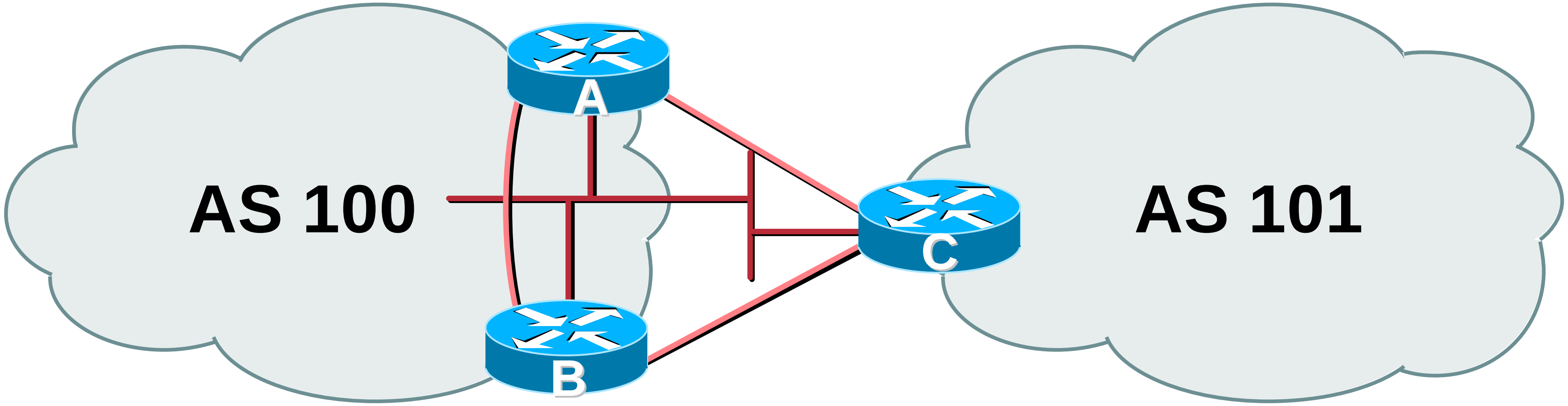
- **BGP "in" process**
  - **receives path information from peers**
  - **results of BGP path selection placed in the BGP table**
  - **"best path" flagged**
- **BGP "out" process**
  - **announces "best path" information to peers**
- **Best paths installed in forwarding table if:**
  - **prefix and prefix length are unique**
  - **lowest "protocol distance"**

Constructing the Forwarding Table



External BGP Peering (eBGP)

Cisco.com



- **Between BGP speakers in different AS**
- **Should be directly connected**
- **Do not run an IGP between eBGP peers**

Configuring External BGP

Router A in AS100

```
interface ethernet 5/0
ip address 222.222.10.2 255.255.255.240
router bgp 100
 network 220.220.8.0 mask 255.255.252.0
 neighbor 222.222.10.1 remote-as 101
 neighbor 222.222.10.1 prefix-list RouterC-in in
 neighbor 222.222.10.1 prefix-list RouterC-out out
```

Router C in AS101

```
interface ethernet 1/0/0
ip address 222.222.10.1 255.255.255.240
router bgp 101
 network 220.220.16.0 mask 255.255.240.0
 neighbor 222.222.10.2 remote-as 100
 neighbor 222.222.10.2 prefix-list RouterA-in in
 neighbor 222.222.10.2 prefix-list RouterA-out out
```

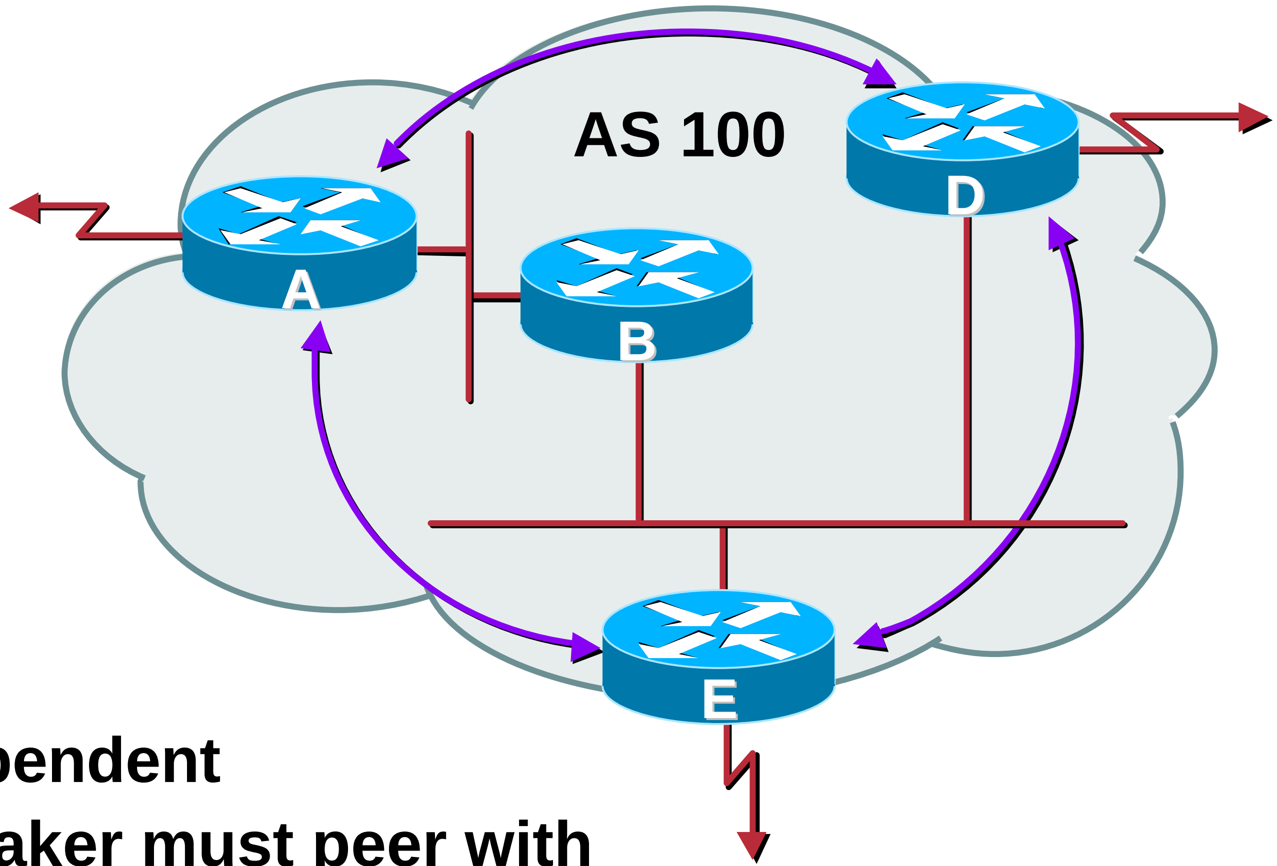
Internal BGP (iBGP)

- BGP peer within the same AS
- Not required to be directly connected
- iBGP speakers need to be fully meshed

  they originate connected networks

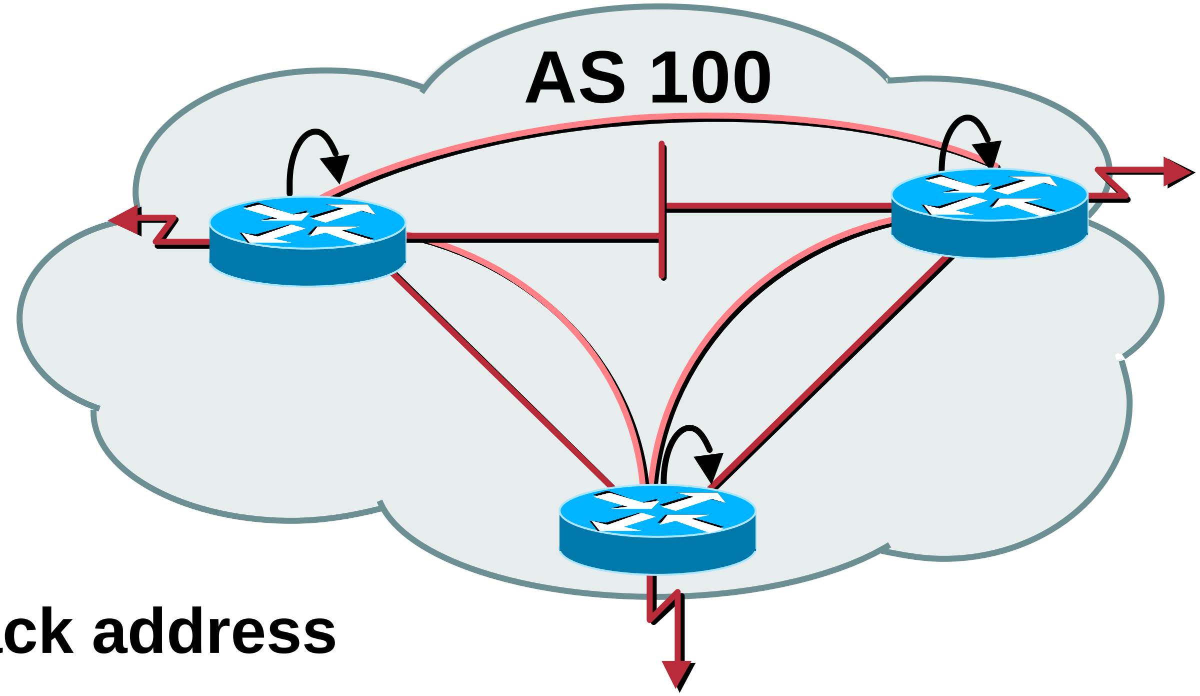
  they do not pass on prefixes learned from other iBGP speakers

Internal BGP Peering (iBGP)



- **Topology independent**
- **Each iBGP speaker must peer with every other iBGP speaker in the AS**

Peering to Loop-back Address



- **Peer with loop-back address**

  Loop-back interface does not go down – ever!

- **iBGP session is not dependent on state of a single interface**
- **iBGP session is not dependent on physical topology**

Configuring Internal BGP

Router A

```
interface loopback 0
ip address 215.10.7.1 255.255.255.255
router bgp 100
  network 220.220.1.0
  neighbor 215.10.7.2 remote-as 100
  neighbor 215.10.7.2 update-source loopback0
  neighbor 215.10.7.3 remote-as 100
  neighbor 215.10.7.3 update-source loopback0
```

Router B

```
interface loopback 0
ip address 215.10.7.2 255.255.255.255
router bgp 100
  network 220.220.5.0
  neighbor 215.10.7.1 remote-as 100
  neighbor 215.10.7.1 update-source loopback0
  neighbor 215.10.7.3 remote-as 100
  neighbor 215.10.7.3 update-source loopback0
```

Inserting prefixes into BGP

Cisco.com

- **Two ways to insert prefixes into BGP**

  redistribute static

  network command

Inserting prefixes into BGP – redistribute static

Cisco.com

- **Configuration Example:**

```
router bgp 100
 redistribute static
ip route 222.10.32.0 255.255.254.0 serial0
```

- **Static route must exist before redistribute command will work**
- **Forces origin to be “incomplete”**
- **Care required!**

Inserting prefixes into BGP – redistribute static

Cisco.com

- Care required with redistribute!

  redistribute <routing-protocol> means everything in the <routing-protocol> will be transferred into the current routing protocol

  Will not scale if uncontrolled

  Best avoided if at all possible

  redistribute normally used with “route-maps” and under tight administrative control

Inserting prefixes into BGP – network command

- **Configuration Example**

```
router bgp 100
 network 222.10.32.0 mask 255.255.254.0
ip route 222.10.32.0 255.255.254.0 serial0
```

- **A matching route must exist in the routing table before the network is announced**
- **Forces origin to be "IGP"**

Configuring Aggregation

- **Three ways to configure route aggregation**

  redistribute static

  aggregate-address

  network command

Configuring Aggregation

Cisco.com

- **Configuration Example:**

```
router bgp 100
 redistribute static
ip route 222.10.0.0 255.255.0.0 null0 250
```

- **static route to “null0” is called a pull up route**

  packets only sent here if there is no more specific match in the routing table

  distance of 250 ensures this is last resort static

  care required – see previously!

Configuring Aggregation – Network Command

Cisco.com

- Configuration Example

```
router bgp 100
 network 222.10.0.0 mask 255.255.0.0
ip route 222.10.0.0 255.255.0.0 null0 250
```

- A matching route must exist in the routing table before the network is announced
- Easiest and best way of generating an aggregate

Configuring Aggregation – aggregate-address command

- **Configuration Example**

```
router bgp 100
 network 222.10.32.0 mask 255.255.252.0
 aggregate-address 222.10.0.0 255.255.0.0 [ summary-only ]
```

- **Requires more specific prefix in routing table before aggregate is announced**
- **{summary-only} keyword**

  optional keyword which ensures that only the summary is announced if a more specific prefix exists in the routing table

Historical Defaults – Auto Summarisation

- Disable historical default 1
- Automatically summarises subprefixes to the classful network when redistributing to BGP from another routing protocol

  Example:

  ```
  61.10.8.0/22 ® 61.0.0.0/8
  ```

- Must be turned off for any Internet connected site using BGP

  ```
  router bgp 100
   no auto-summary
  ```

Historical Defaults – Synchronisation

Cisco.com

- Disable historical default 2
- **In Cisco IOS, BGP does not advertise a route before all routers in the AS have learned it via an IGP**
- **Disable synchronisation if:**

  AS doesn't pass traffic from one AS to another, or

  All transit routers in AS run BGP, or

  iBGP is used across backbone

```
router bgp 100
 no synchronization
```

Summary

Cisco.com

- BGP4 – path vector protocol
- iBGP versus eBGP
- stable iBGP – peer with loopbacks
- announcing prefixes & aggregates
- no synchronization & no auto-summary

Introduction to BGP

BGP Attributes and Policy Control

Agenda

- **BGP Attributes**
- **BGP Path Selection**
- **Applying Policy**

BGP Attributes

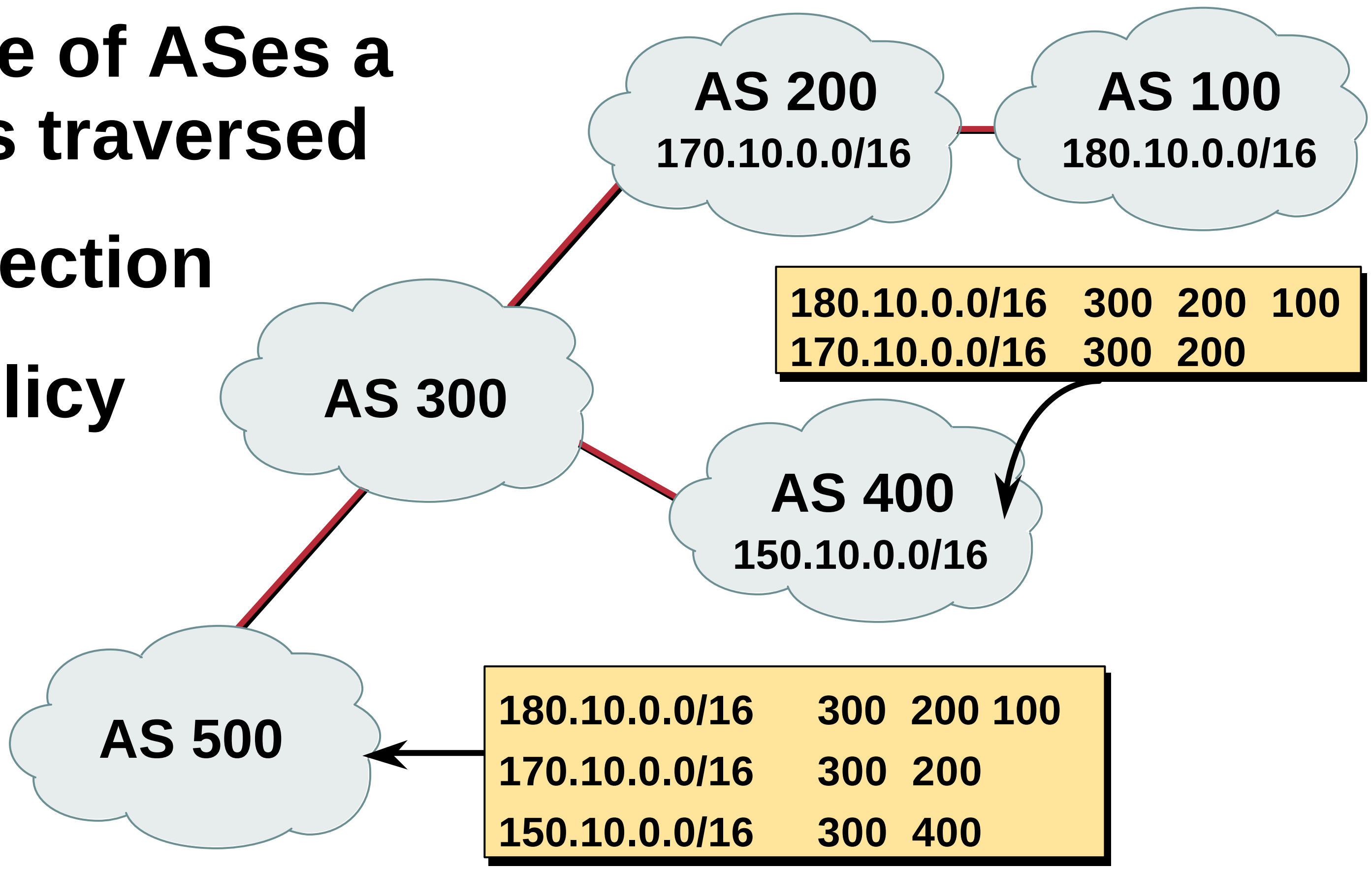
The “tools” available for the job

What Is an Attribute?

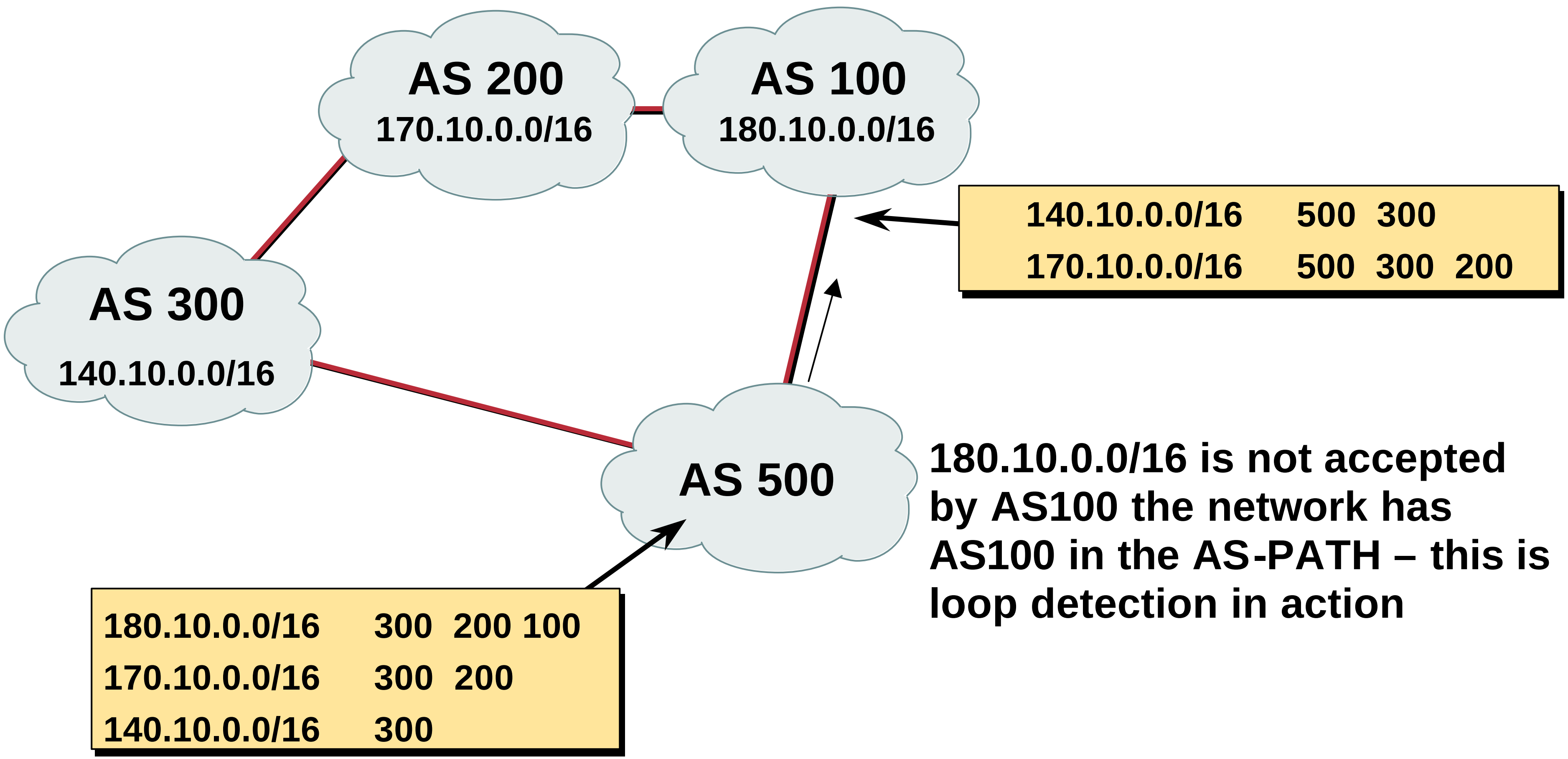
- Describes the characteristics of prefix
- Transitive or non-transitive
- Some are mandatory

AS-Path

- Sequence of ASes a route has traversed
- Loop detection
- Apply policy



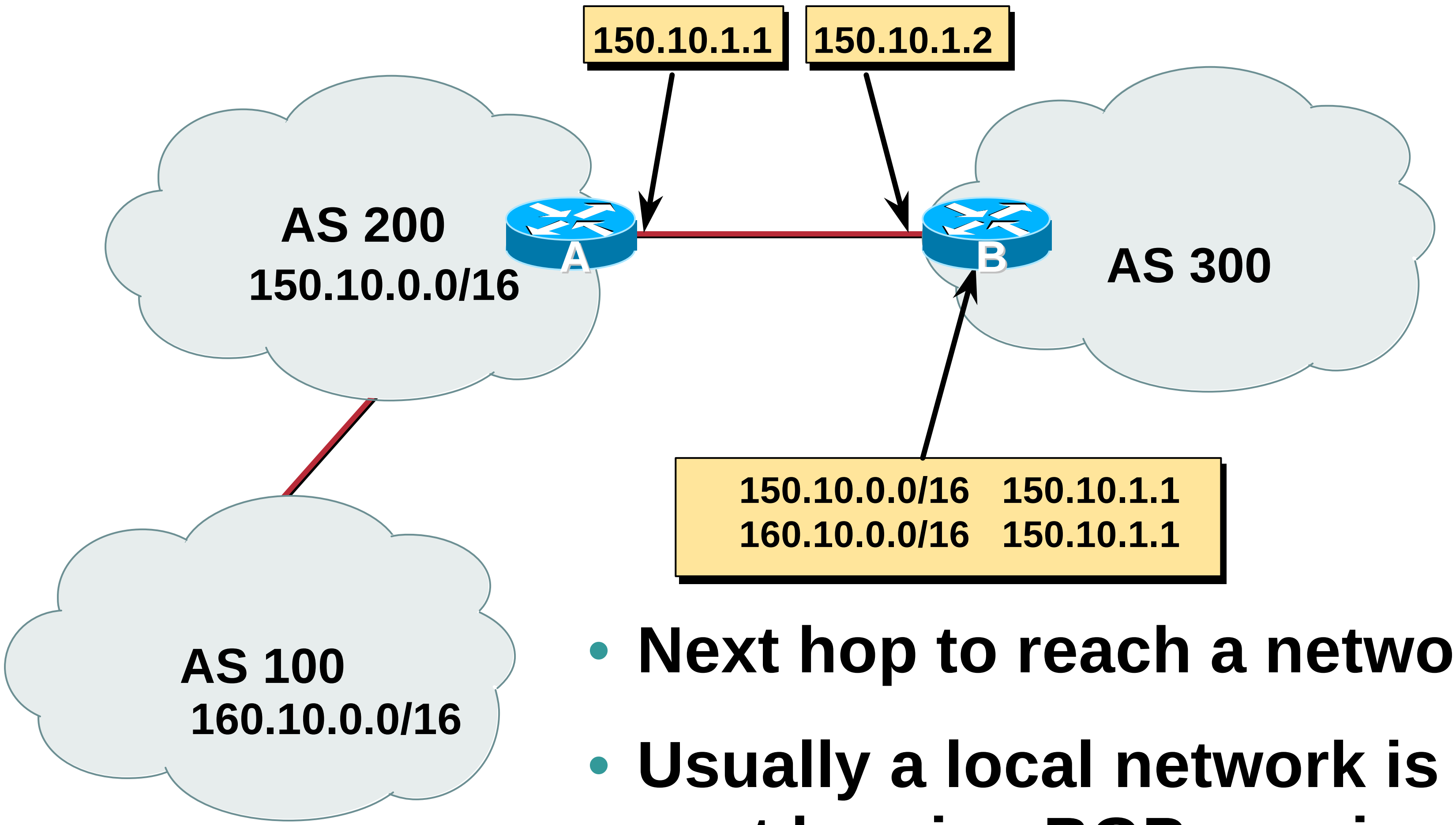
AS-Path loop detection



180.10.0.0/16 is not accepted by AS100 the network has AS100 in the AS-PATH – this is loop detection in action

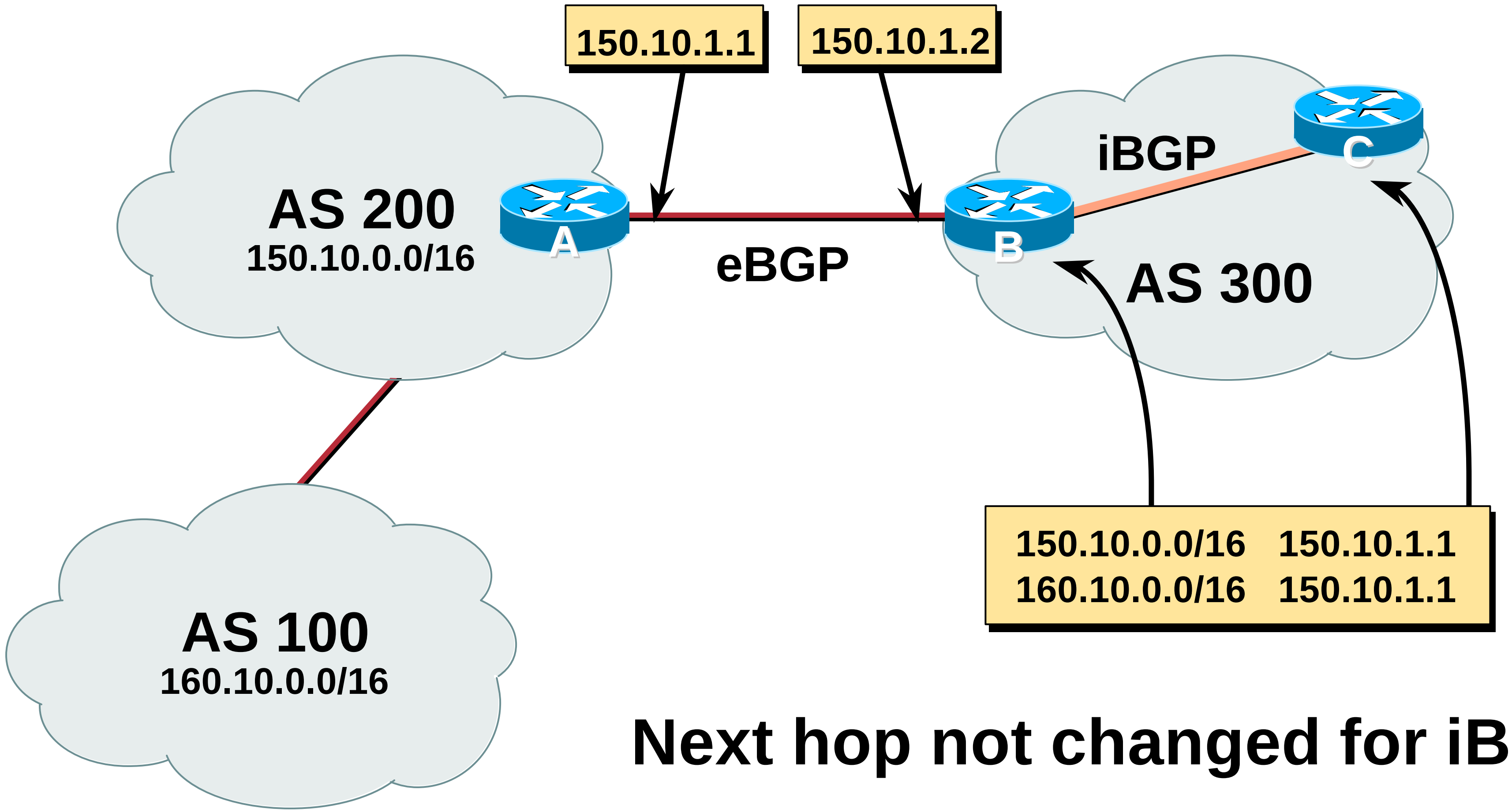
Next Hop

Cisco.com



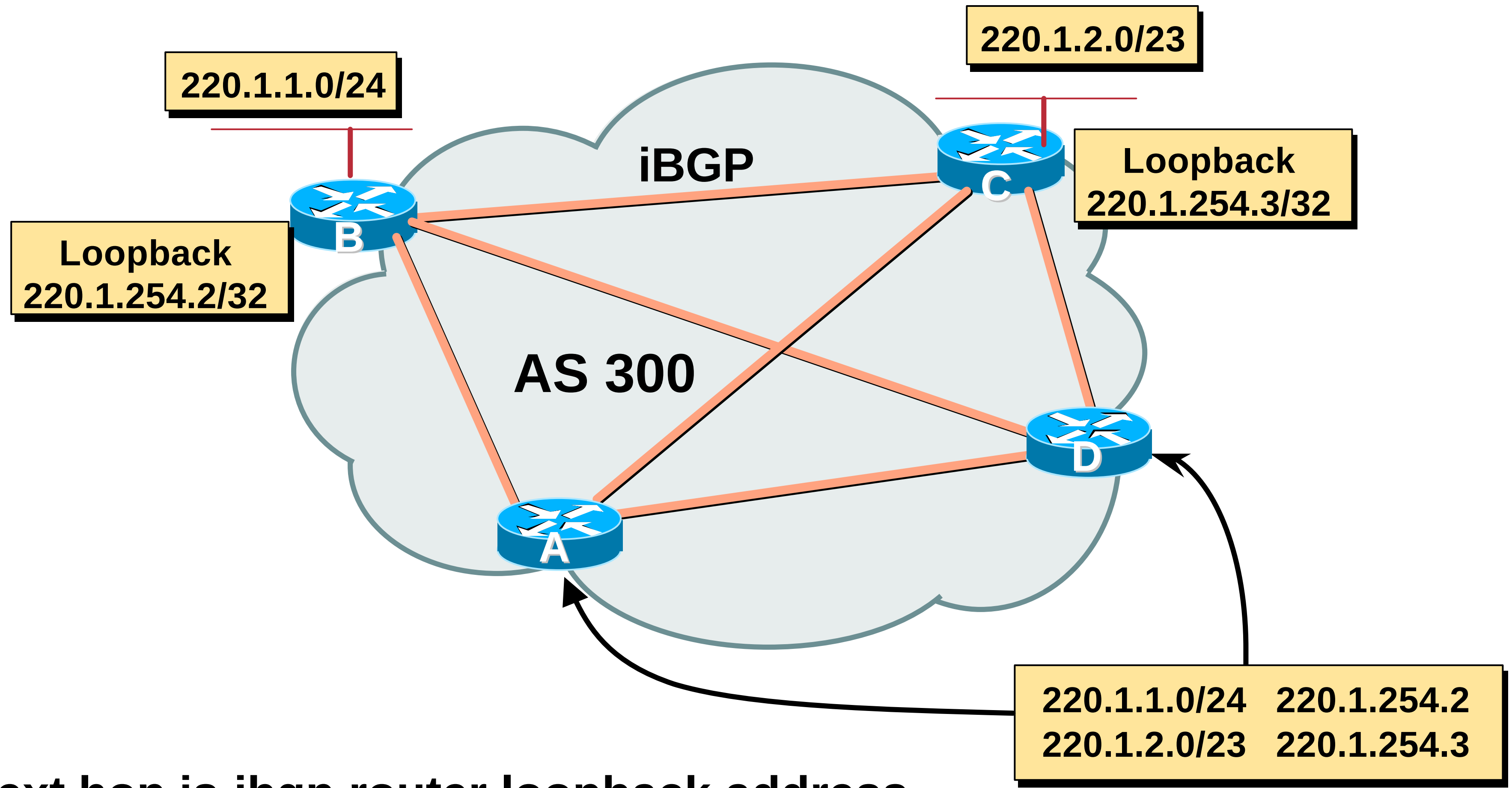
- Next hop to reach a network
- Usually a local network is the next hop in eBGP session

Next Hop



Next hop not changed for iBGP

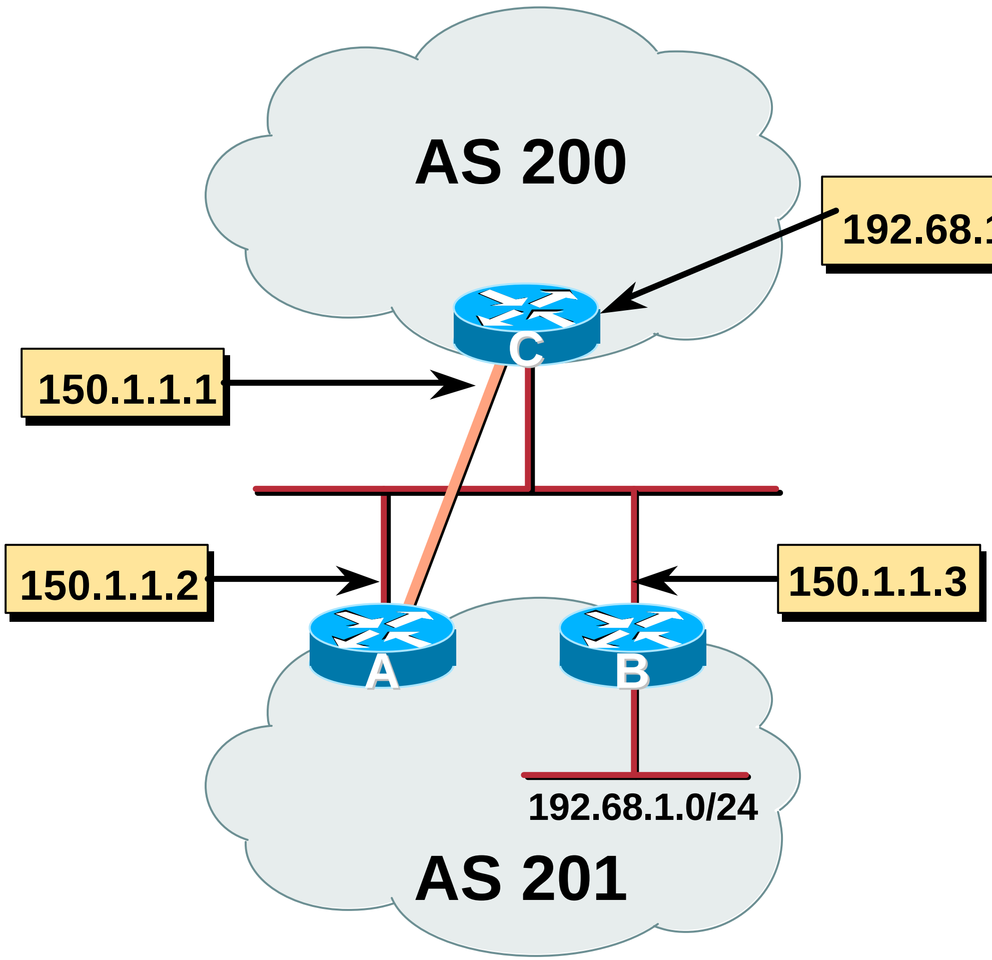
iBGP Next Hop



Next hop is ibgp router loopback address

Recursive route look-up

Third Party Next Hop



- **eBGP between Router A and Router C**
- **iBGP between RouterA and RouterB**
- **192.68.1/24 prefix has next hop address of 150.1.1.3 – this is passed on to RouterC instead of 150.1.1.2**
- **More efficient**
- **No extra config needed**

Next Hop (summary)

Cisco.com

- IGP should carry route to next hops
- Recursive route look-up
- Unlinks BGP from actual physical topology
- Allows IGP to make intelligent forwarding decision

Origin

Cisco.com

- Conveys the origin of the prefix
- Influence best path selection
- Three values – IGP, EGP, incomplete

  IGP – generated from BGP network statement

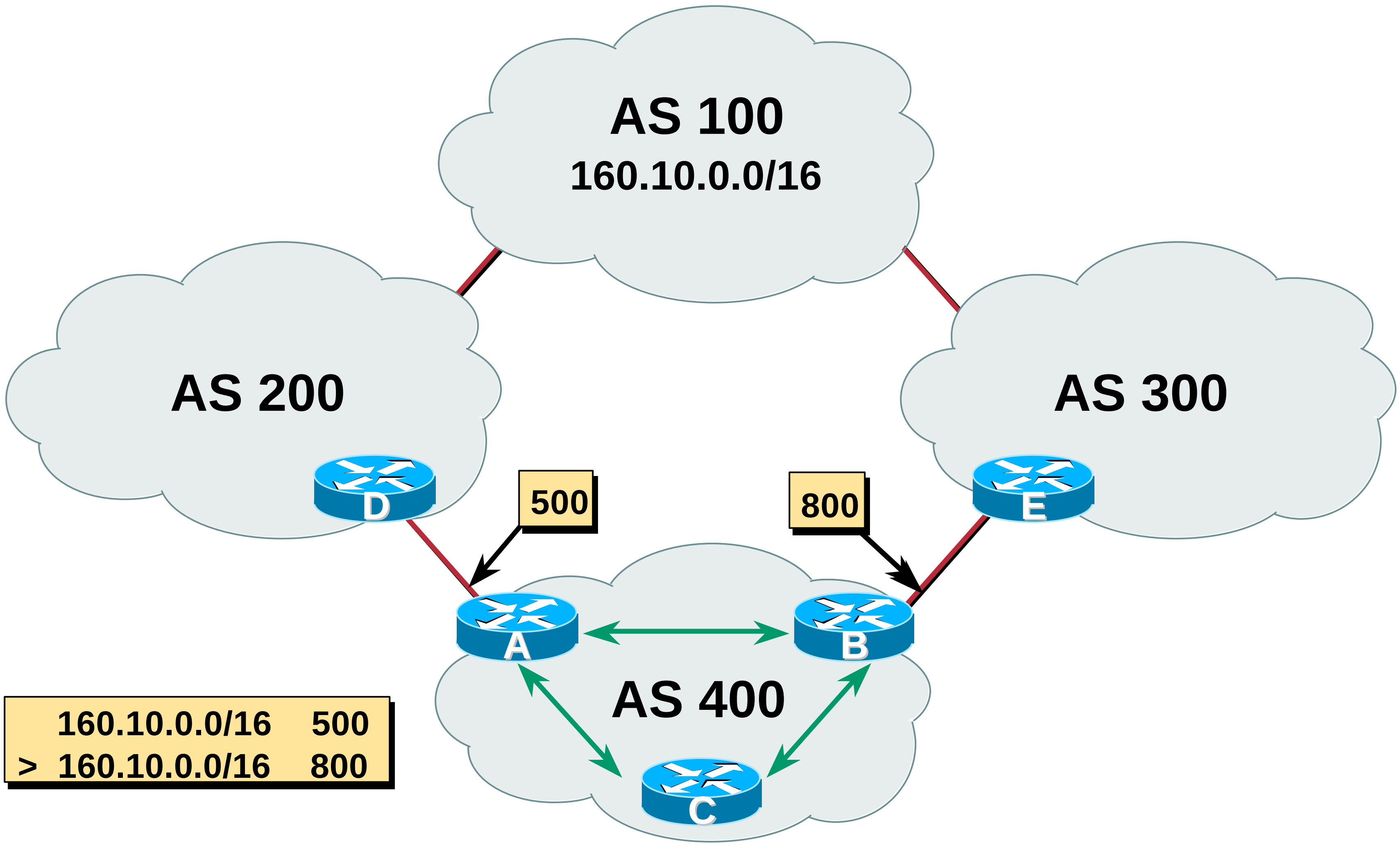
  EGP – generated from EGP

  incomplete – generated by “redistribute” action

Aggregator

- Useful for debugging purposes
- Conveys the IP address of the router/BGP speaker generating the aggregate route
- Does not influence path selection

Local Preference



Local Preference

- Local to an AS – non-transitive

  local preference set to 100 when heard from neighbouring AS

- Used to influence BGP path selection

  determines best path for *outbound* traffic

- Path with highest local preference wins

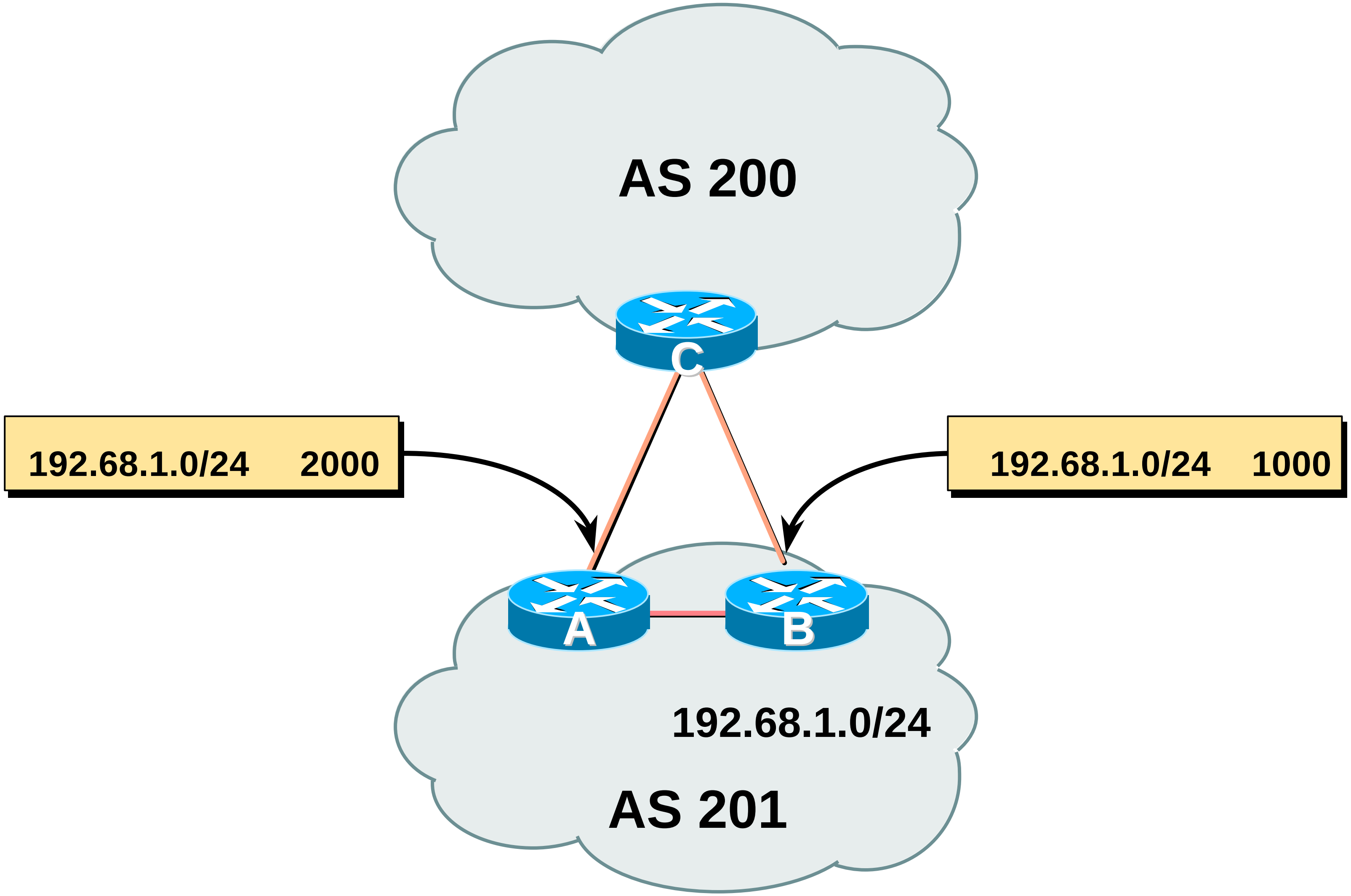
Local Preference

- **Configuration of Router B:**

```
router bgp 400
 neighbor 220.5.1.1 remote-as 300
 neighbor 220.5.1.1 route-map local-pref in
!
route-map local-pref permit 10
 match ip address prefix-list MATCH
 set local-preference 800
!
ip prefix-list MATCH permit 160.10.0.0/16
ip prefix-list MATCH deny 0.0.0.0/0 le 32
```

Multi-Exit Discriminator (MED)

Cisco.com



Multi-Exit Discriminator

- **Inter-AS – non-transitive**
  - **metric attribute not announced to next AS**
- **Used to convey the relative preference of entry points**
  - **determines best path for *inbound* traffic**
- **Comparable if paths are from same AS**
- **IGP metric can be conveyed as MED**
  - **set metric-type internal in route-map**

MED & IGP Metric

- **set metric-type internal**

  enable BGP to advertise a MED which corresponds to the IGP metric values

  changes are monitored (and re-advertised if needed) every 600s

  bgp dynamic-med-interval <secs>

Multi-Exit Discriminator

- **Configuration of Router B:**

```
router bgp 400
 neighbor 220.5.1.1 remote-as 200
 neighbor 220.5.1.1 route-map set-med out
!
route-map set-med permit 10
 match ip address prefix-list MATCH
 set metric 1000
!
ip prefix-list MATCH permit 192.68.1.0/24
ip prefix-list MATCH deny 0.0.0.0/0 le 32
```

Weight

- Not really an attribute – local to router
- Highest weight wins
- Applied to all routes from a neighbour

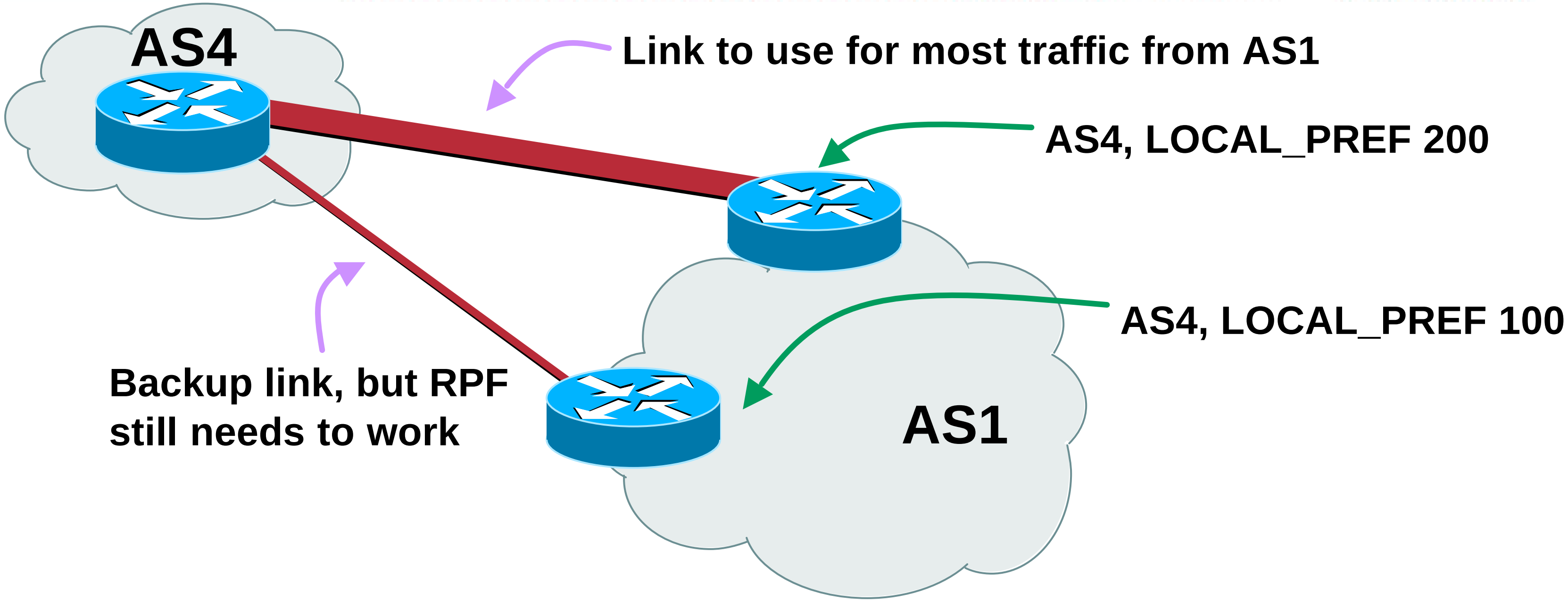
```
neighbor 220.5.7.1 weight 100
```

- Weight assigned to routes based on filter

```
neighbor 220.5.7.3 filter-list 3 weight 50
```

Weight – Used to Deploy RPF

Cisco.com



- Local to router on which it's configured

  Not really an attribute

- route-map: *set weight*
- Highest weight wins over all valid paths
- Weight customer eBGP on edge routers to allow RPF to work correctly

Community

Cisco.com

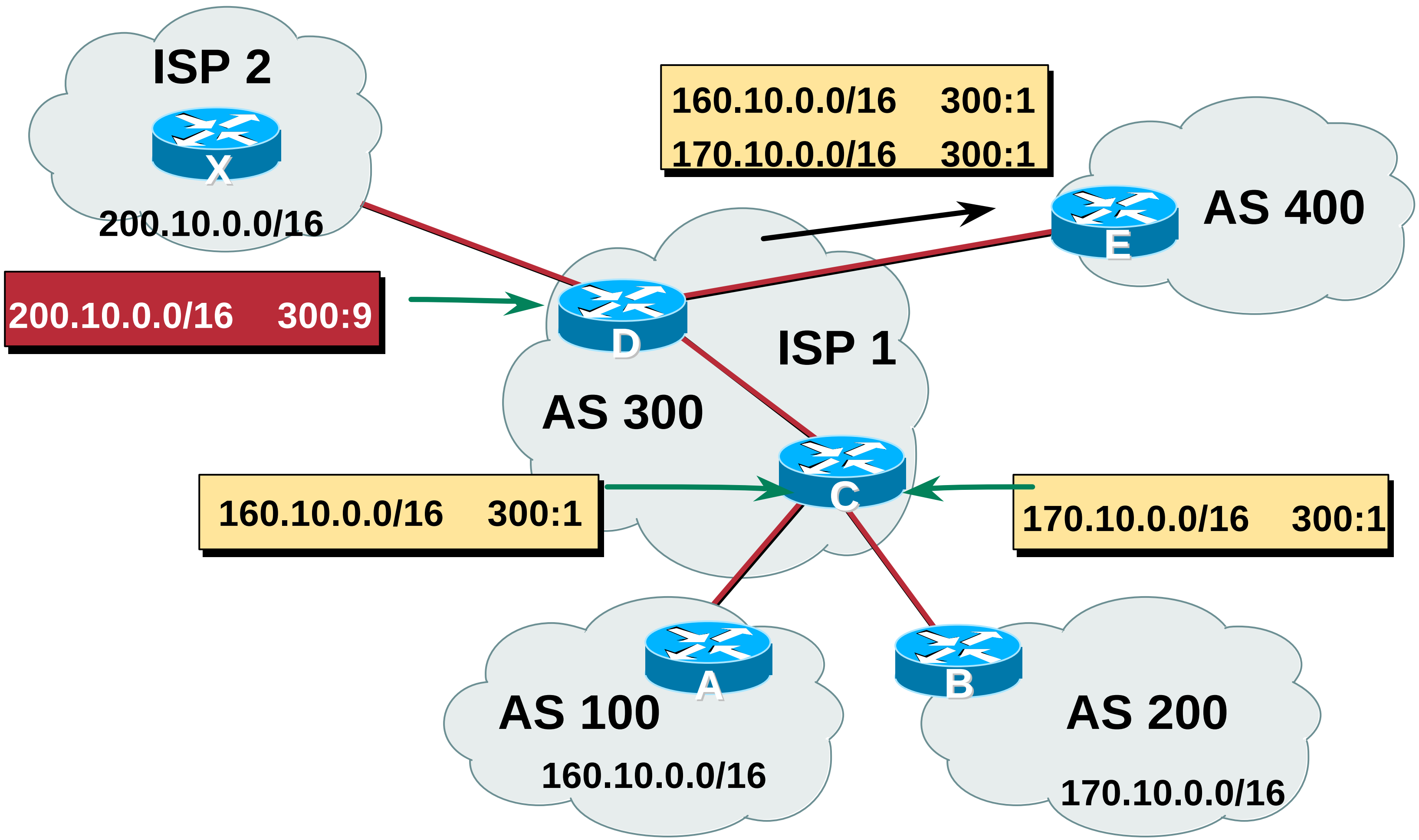
- **Communities described in RFC1997**
- **32 bit integer**

  Commonly represented as two 16 bit integers (RFC1998)
- **Used to group destinations**

  Each destination could be member of multiple communities
- **Community attribute carried across AS's**
- **Very useful in applying policies**

Community

Cisco.com



Well-Known Communities

- **no-export**

  do not advertise to eBGP peers

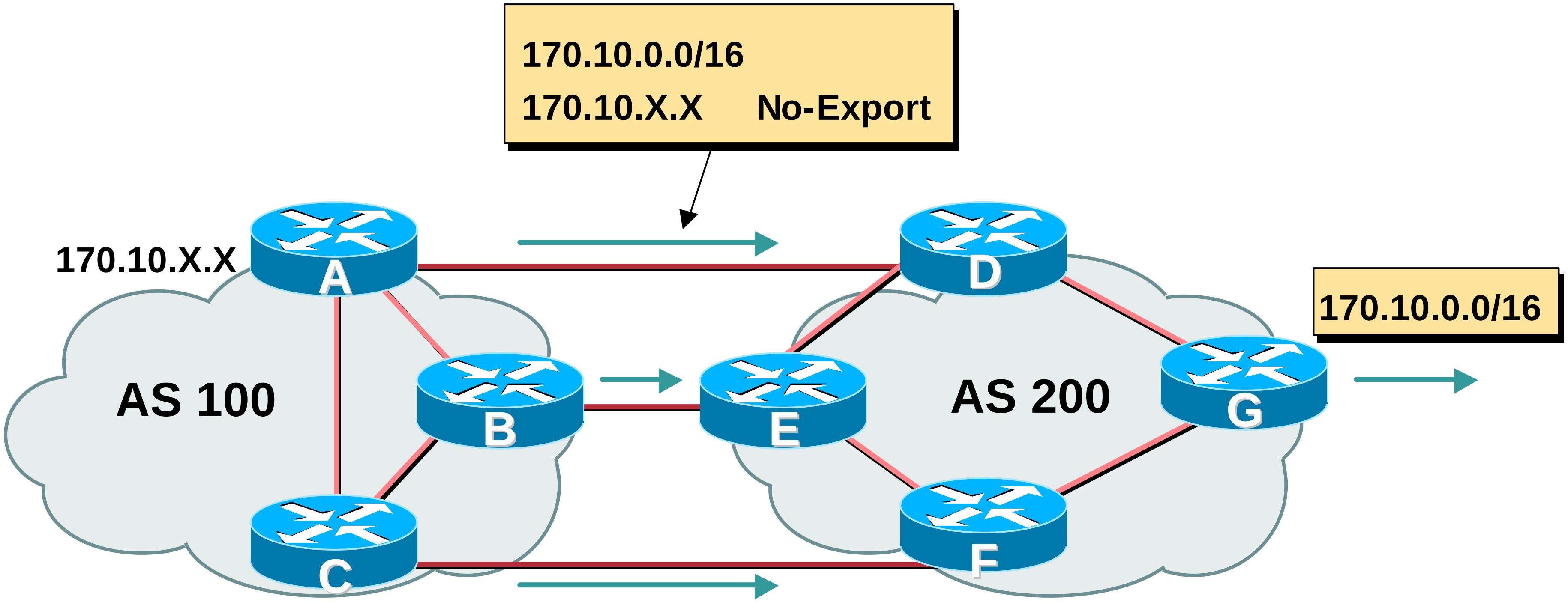
- **no-advertise**

  do not advertise to any peer

- **local-AS**

  do not advertise outside local AS (only used with confederations)

No-Export Community



No-Export Community

- **AS100 announces aggregate and subprefixes**

  aim is to improve loadsharing between AS100 and AS200 by leaking subprefixes

- **Subprefixes marked with no-export community**
- **Router G in AS200 strips out all prefixes with no-export community set**

BGP Path Selection Algorithm

Why is this the best path?

BGP Path Selection Algorithm

Cisco.com

- Do not consider path if no route to next hop
- Do not consider iBGP path if not synchronised
- Highest weight (local to router)
- Highest local preference (global within AS)
- Prefer locally originated route
- Shortest AS path

BGP Path Selection Algorithm (continued)

Cisco.com

- **Lowest origin code**

  IGP < EGP < incomplete

- **Lowest Multi-Exit Discriminator (MED)**

  If bgp deterministic-med, order the paths before comparing

  If bgp always-compare-med, then compare for all paths

  otherwise MED only considered if paths are from the same AS (default)

BGP Path Selection Algorithm (continued)

Cisco.com

- **Prefer eBGP path over iBGP path**
- **Path with lowest IGP metric to next-hop**
- **For eBGP paths:**

  If multipath is enabled, install N parallel paths in forwarding table

  If router-id is the same, go to next step

  If router-id is not the same, select the oldest path

BGP Path Selection Algorithm (continued)

Cisco.com

- **Lowest router-id (originator-id for reflected routes)**
- **Shortest cluster-list**

  Client must be aware of Route Reflector attributes!
- **Lowest neighbour address**

Applying Policy with BGP

How to use the "tools"

Applying Policy with BGP

Cisco.com

- Policy-based on AS path, community or the prefix
- Rejecting/accepting selected routes
- Set attributes to influence path selection
- Tools:
  - Prefix-list (filters prefixes)
  - Filter-list (filters ASes)
  - Route-maps and communities

Policy Control – Prefix List

- Per neighbour prefix filter

  incremental configuration

- High performance access-list
- Inbound or Outbound
- Based upon network numbers (using familiar IPv4 address/mask format)

Prefix-list Command

[no] ip prefix-list <*list-name***> [seq <***seq-value***>] deny | permit <***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***>**

Prefix Lists – 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 Lists – 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

Policy Control – Prefix List

Cisco.com

- **Example Configuration**

```
router bgp 200
 network 215.7.0.0
 neighbor 220.200.1.1 remote-as 210
 neighbor 220.200.1.1 prefix-list PEER-IN in
 neighbor 220.200.1.1 prefix-list PEER-OUT out
!
ip prefix-list PEER-IN deny 218.10.0.0/16
ip prefix-list PEER-IN permit 0.0.0.0/0 le 32
ip prefix-list PEER-OUT permit 215.7.0.0/16
ip prefix-list PEER-OUT deny 0.0.0.0/0 le 32
```

Policy Control – Filter List

- Filter routes based on AS path
- Inbound or Outbound
- Example Configuration:

```
router bgp 100
 network 215.7.0.0
 neighbor 220.200.1.1 filter-list 5 out
 neighbor 220.200.1.1 filter-list 6 in
!
ip as-path access-list 5 permit ^200$
ip as-path access-list 6 permit ^150$
```

Policy Control – Regular Expressions

- **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**
_	**Beginning, end, white-space, brace**
|	**Or**
()	**brackets to contain expression**

Policy Control – Regular Expressions

- **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)**

Policy Control – Regular Expressions

Cisco.com

- **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

Policy Control – Route Maps

Cisco.com

- **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 Maps – Caveats

- Lines can have multiple set statements but only one match statement
- 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 go through

    the rest are dropped

Route Maps – Caveats

- **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
```

Policy Control – Route Maps

- **Example Configuration – route map and prefix-lists**

```
router bgp 100
 neighbor 1.1.1.1 route-map infilter in
!
route-map infilter permit 10
 match ip address prefix-list HIGH-PREF
 set local-preference 120
!
route-map infilter permit 20
 match ip address prefix-list LOW-PREF
 set local-preference 80
!
ip prefix-list HIGH-PREF permit 10.0.0.0/8
ip prefix-list LOW-PREF permit 20.0.0.0/8
```

Policy Control – Route Maps

- **Example Configuration – route map and filter lists**

```
router bgp 100
 neighbor 220.200.1.2 remote-as 200
 neighbor 220.200.1.2 route-map filter-on-as-path in
!
route-map filter-on-as-path permit 10
 match as-path 1
 set local-preference 80
!
route-map filter-on-as-path permit 20
 match as-path 2
 set local-preference 200
!
ip as-path access-list 1 permit _150$
ip as-path access-list 2 permit _210_
```

Policy Control – Route Maps

- Example configuration of AS-PATH prepend

```
router bgp 300
 network 215.7.0.0
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 route-map SETPATH out
!
route-map SETPATH permit 10
 set as-path prepend 300 300
```

- Use your own AS number when prepending

  Otherwise BGP loop detection may cause disconnects

Policy Control – Route Maps

- **Route Map MATCH Articles**

as-path	ip next-hop
clns address	ip route-source
clns next-hop	length
clns route-source	metric
community	nlri
interface	route-type
ip address	tag

Policy Control – Route Maps

- **Route map SET Articles**

  as-path
  automatic-tag
  clns
  comm-list
  community
  dampening
  default interface
  interface
  ip default next-hop
  ip next-hop

Policy Control – Route Maps

- Route map SET Articles

ip precedence	next-hop
ip qos-group	nlri multicast
ip tos	nlri unicast
level	origin
local preference	tag
metric	traffic-index
metric-type	weight

Policy Control – Matching Communities

- Example Configuration

```
router bgp 100
 neighbor 220.200.1.2 remote-as 200
 neighbor 220.200.1.2 route-map filter-on-community in
!
route-map filter-on-community permit 10
 match community 1
 set local-preference 50
!
route-map filter-on-community permit 20
 match community 2 exact-match
 set local-preference 200
!
ip community-list 1 permit 150:3 200:5
ip community-list 2 permit 88:6
```

Policy Control – Setting Communities

- Example Configuration

```
router bgp 100
 network 215.7.0.0
 neighbor 220.200.1.1 remote-as 200
 neighbor 220.200.1.1 send-community
 neighbor 220.200.1.1 route-map set-community out
!
route-map set-community permit 10
 match ip address prefix-list NO-ANNOUNCE
 set community no-export
!
route-map set-community permit 20
 match ip address prefix-list EVERYTHING
!
ip prefix-list NO-ANNOUNCE permit 172.168.0.0/16 ge 17
ip prefix-list EVERYTHING permit 0.0.0.0/0 le 32
```

Aggregation Policies

- **Suppress Map**

  Used to suppress selected more-specific prefixes (e.g. defined through a route-map) in the absence of the summary-only keyword.

- **Unsuppress Map**

  Used to unsuppress selected more-specific prefixes per BGP peering when the summary-only keyword is in use.

Aggregation Policies – Suppress Map

- **Example**

```
router bgp 100
 network 220.10.10.0
 network 220.10.11.0
 network 220.10.12.0
 network 220.10.33.0
 network 220.10.34.0
 aggregate-address 220.10.0.0 255.255.0.0 suppress-map block-net
 neighbor 222.5.7.2 remote-as 200
!
route-map block-net permit 10
 match ip address prefix-list SUPPRESS
!
ip prefix-list SUPPRESS permit 220.10.8.0/21 le 32
ip prefix-list SUPPRESS deny 0.0.0.0/0 le 32
!
```

Aggregation Policies – Suppress Map

- **show ip bgp** on the local router

```
router1#sh ip bgp
BGP table version is 11, local router ID is 222.5.7.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network           Next Hop          Metric LocPrf Weight Path
*> 220.10.0.0/16  0.0.0.0                     32768 i
s> 220.10.10.0    0.0.0.0             0              32768 i
s> 220.10.11.0    0.0.0.0             0              32768 i
s> 220.10.12.0    0.0.0.0             0              32768 i
*> 220.10.33.0    0.0.0.0             0              32768 i
*> 220.10.34.0    0.0.0.0             0              32768 i
```

Aggregation Policies – Suppress Map

- **show ip bgp** on the remote router

```
router2#sh ip bgp
BGP table version is 90, local router ID is 222.5.7.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network             Next Hop       Metric LocPrf Weight Path
*> 220.10.0.0/16    222.5.7.1                       0  100 i
*> 220.10.33.0      222.5.7.1         0             0  100 i
*> 220.10.34.0      222.5.7.1         0             0  100 i
```

Aggregation Policies – Unsuppress Map

Cisco.com

- **Example**

```
router bgp 100
 network 220.10.10.0
 network 220.10.11.0
 network 220.10.12.0
 network 220.10.33.0
 network 220.10.34.0
 aggregate-address 220.10.0.0 255.255.0.0 summary-only
 neighbor 222.5.7.2 remote-as 200
 neighbor 222.5.7.2 unsuppress-map leak-net
!
route-map leak-net permit 10
 match ip address prefix-list LEAK
!
ip prefix-list LEAK permit 220.10.8.0/21 le 32
ip prefix-list LEAK deny 0.0.0.0/0 le 32
!
```

Aggregation Policies – Unsuppress Map

- **show ip bgp** on the local router

```
router1#sh ip bgp
BGP table version is 11, local router ID is 222.5.7.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network           Next Hop          Metric LocPrf Weight Path
*> 220.10.0.0/16  0.0.0.0                     32768 i
s> 220.10.10.0    0.0.0.0              0             32768 i
s> 220.10.11.0    0.0.0.0              0             32768 i
s> 220.10.12.0    0.0.0.0              0             32768 i
s> 220.10.33.0    0.0.0.0              0             32768 i
s> 220.10.34.0    0.0.0.0              0             32768 i
```

Aggregation Policies – Unsuppress Map

- **show ip bgp** on the remote router

```
router2#sh ip bgp
BGP table version is 90, local router ID is 222.5.7.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network             Next Hop       Metric LocPrf Weight Path
*> 220.10.0.0/16    222.5.7.1                         0  100 i
*> 220.10.10.0      222.5.7.1         0               0  100 i
*> 220.10.11.0      222.5.7.1         0               0  100 i
*> 220.10.12.0      222.5.7.1         0               0  100 i
```

Aggregation Policies – Aggregate Address

- **Summary-only used**

  all subprefixes suppressed

  unsuppress-map to selectively leak subprefixes

  bgp per neighbour configuration

- **Absence of summary-only**

  no subprefixes suppressed

  suppress-map to selectively suppress subprefixes

  bgp global configuration

BGP Attributes and Policy Control

BGP Communities

Problem: Scale Routing Policy Solution: COMMUNITY

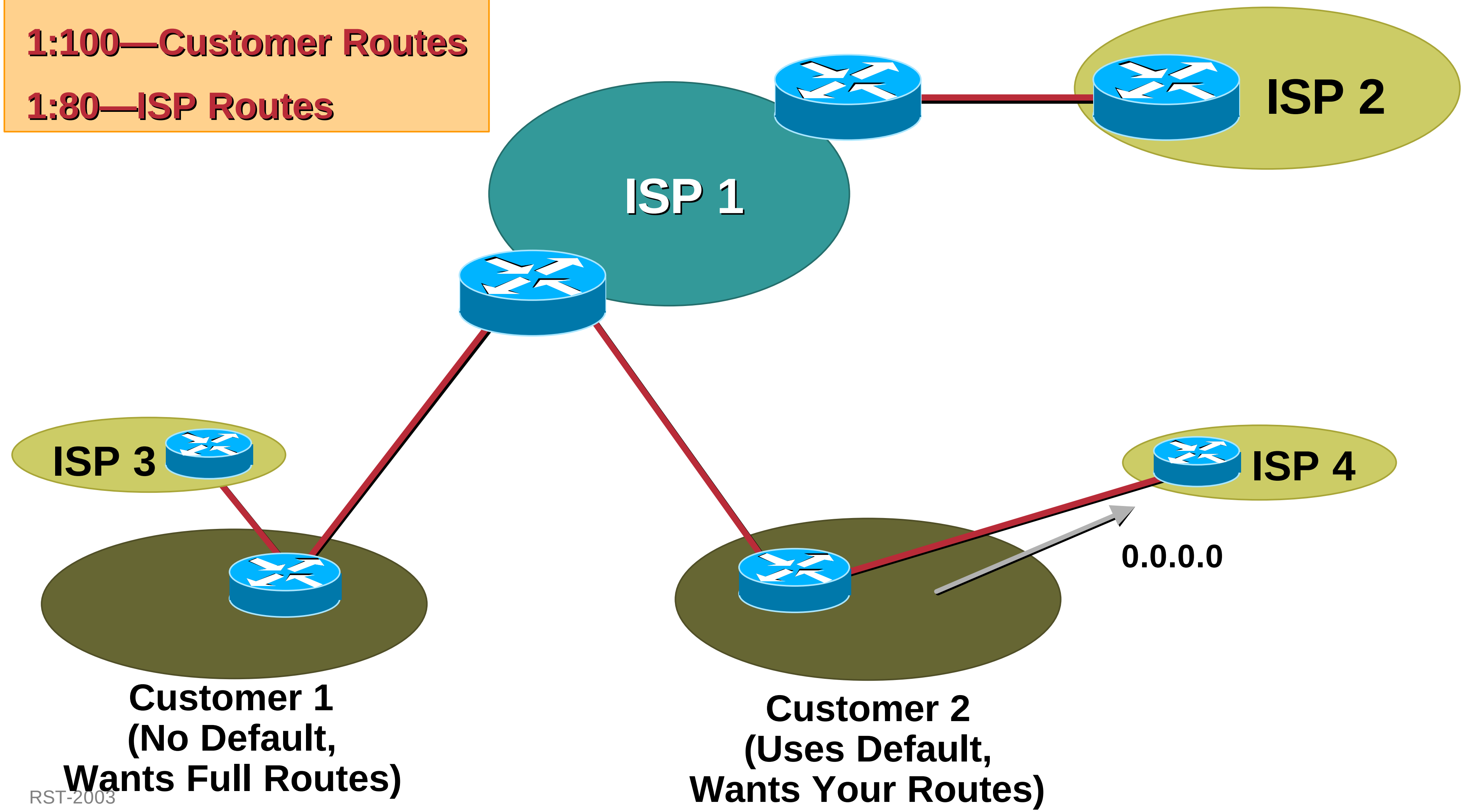
- NOT in decision algorithm
- BGP route can be a member of many communities
- Typical communities:
  - Destinations learned from customers
  - Destinations learned from ISPs or peers
  - Destinations in VPN—BGP community is fundamental to the operation of BGP VPNs

Problem: Scale Routing Policy Solution: COMMUNITY

Communities:

1:100—Customer Routes

1:80—ISP Routes

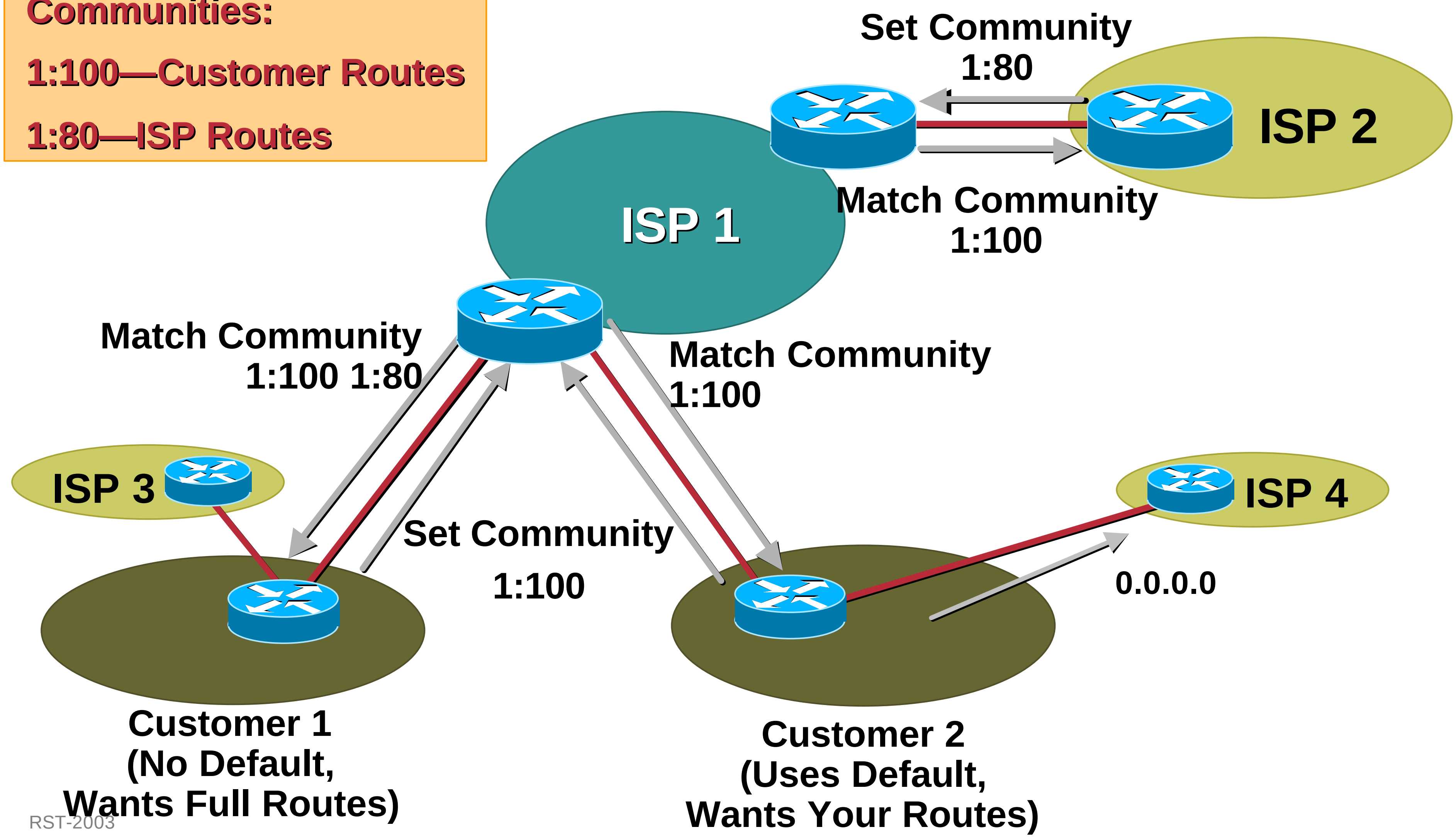


Problem: Scale Routing Policy
Solution: COMMUNITY

Communities:

1:100—Customer Routes

1:80—ISP Routes



BGP Attributes: COMMUNITY

- **Activated per neighbor/peer-group:**

  neighbor {peer-address | peer-group-name} send-community

- **Carried across AS boundaries**
- **Common convention is string of four bytes: <AS>:[0-65536]**

BGP Attributes: COMMUNITY (Cont.)

Cisco.com

- **Each destination can be a member of multiple communities**
- **Using a route-map: *set community***

<1-4294967295> community number

aa:nn community number in aa:nn format

additive Add to the existing community

none No community attribute

local-AS Do not send to EBGP peers (well-known community)

no-advertise Do not advertise to any peer (well-known community)

no-export Do not export outside AS/confed (well-known community)

Community Filters

- **Filter based on Community Strings**
  - **ip community-list <*1-99*> [permit|deny] *comm***
  - **ip community-list <*100-199*> [permit|deny] *regexp***
- **Per neighbor**
  - **Inbound or outbound route-maps**
    - **Match community <number> [exact-match]**
    - **Exact match only for standard lists**

Community Filters

- **Example 1:**

  Mark some prefixes as part of the 1:120 community (+remove existing community!)

- **Configuration:**

```
router bgp 1
neighbor 10.0.0.1 remote-as 2
neighbor 10.0.0.1 send-community
neighbor 10.0.0.1 route-map set_community out
!
route-map set_community 10 permit
match ip address 1
set community 1:120
!
access-list 1 permit 10.10.0.0 0.0.255.255
```

Community Filters

- **Example 2:**

  Set LOCAL_PREF depending on the community that the prefix belongs to

- **Configuration:**

```
router bgp 1
neighbor 10.0.0.1 remote-as 2
neighbor 10.0.0.1 route-map filter_on_community in
!
route-map filter_on_community 10 permit
match community 1
set local-preference 150
!
ip community-list 1 permit 2:150
```

Deploying iBGP

Guidelines for Stable IBGP

- **Peer using loopback addresses**

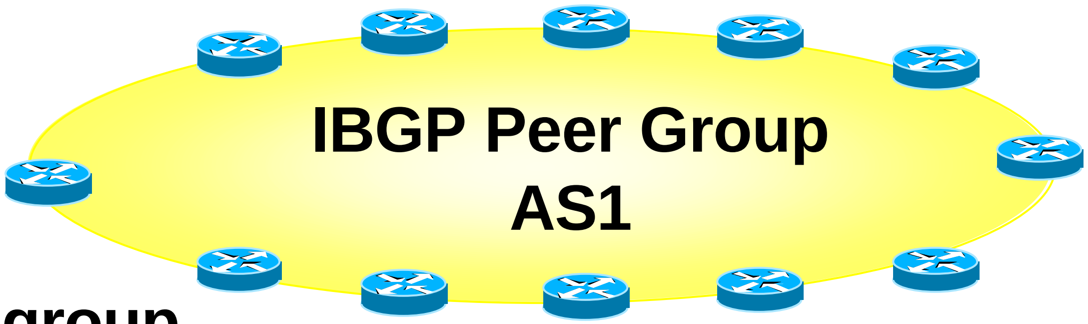
  neighbor { ip address | peer-group}
  update-source loopback0

- **Independent of physical interface failure**
- **IGP performs any load-sharing**

Guidelines for Scaling IBGP

- Use peer groups and RRs
- Carry only next-hops in IGP
- Carry full routes in BGP only if necessary
- Do not redistribute BGP into IGP

Using Peer Groups



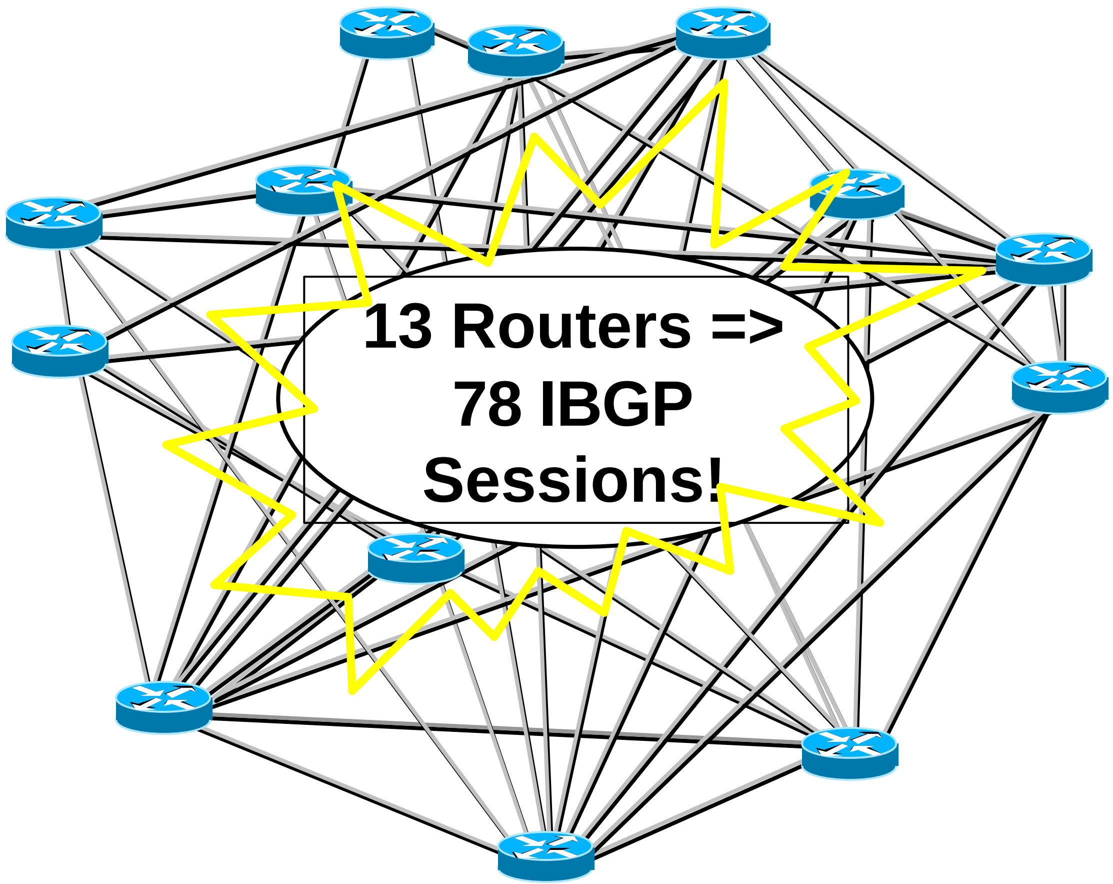

```
router bgp  1
neighbor internal peer-group
neighbor internal description ibgp peers
neighbor internal remote-as 1
neighbor internal update-source Loopback0
neighbor internal next-hop-self
neighbor internal send-community
neighbor internal version 4
neighbor internal password 7 03085A09
neighbor 1.0.0.1 peer-group internal
neighbor 1.0.0.2 peer-group internal
```

What Is a Peer Group?

- **All peer-group members have a common outbound policy**
- **Updates generated once per peer group**
- **Simplifies configuration**
- **Members can have different inbound policy**

Why Route Reflectors?

Avoid n(n-1)/2 IBGP mesh

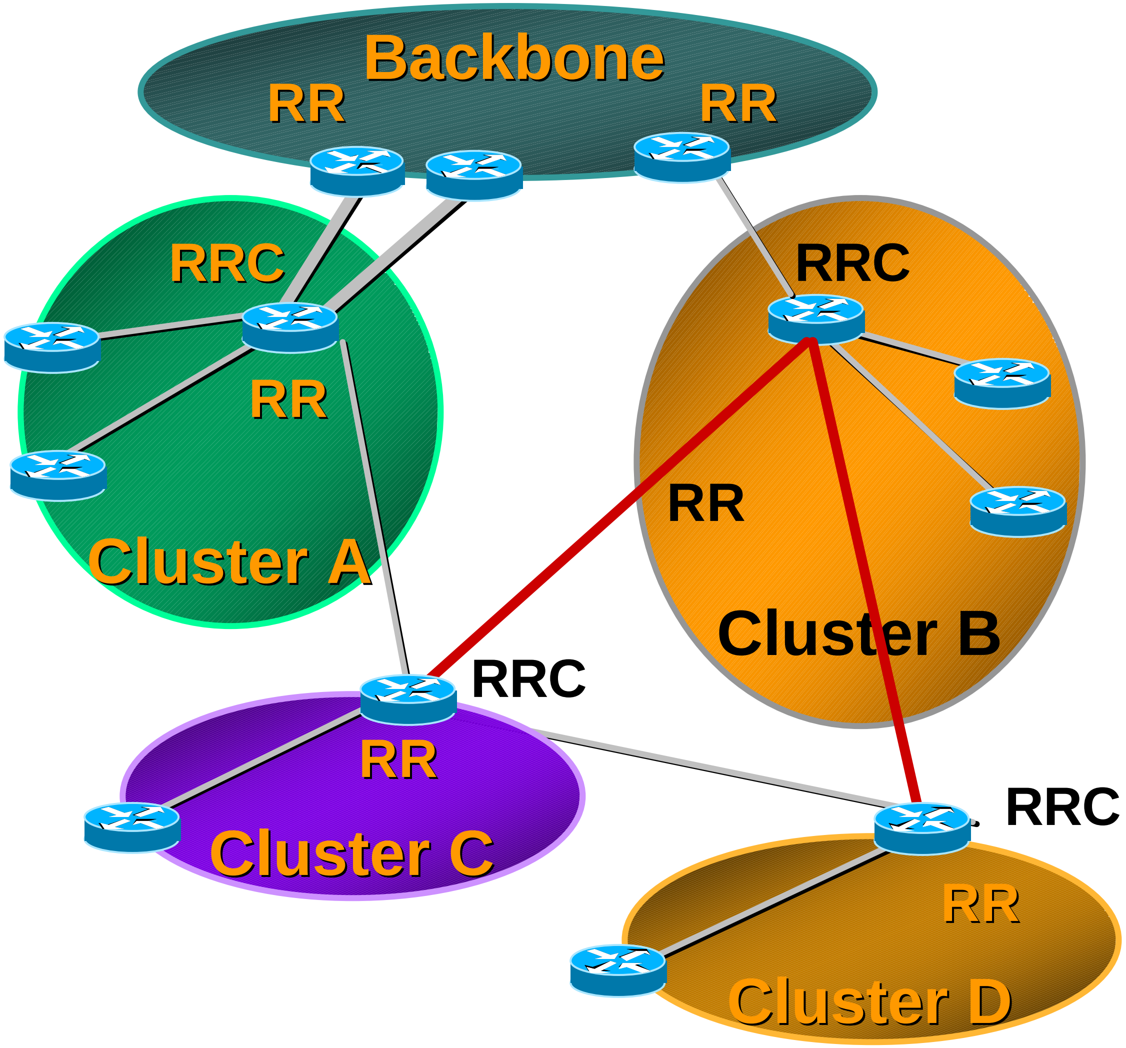


n=1000 => nearly half a million ibgp sessions!

Using Route Reflectors

Golden Rule of RR Loop Avoidance:

RR Topology Should Follow Physical Topology



What Is a Route Reflector?

- Reflector receives path from clients and non clients
- If best path is from a client, reflect to clients and non-clients
- If best path is from a non-client, reflect to clients

Configuration of RR Peer Groups

Cisco.com

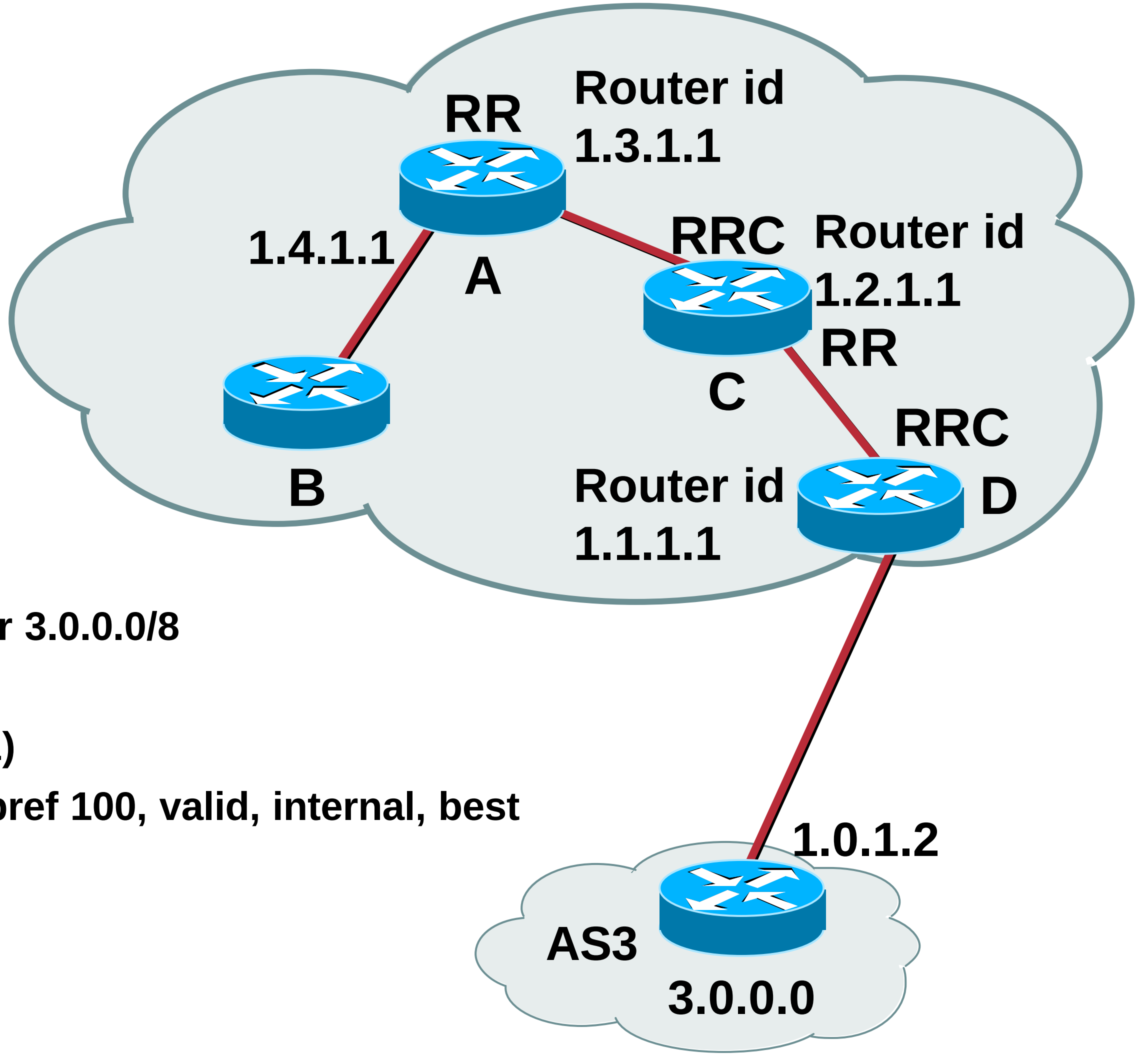
```
router bgp  1
neighbor rr-client peer-group
neighbor rr-client description RR clients
neighbor rr-client remote-as 1
neighbor rr-client update-source Loopback0
neighbor rr-client route-reflector-client
neighbor rr-client next-hop-self
neighbor rr-client send-community
neighbor rr-client version 4
neighbor rr-client password 7 03085A09
neighbor 10.0.1.1 peer-group rr-client
neighbor 10.0.2.2 peer-group rr-client
```

This line on RRs only
RRCs use still use
internal peer
group

Deploying Route Reflectors

- **Divide backbone into multiple clusters**
- **Each cluster contains at least one RR (multiple for redundancy), and multiple clients**
- **RRs are fully meshed via IBGP**
- **Still use single IGP—next-hop unmodified by RR; unless via explicit inbound route-map**

Hierarchical Route Reflector



- **Example:**

  RouterB>sh ip bgp 3.0.0.0

  BGP routing table entry for 3.0.0.0/8

  3

  1.0.1.2 from 1.4.1.1 (1.3.1.1)

  Origin IGP, metric 0, localpref 100, valid, internal, best

Originator: 1.1.1.1

Cluster list: 1.2.1.1, 1.3.1.1

BGP Attributes: ORIGINATOR_ID

- **ORIGINATOR_ID**

  Router ID of IBGP speaker that reflects RR client routes to non-clients

  Overridden by: *bgp cluster-id x.x.x.x*

- **Useful for troubleshooting and loop detection**

BGP Attributes: CLUSTER_LIST

- **CLUSTER_LIST**

  String of ORIGINATOR_IDs through which the route has passed

- **Useful for troubleshooting and loop detection**

So Far…

- Is IBGP peering Stable?

  Use loopbacks for peering
- Will it Scale?

  Use peer groups

  Use route reflectors
- Simple, hierarchical config?

Deploying eBGP

Customer Issues

- Steps

    Configure BGP (use session passwords!)

    Generate a stable aggregate

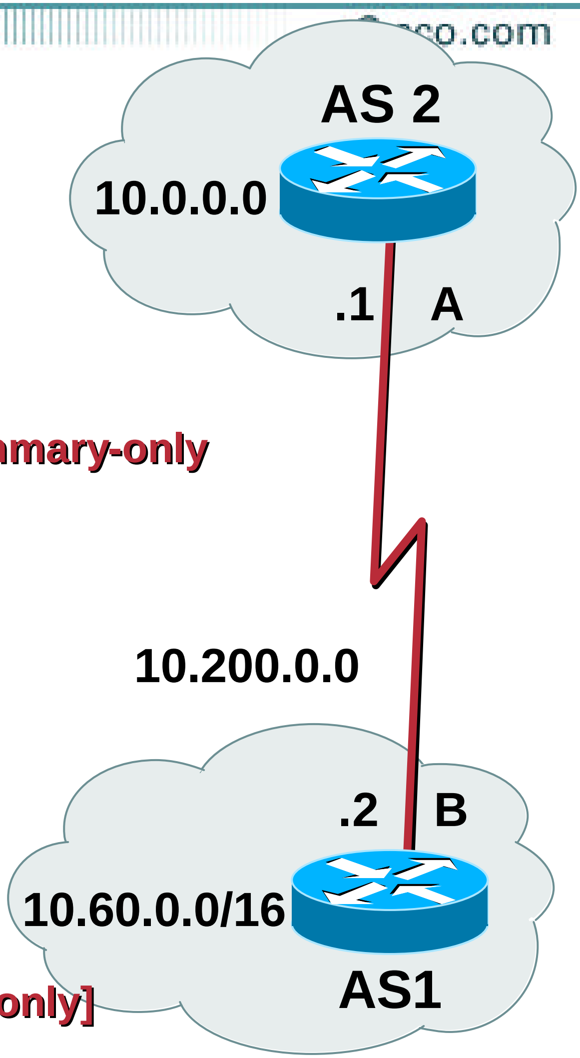
    Set inbound policy

    Set output policy

    Configure loadsharing/multihoming

Connecting to an ISP

- AS 100 is a customer of AS 200




```
Router B:
router bgp 100
aggregate-address 10.60.0.0 255.255.0.0 as-set summary-only
neighbor external remote-as 2
neighbor external description ISP connection
neighbor external remove-private-AS
neighbor external version 4
neighbor external prefix-list ispout out
neighbor external route-map ispout out
neighbor external route-map ispin in
neighbor external password 7 020A0559
neighbor external maximum-prefix 65000 [warning-only]
neighbor 10.200.0.1 peer-group external
```

What Is Aggregation?

- Summarization based on specifics from the BGP routing table

  10.60.1.0 255.255.255.0

  10.60.2.1 255.255.255.240

  => 10.60.0.0 255.255.0.0

How to Aggregate

- **aggregate-address 10.60.0.0 255.255.0.0 {as-set} {summary-only} {route-map}**
- **Use *as-set* to include path and community info from specifics**
- ***summary-only* suppresses specifics**
- **route-map sets other attributes**

Why Aggregate?

Cisco.com

- Reduce number of Internet prefixes
- Increase stability—aggregate stays even specifics come and go
- Stable aggregate generation:

```
router bgp 1
aggregate-address 10.60.0.0 255.255.0.0 as-set summary-only
network 10.60.1.0 255.255.255.0
ip route 10.60.1.0 255.255.255.0 null0 254
```

BGP Attributes
Atomic Aggregate

- Indicates loss of AS-PATH information
- Must not be removed once set
- Set by: *aggregate-address x.x.x.x*
- Not set if as-set keyword is used, however, AS-SET and COMMUNITY then carries information about specifics

BGP Attributes: Aggregator

- AS number and IP address of router generating aggregate
- Useful for troubleshooting

Aggregate Attributes

NEXT_HOP = local (0.0.0.0)

WEIGHT = 32768

LOCAL_PREF = none (assume 100)

AS_PATH = AS_SET or nothing

ORIGIN = IGP

MED = none

Why Inbound Policy?

- Apply a recognizable community to use in outbound filters or other policy
- Possibly adjust local-preference to override default of 100
- Multihoming loadsharing—more later

```
route-map ISPin permit 10
set local-preference 200
set community 1:2   ; routes from ISP
```

Why Outbound Policy?

- Main filter based on communities
- Adding a prefix filter helps protect against mistakes (can apply as-path filters too)
- Send community based on agreements with ISP (remember to add send-community line to config)
- Multihoming loadsharing policy

Outgoing Policy Config

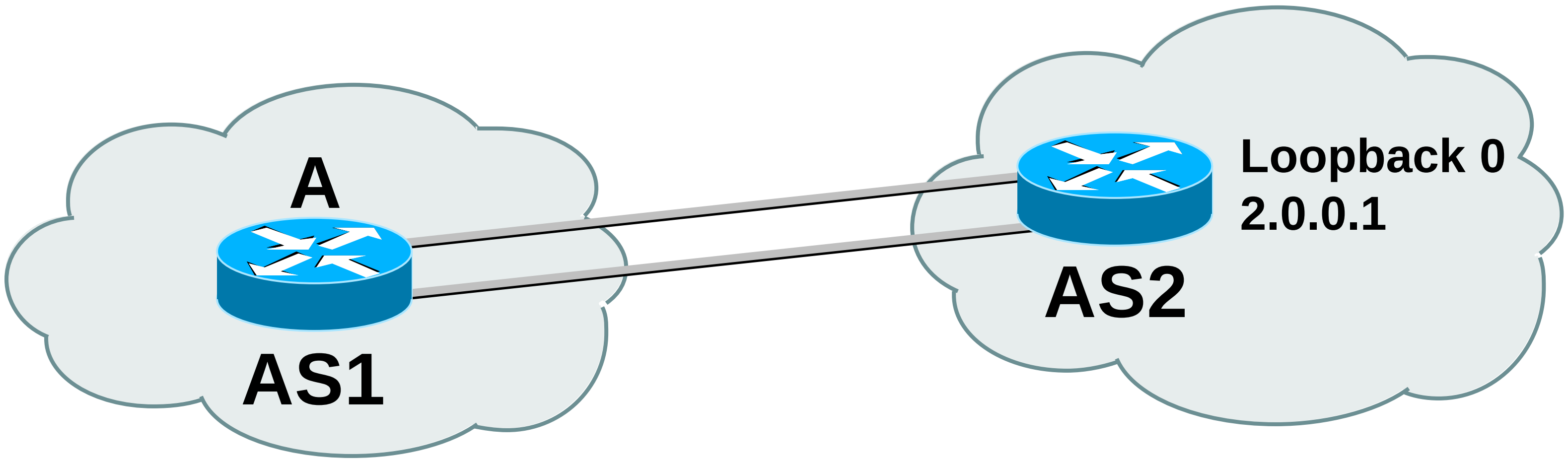
Cisco.com

```
ip prefix-list ISPout seq 5 permit 10.60.0.0 255.255.0.0
:
ip community-list 1 permit 1:1    ;all routes to send to ISP
:
route-map ISPout permit 10
match community 1    ;  Internet transit community
set community 1:3 [additive]  ; something agreed with ISP
```

Load-Sharing—Single Path

Cisco.com

```
Router A:
interface loopback 0
ip address 1.0.0.1 255.255.255.255
!
router bgp 1
neighbor 2.0.0.1 remote-as 2
neighbor 2.0.0.1 update-source loopback0
neighbor 2.0.0.1 ebgp-multi-hop 2
```



Load-Sharing—Multiple Paths from Same AS

Cisco.com

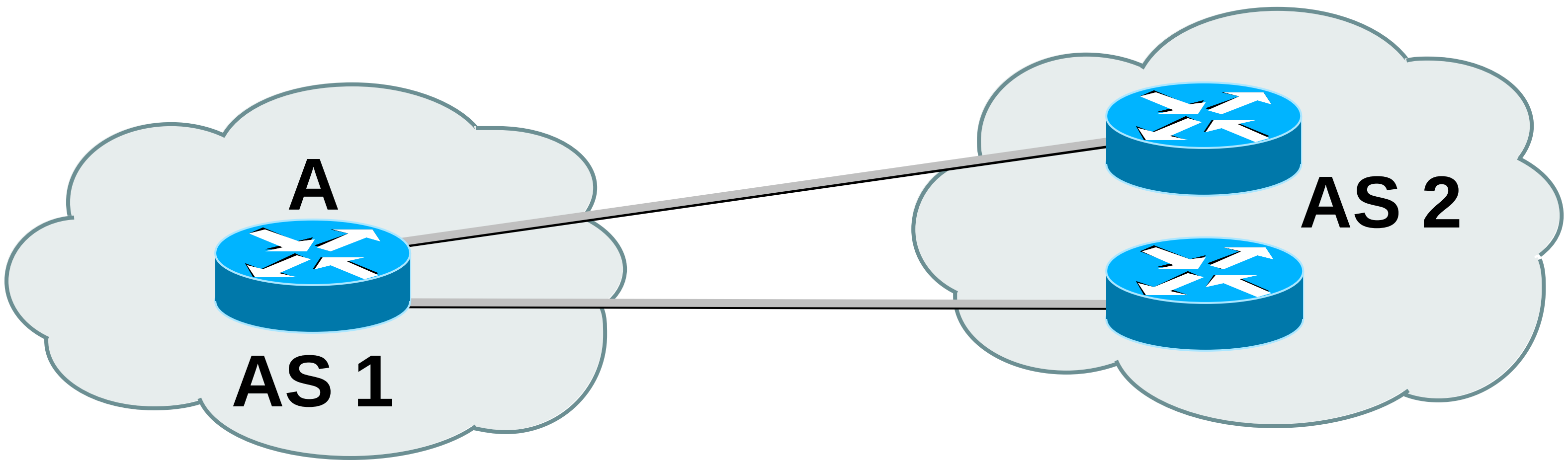
Router A:

router bgp 1

neighbor 2.0.0.1 remote-as 2

neighbor 2.0.0.1 remote-as 2

maximum-paths 2 ; can configure up to 6



What Is Multihoming?

- Connecting to two or more ISPs to increase:

  Reliability—one ISP fails, still OK

  Performance—better paths to common Internet destinations

Types of Multihoming

- **Three common cases:**
  - **Default from all ISPs**
  - **Customer+default routes from all ISPs**
  - **Full routes from ISPs**

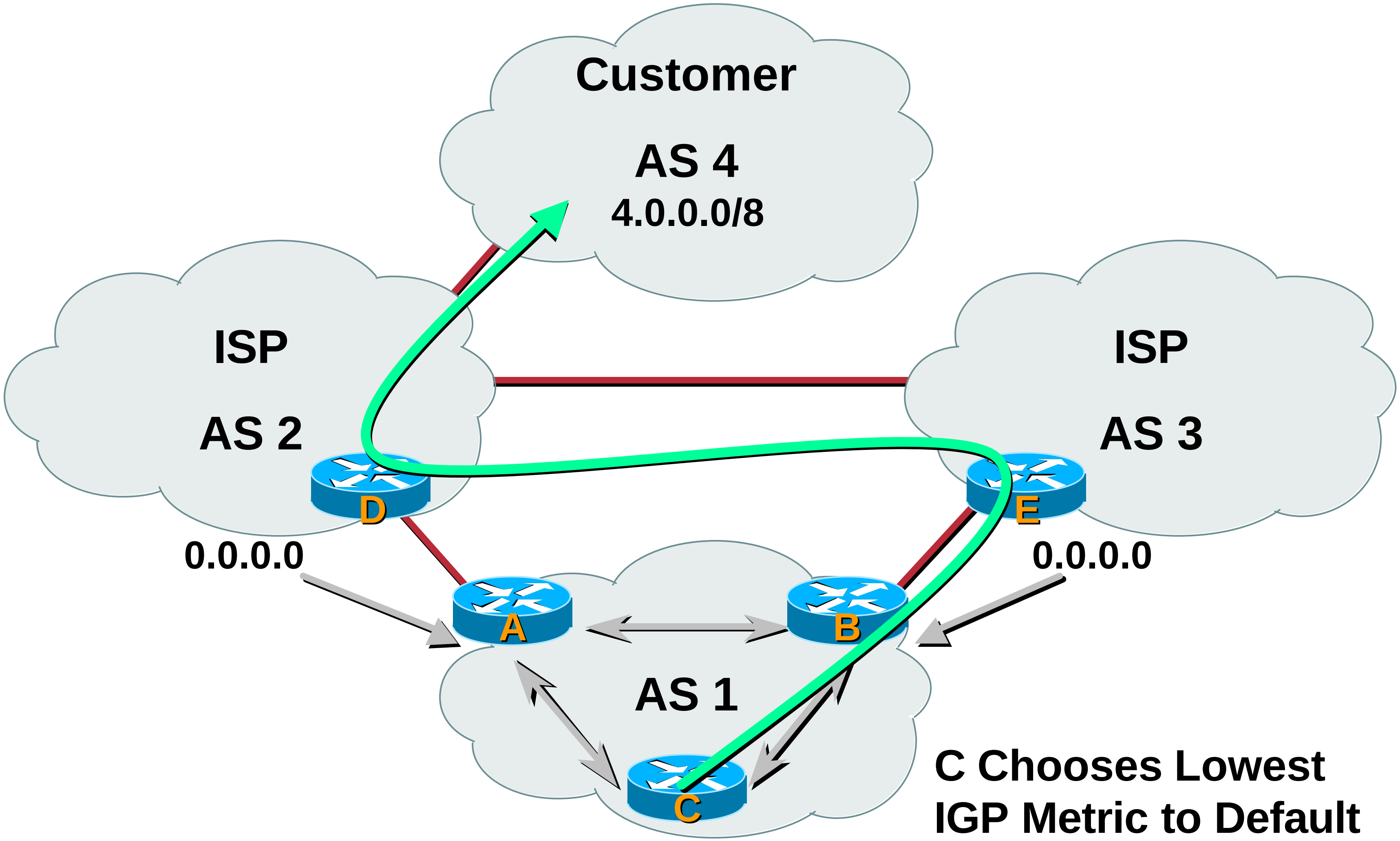
Default from All ISPs

- Low memory/CPU solution
- ISP sends BGP default => ISP decided by IGP metrics
  to reach default
- You send all your routes to ISP =>inbound path decided by Internet

  You can influence using AS-path prepend

Default from All ISPs

Cisco.com



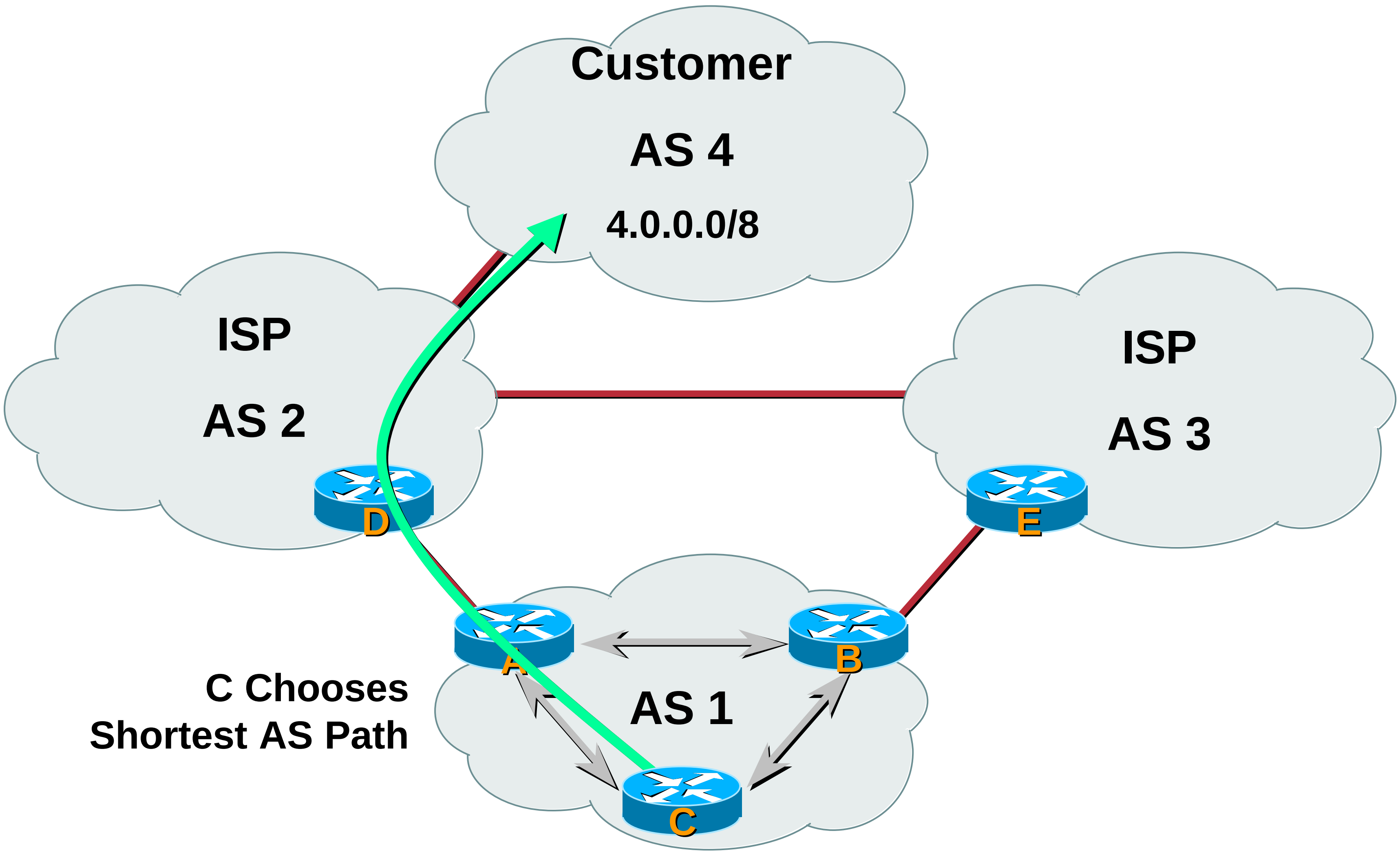
C Chooses Lowest IGP Metric to Default

Customer+Default from All ISPs

Cisco.com

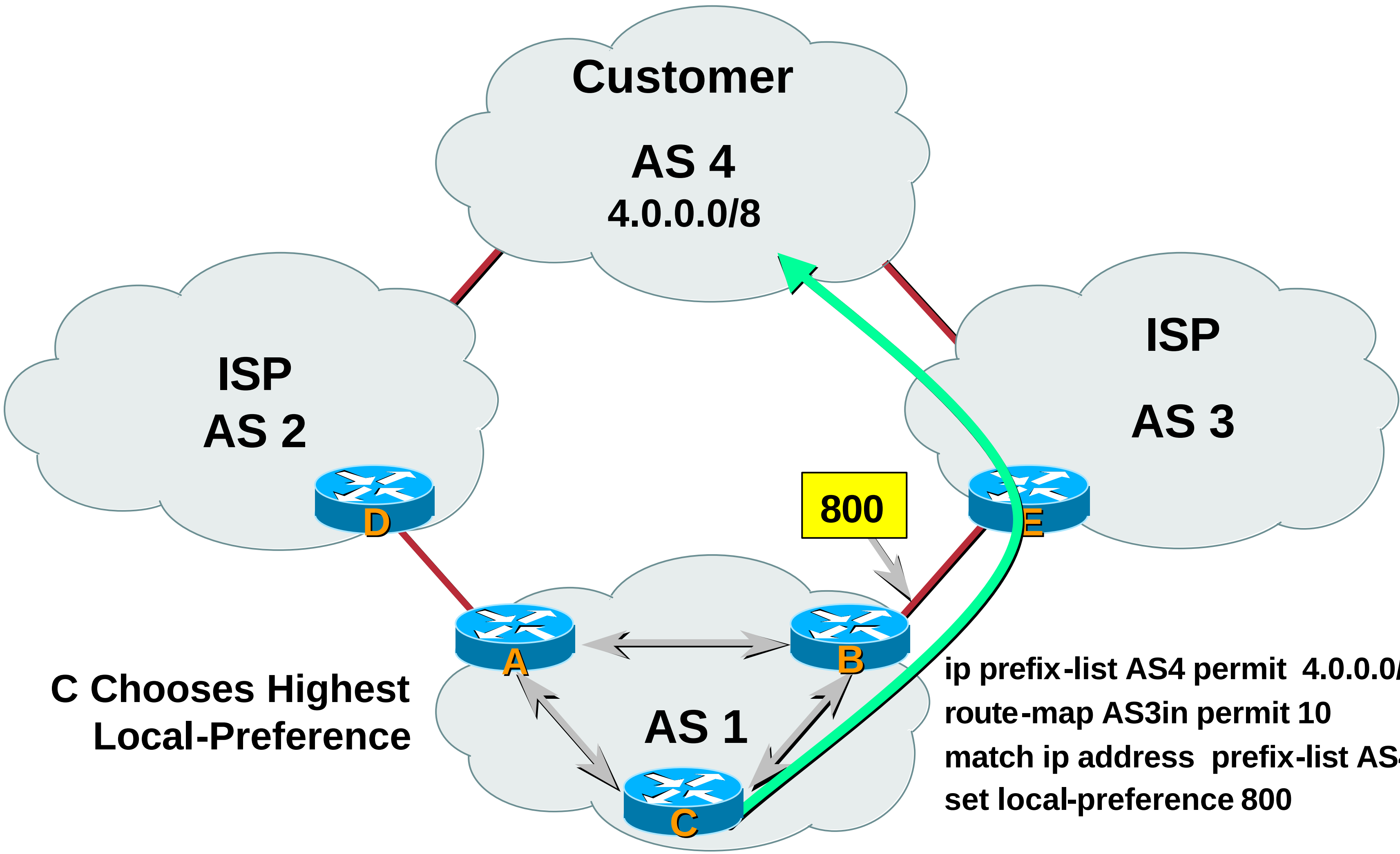
- Medium memory and CPU
- “Best” path—usually shortest AS-path
- Use local-preference to override based on prefix, as-path, or community
- IGP metric to default used for all other destinations

Customer Routers from All ISPs



Customer Routes from All ISPs

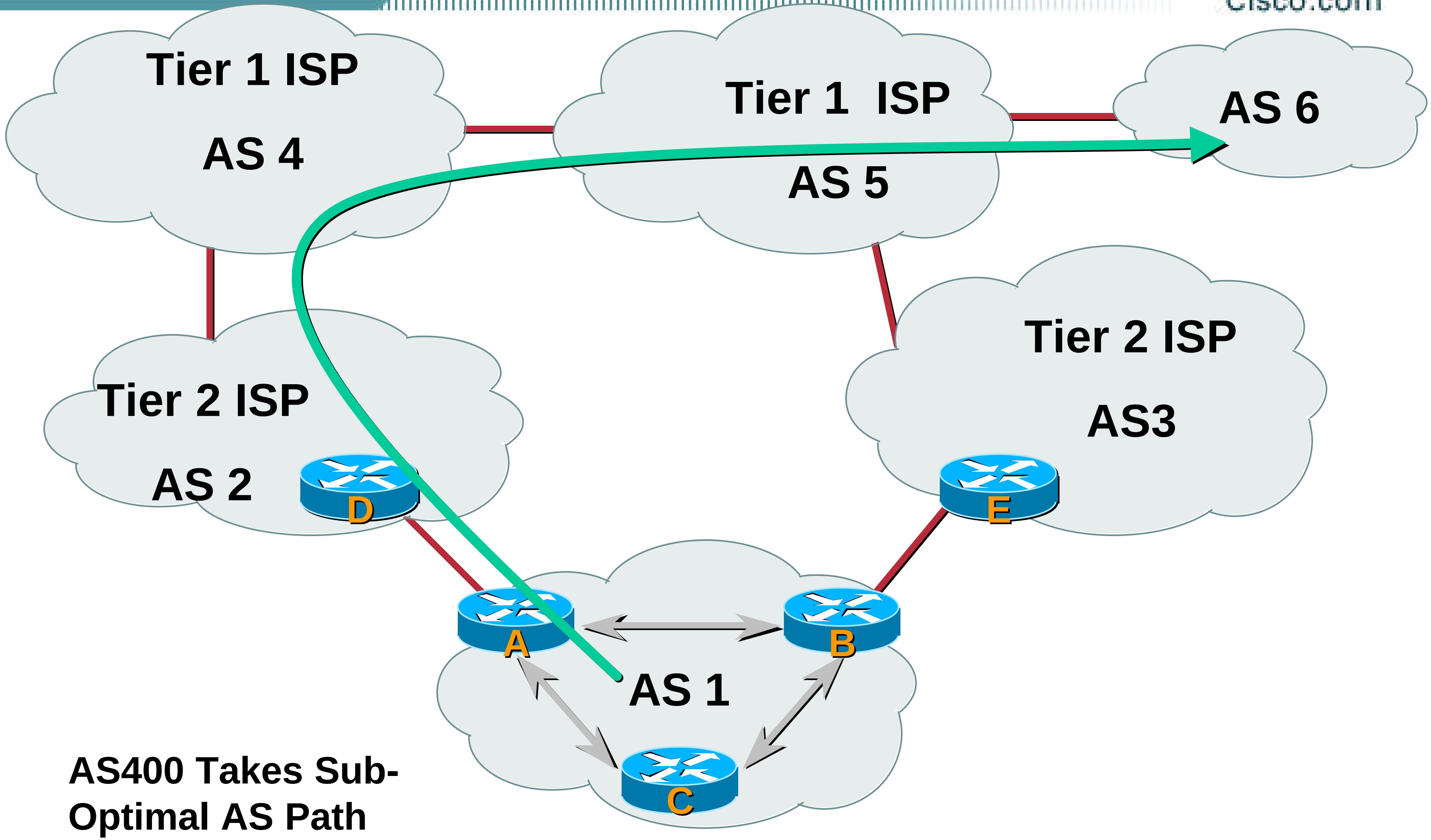
Cisco.com



C Chooses Highest Local-Preference

ip prefix-list AS4 permit 4.0.0.0/8
route-map AS3in permit 10
match ip address prefix-list AS4
set local-preference 800

Customer Routes from All ISPs

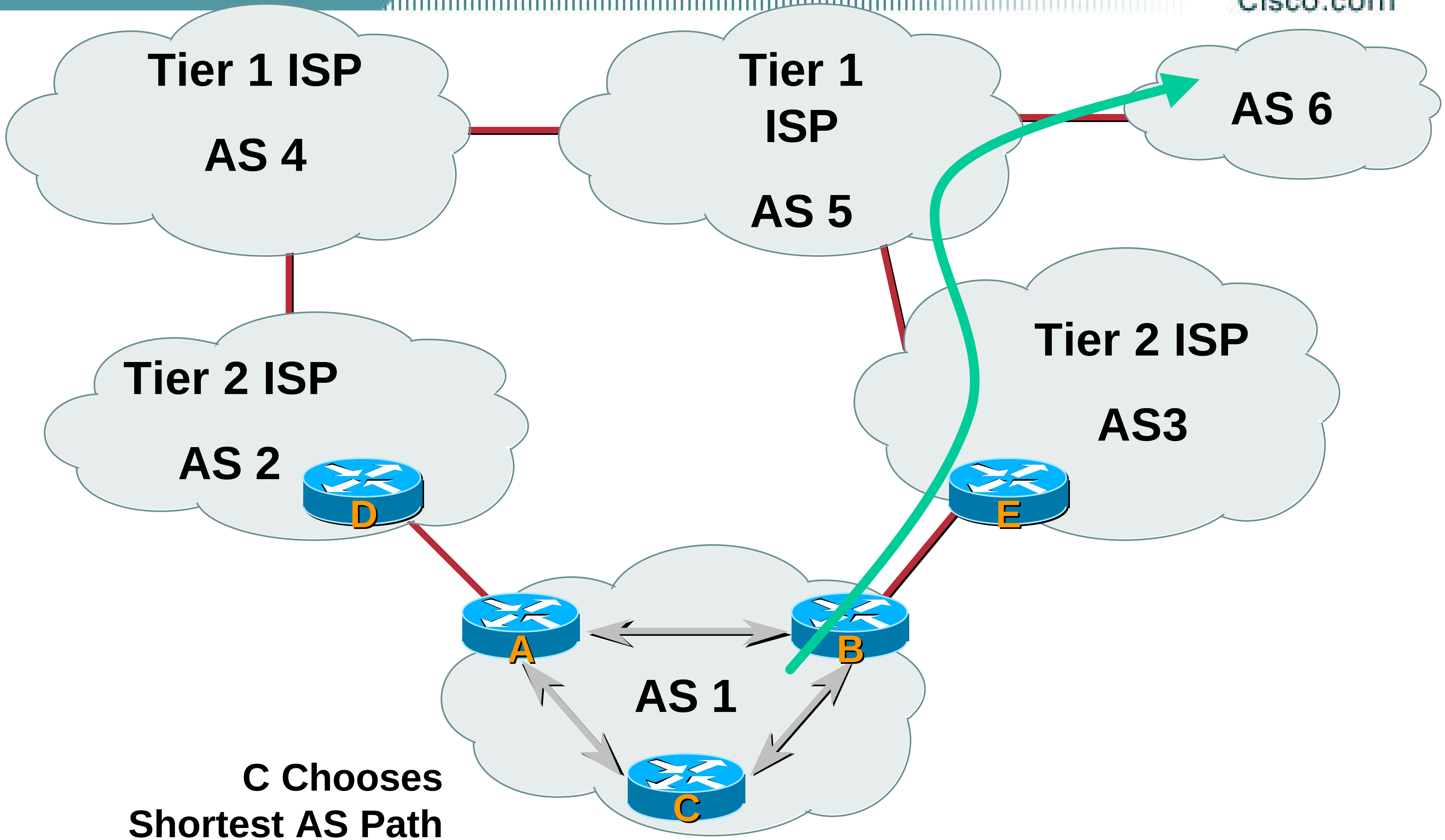


AS400 Takes Sub-Optimal AS Path

Full Routes from All ISPs

- Higher memory/CPU solution
- Reach all destinations by "best" path—usually shortest AS path
- Can still manually tune using local-pref and as-path/community/prefix matches

Full Routes from All ISPs



C Chooses Shortest AS Path

Controlling Inbound Traffic?

- Inbound is very difficult due to lack of transitive metric
- Can divide outgoing updates across providers, but what happens to redundancy?

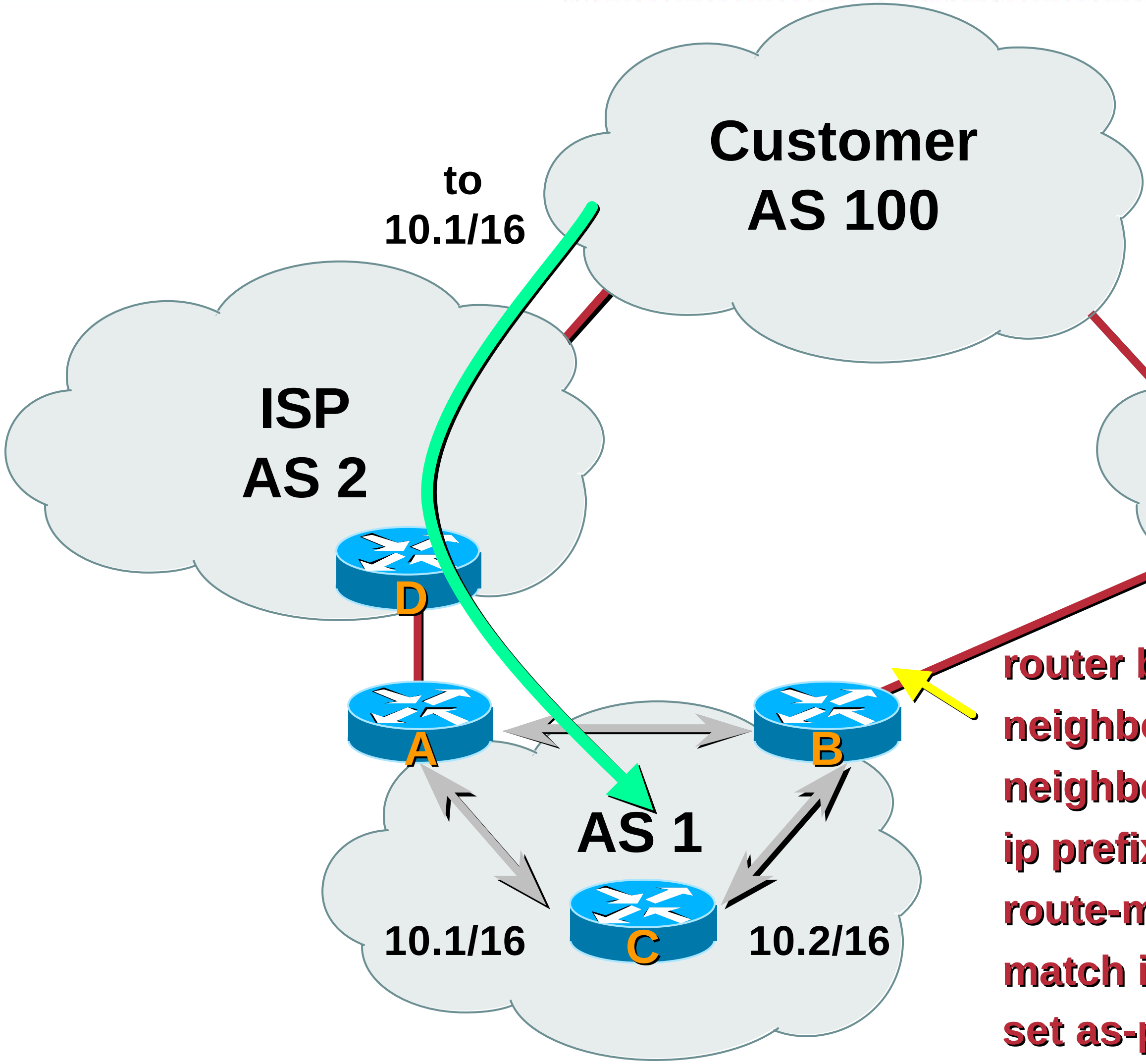
Controlling Inbound Traffic? (Cont.)

- **Bad Internet citizen:**
  - **Divide address space**
  - **Set as-path prepend**

- **Good Internet citizen**
  - **Divide address space**
  - **Use “advertise maps”**

Using AS-PATH Prepend

Cisco.com

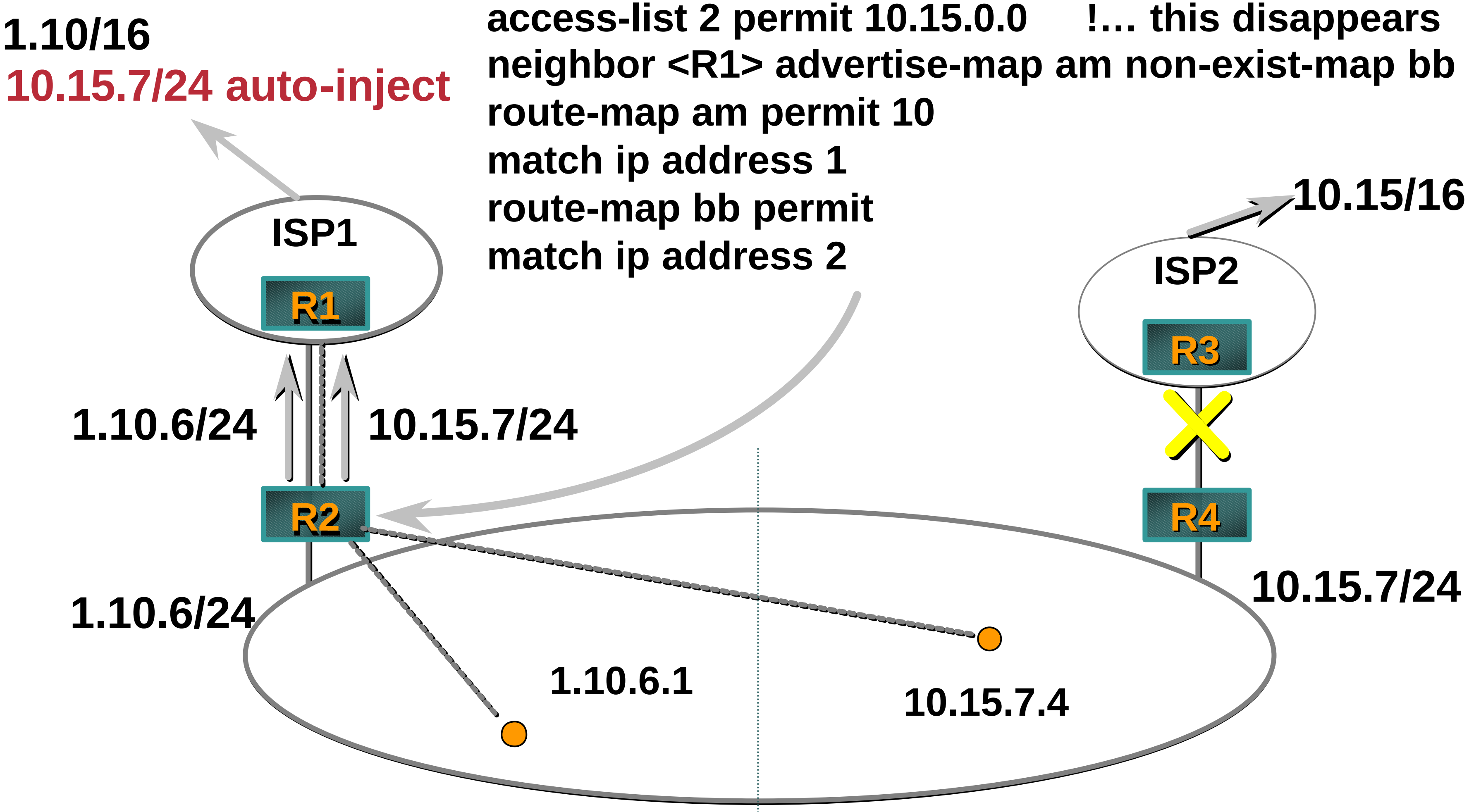


10.1.0.0/16 3 1 1
10.1.0.0/16 2 1 (best)
10.2.0.0/16 3 1 (best)
10.2.0.0/16 2 1 1

```
router bgp 1
neighbor 30.0.0.1 remote-as 3
neighbor 30.0.0.1 route-map AS3out out
ip prefix-list AS1 permit 10.1.0.0/16
route-map AS3out permit 10
match ip address prefix-list AS1
set as-path prepend 1
```

Using an Advertise-Map


```
access-list 1 permit 10.15.7.0     !Advertise when...
access-list 2 permit 10.15.0.0     !… this disappears
neighbor <R1> advertise-map am non-exist-map bb
route-map am permit 10
match ip address 1
route-map bb permit
match ip address 2
```



So Far…

- Stability through:
    - Aggregation
    - Multihoming
    - Inbound/outbound filtering
- Scalability of memory/CPU:
    - Default, customer routes, full routes
- Simplicity using “standard” solutions

Summary

Cisco.com

- **Scalability:**
  - Use attributes, especially community
  - Use peer groups and route reflectors
- **Stability:**
  - Use loopback addresses for IBGP
  - Generate aggregates
  - Apply passwords
  - Always filter inbound and outbound

Summary

- Simplicity—standard solutions:

  Three multihoming options

  Group customers into communities

  Apply standard policy at the edge

  Avoid “special configs”

  Script your config generation

BGP Scaling Techniques

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

- Dynamic reconfiguration
- Peer groups
- Route flap damping
- Route reflectors
- (Confederations)

Dynamic Reconfiguration

Route Refresh and

Soft Reconfiguration

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:**

  Consumes CPU

  Severely disrupts connectivity for all networks

Solution:

- **Route Refresh**

Route Refresh Capability

- Facilitates non-disruptive policy changes
- No configuration is needed
- No additional memory is used
- Requires peering routers to support “route refresh capability” – RFC2918
- clear ip bgp x.x.x.x in tells peer to resend full BGP announcement
- clear ip bgp x.x.x.x out resends full BGP announcement to peer

Dynamic Reconfiguration

Cisco.com

- **Use Route Refresh capability if supported**
  - **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 to be equivalent to a router reboot**

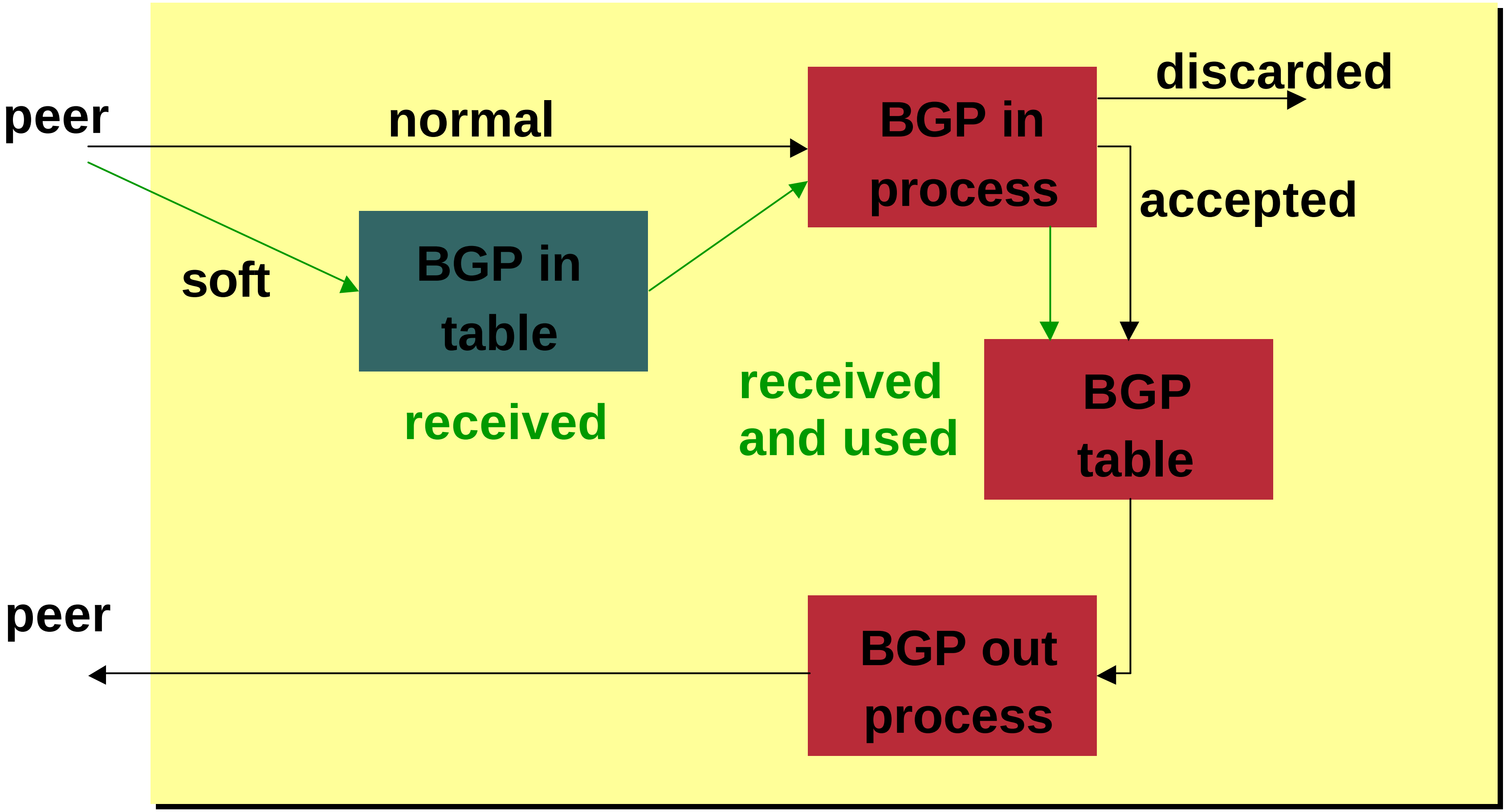
Soft Reconfiguration

- 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

- New policies can be activated without tearing down and restarting the peering session
- Configured on a per-neighbour basis
- Uses more memory to keep prefixes whose attributes have been changed or have not been accepted
- Also advantageous when operator requires to know which prefixes have been sent to a router prior to the application of any inbound policy

Soft Reconfiguration



Configuring Soft Reconfiguration

Cisco.com

```
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]
```

Managing Policy Changes

- **Ability to clear the BGP sessions of groups of neighbours configured according to several criteria**
- **`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

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

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 a Peer Group

Cisco.com

```
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 peer-group* !

Configuring a 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

Route Flap Damping

Stabilising the Network

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 aims to reduce scope of route flap propagation**

Route Flap Damping (continued)

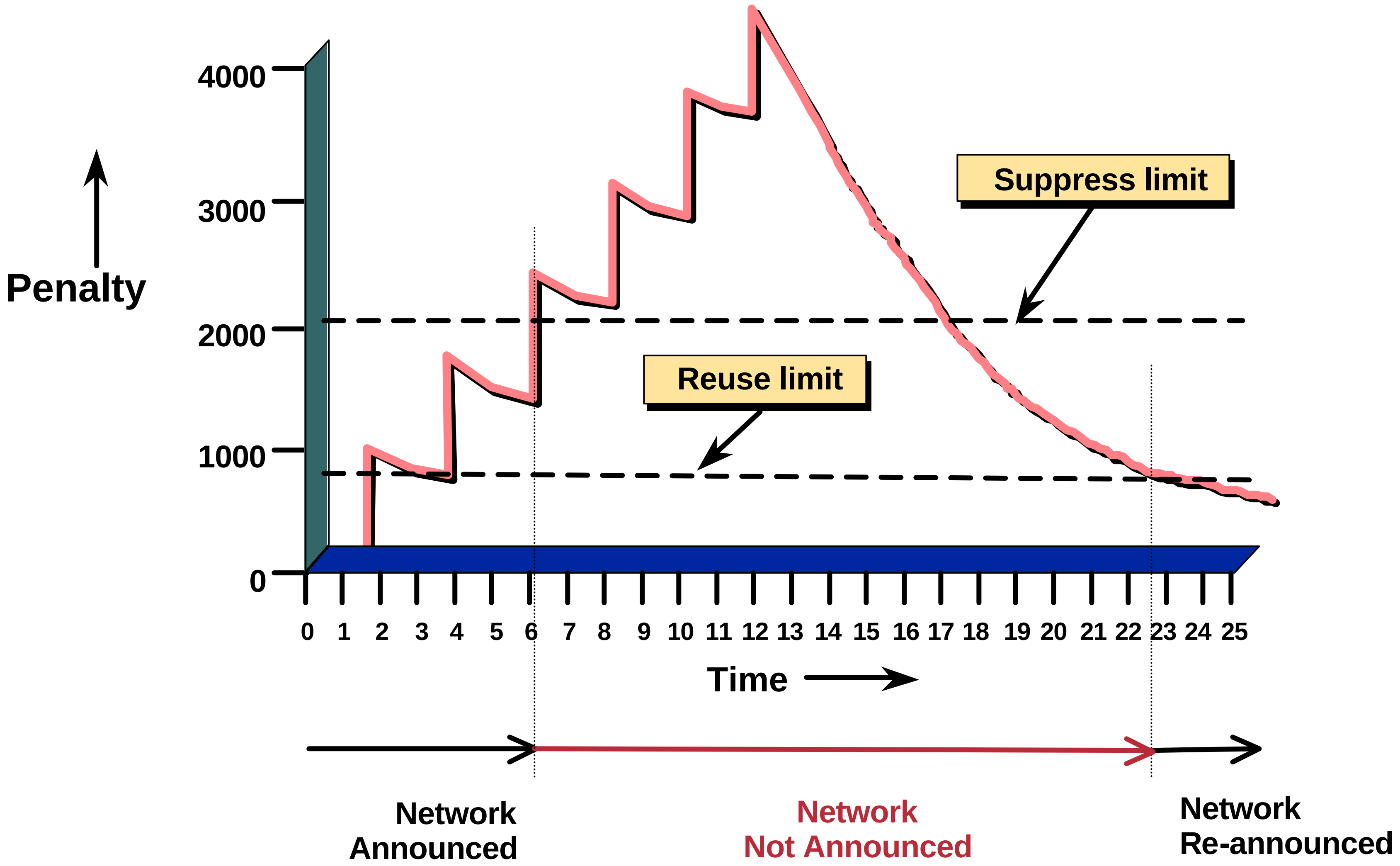
Cisco.com

- Requirements
  - Fast convergence for normal route changes
  - History predicts future behaviour
  - Suppress oscillating routes
  - Advertise stable routes
- Implementation described in RFC 2439

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
```

Operation

- Care required when setting parameters
- Penalty must be less than reuse-limit at the maximum suppress time
- Maximum suppress time and half life must allow penalty to be larger than suppress limit

Configuration

Cisco.com

- **Examples – ✘**

  bgp dampening 30 750 3000 60

  reuse-limit of 750 means maximum possible penalty is 3000 – no prefixes suppressed as penalty cannot exceed suppress-limit

- **Examples – ✓**

  bgp dampening 30 2000 3000 60

  reuse-limit of 2000 means maximum possible penalty is 8000 – suppress limit is easily reached

Configuration

- **Examples – ✘**

  bgp dampening 15 500 2500 30

  reuse-limit of 500 means maximum possible penalty is 2000 – no prefixes suppressed as penalty cannot exceed suppress-limit

- **Examples – ✓**

  bgp dampening 15 750 3000 45

  reuse-limit of 750 means maximum possible penalty is 6000 – suppress limit is easily reached

Maths!

Cisco.com

- **Maximum value of penalty is**

$$\text{max-penalty} = \text{reuse-limit} \times 2^{\left(\frac{\text{max-suppress-time}}{\text{half-life}}\right)}$$

- **Always make sure that suppress-limit is LESS than max-penalty otherwise there will be no route damping**

Enhancements

- **Selective damping based on**

  AS-path, Community, Prefix

- **Variable damping**

  recommendations for ISPs

  http://www.ripe.net/docs/ripe-229.html

- **Flap statistics**

  ```
  show ip bgp neighbor <x.x.x.x> [dampened-routes |
  flap-statistics]
  ```

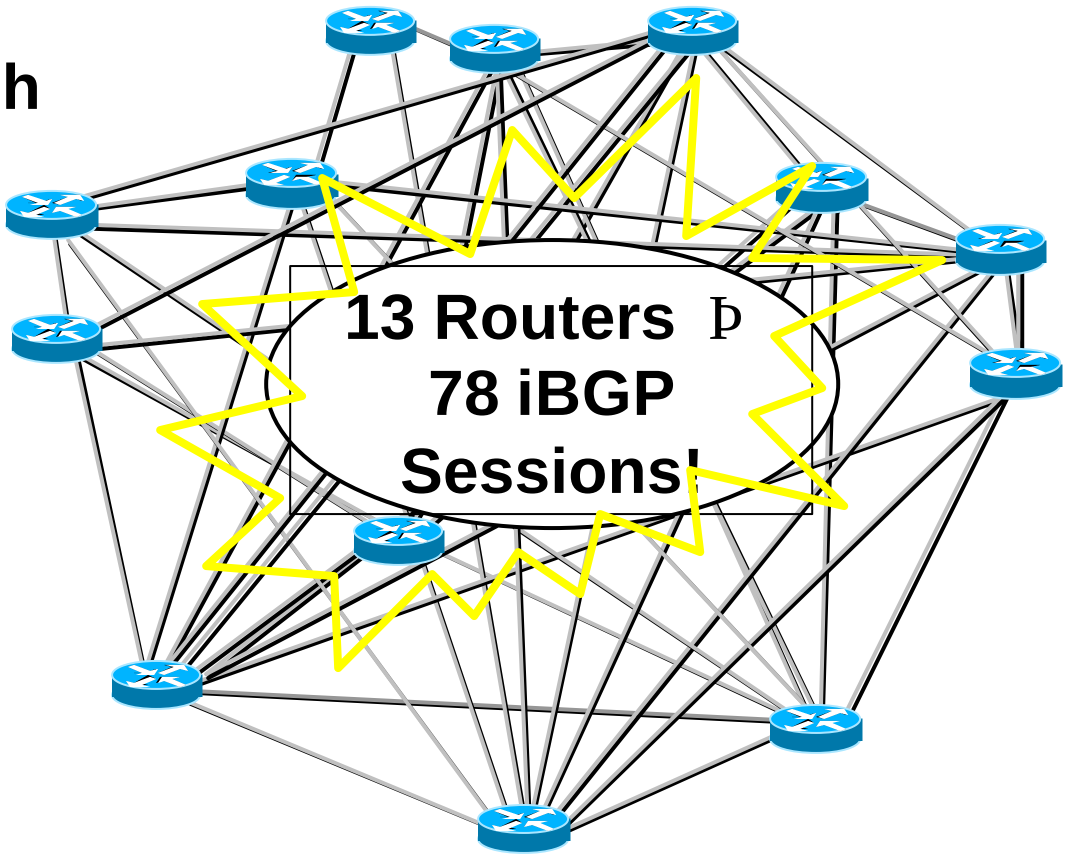
Route Reflectors

Scaling the iBGP mesh

Scaling iBGP mesh

Avoid ½n(n-1) iBGP mesh

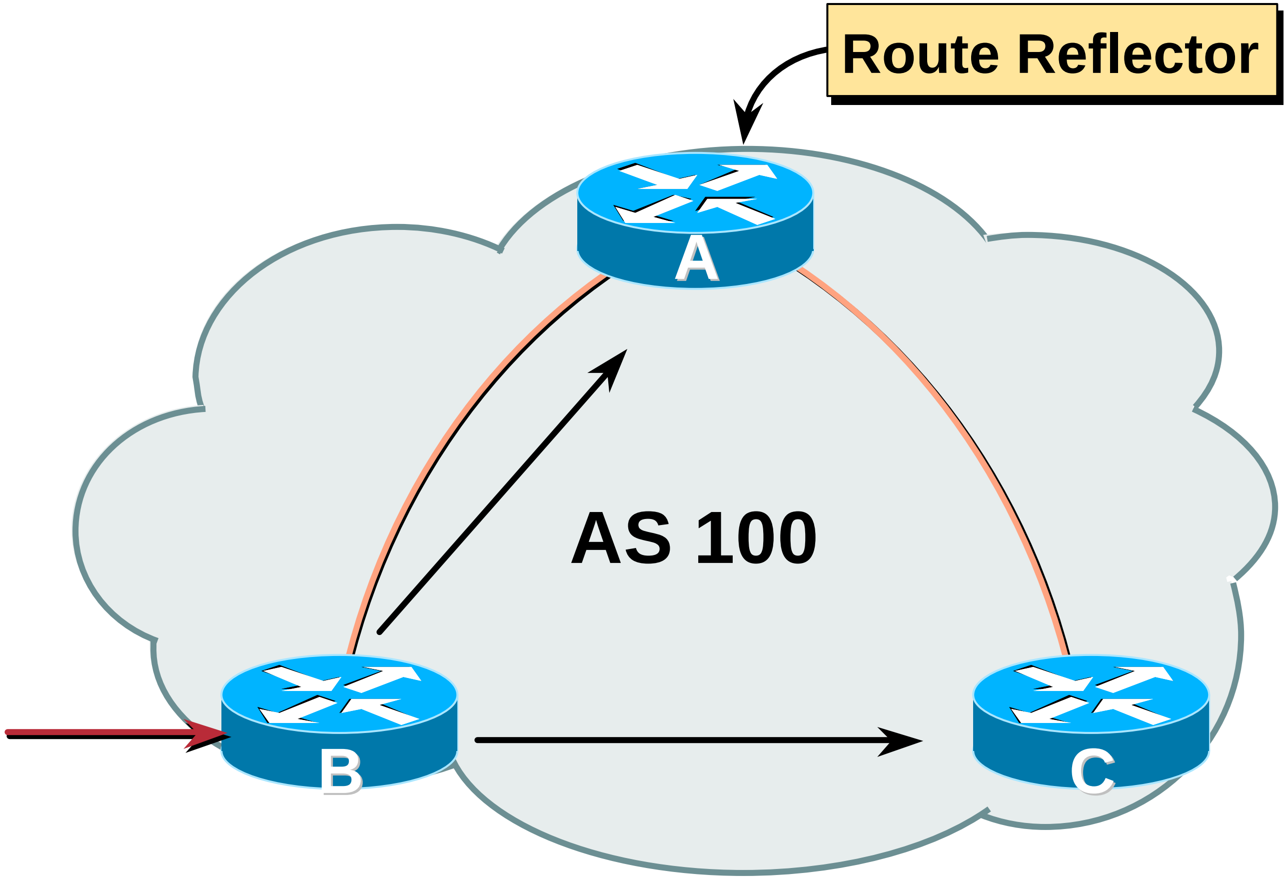
n=1000 Þ nearly half a million ibgp sessions!



Two solutions

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

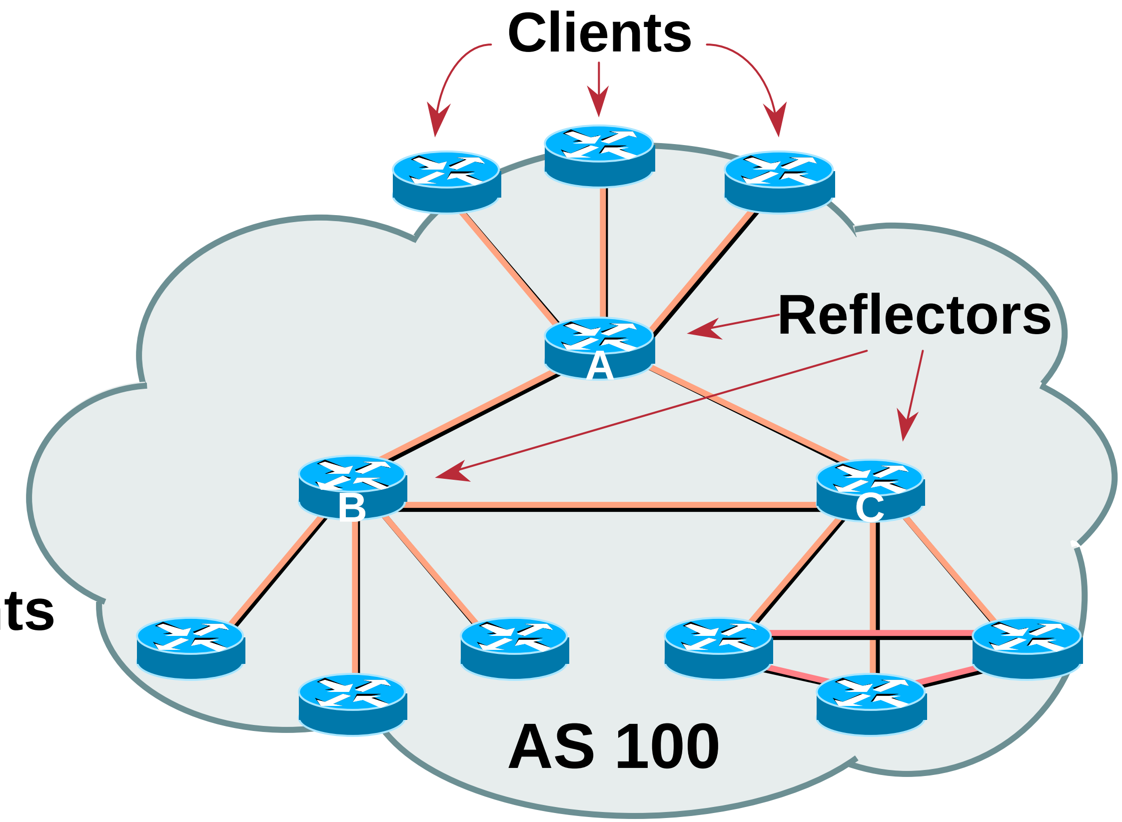
Route Reflector: Principle



Route Reflector

Cisco.com

- 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 RFC2796



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

Cisco.com

- **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 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: Benefits

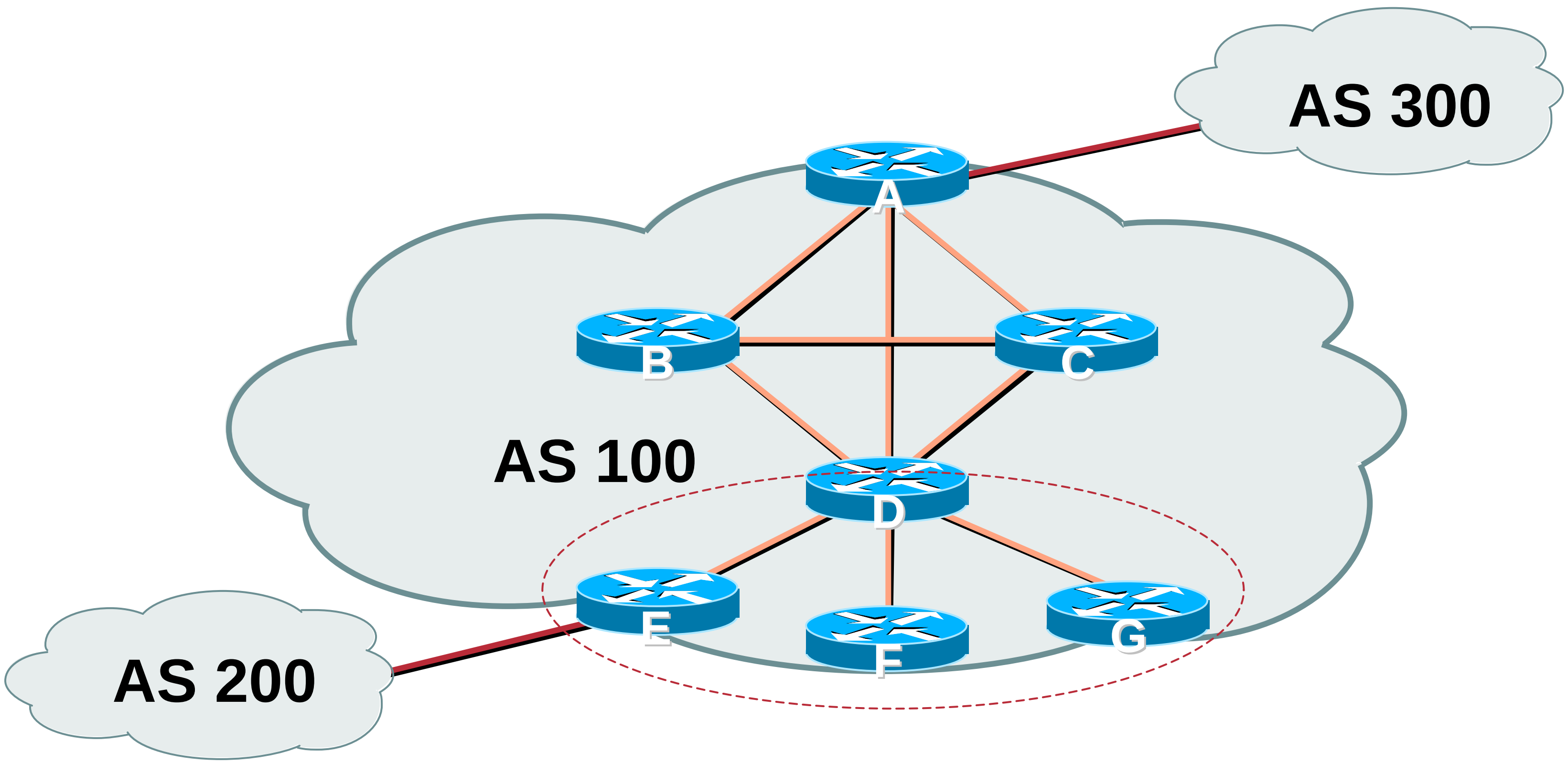
- Solves iBGP mesh problem
- Packet forwarding is not affected
- Normal BGP speakers co-exist
- Multiple reflectors for redundancy
- Easy migration
- Multiple levels of route reflectors

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

Cisco.com



- **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 4 techniques should be core requirements on all ISP networks
  - Route Refresh (or Soft Reconfiguration)
  - Peer groups
  - Route Flap Damping
  - Route Reflectors

BGP Confederations

Confederations

- Divide the AS into sub-AS

  eBGP between sub-AS, but some iBGP information is kept

    Preserve NEXT_HOP across the sub-AS (IGP carries this information)

    Preserve LOCAL_PREF and MED

- Usually a single IGP
- Described in RFC3065

Confederations

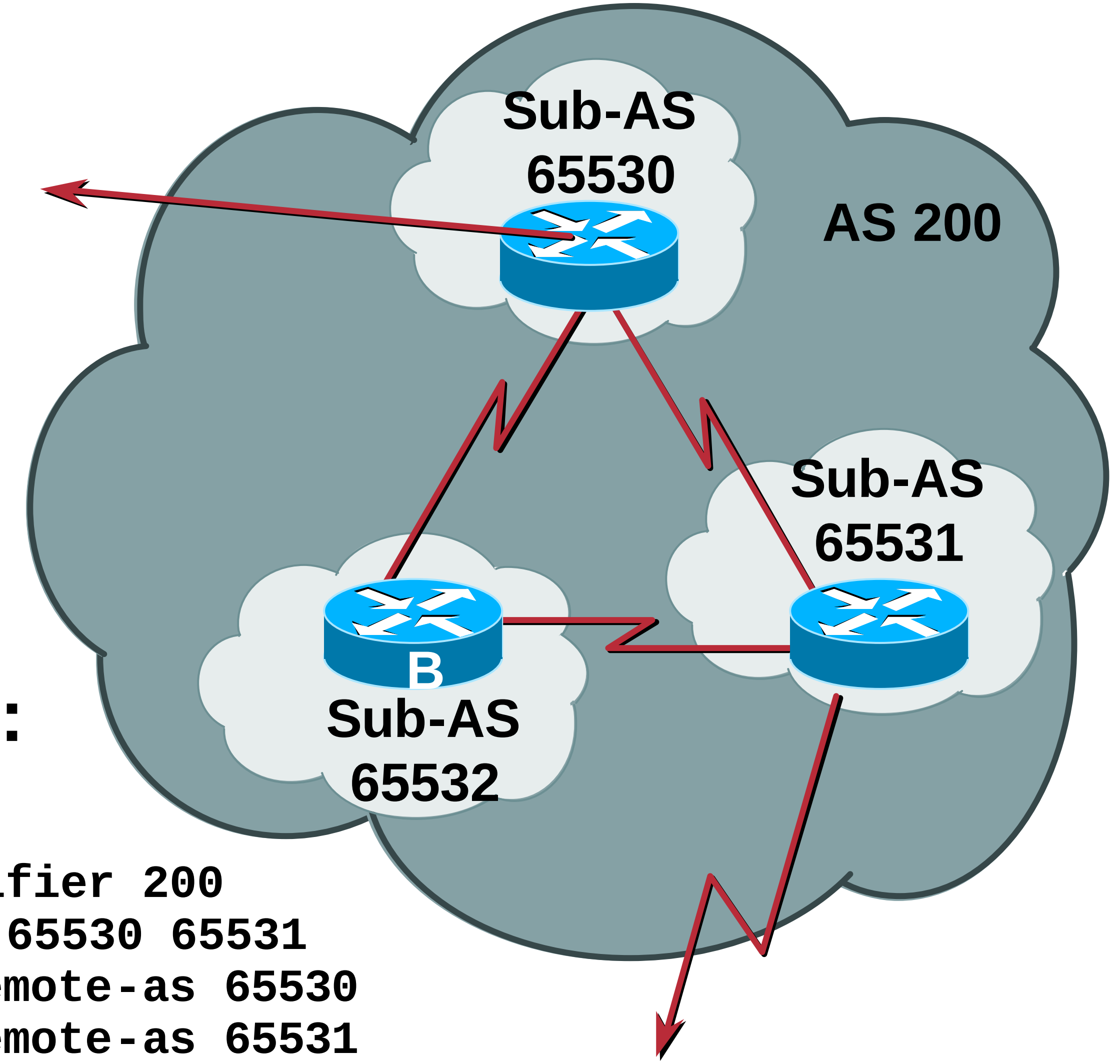
- **Visible to outside world as single AS – "Confederation Identifier"**

  Each sub-AS uses a number from the private space (64512-65534)

- **iBGP speakers in sub-AS are fully meshed**

  The total number of neighbors is reduced by limiting the full mesh requirement to only the peers in the sub-AS

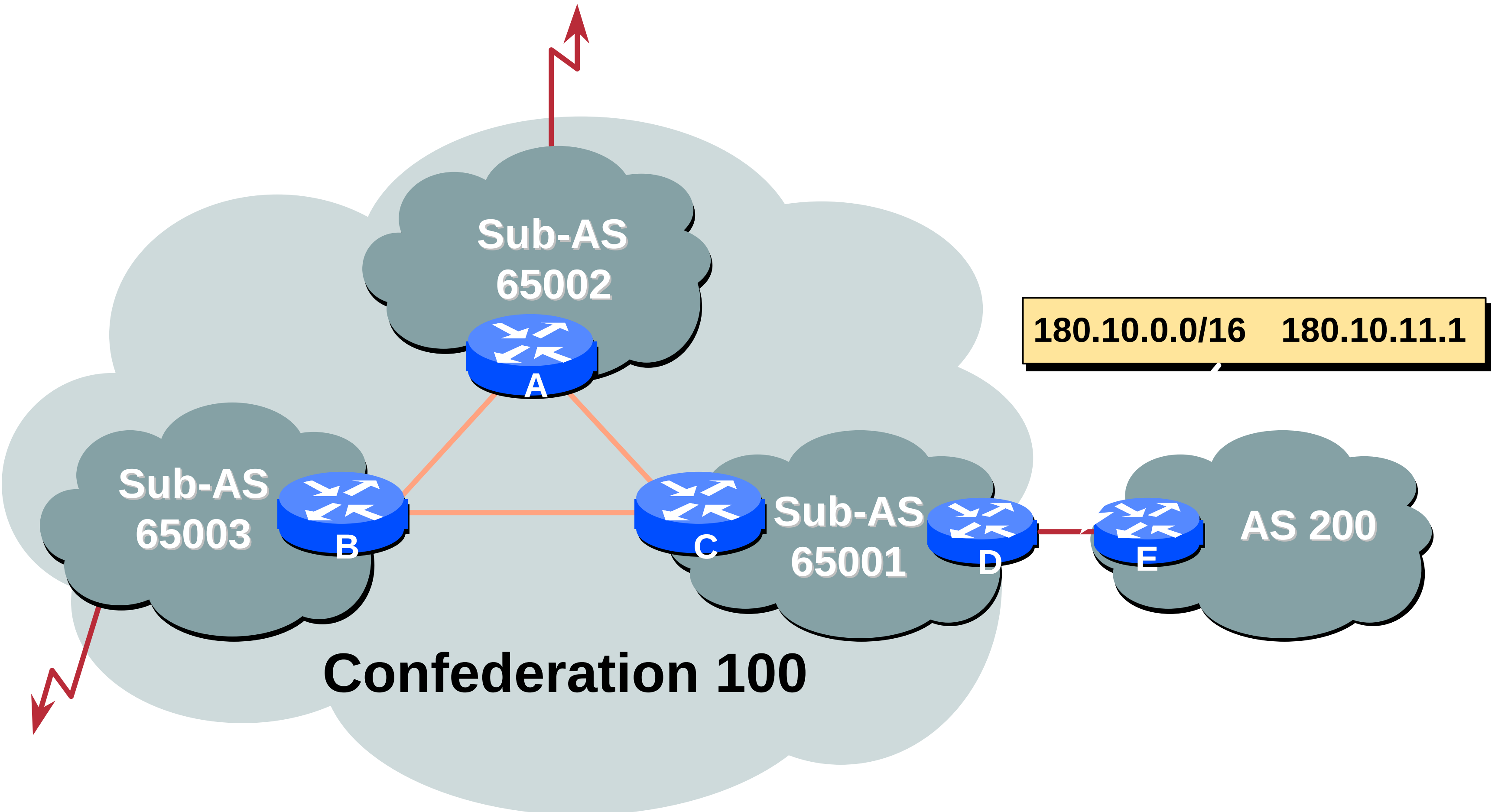
Confederations



- **Configuration (rtr B):**

```
router bgp 65532
 bgp confederation identifier 200
 bgp confederation peers 65530 65531
 neighbor 141.153.12.1 remote-as 65530
 neighbor 141.153.17.2 remote-as 65531
```

Confederations: Next Hop



Confederation: Principle

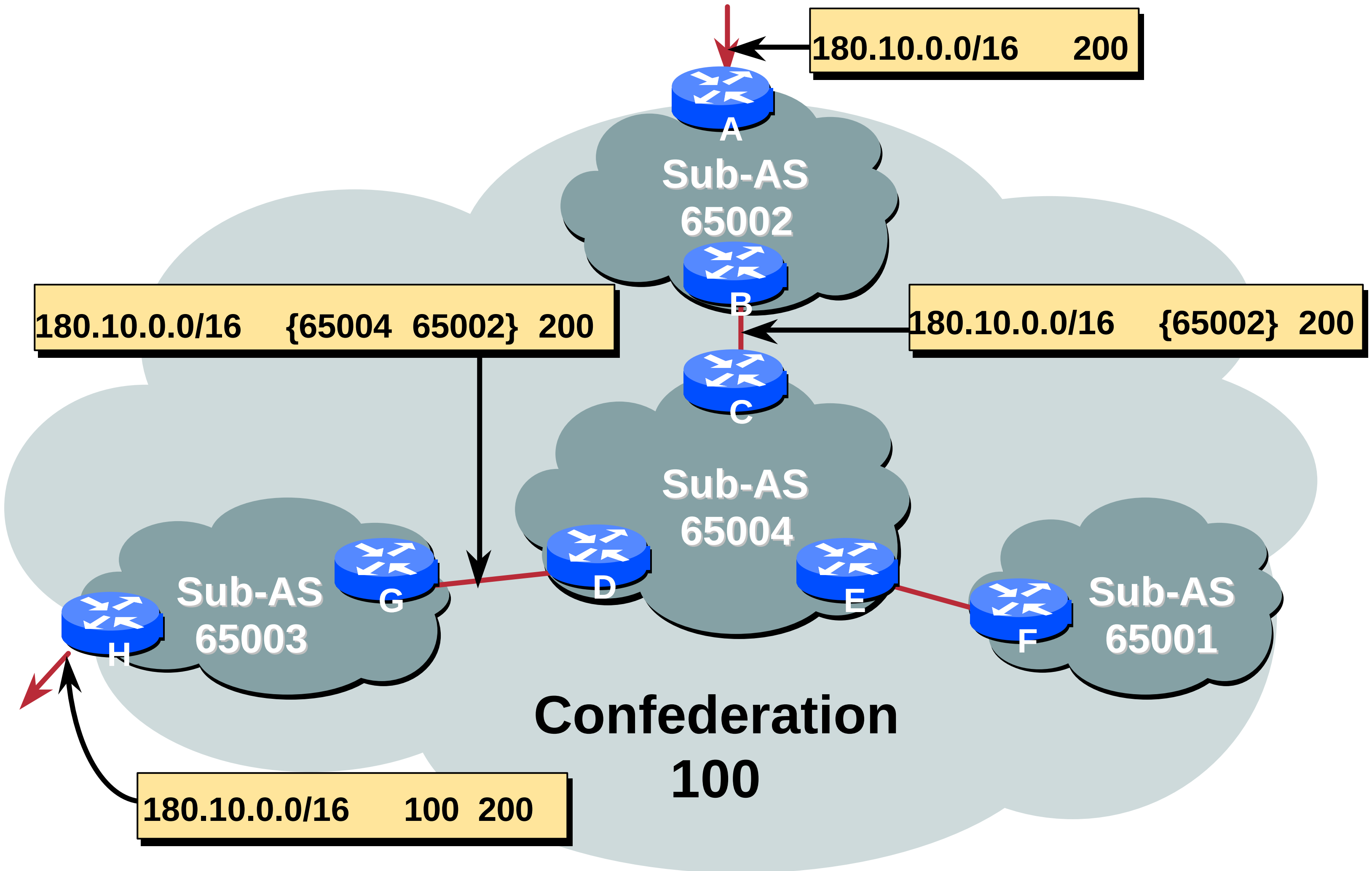
- Local preference and MED influence path selection
- Preserve local preference and MED across sub-AS boundary
- Sub-AS eBGP path administrative distance

Confederations: Loop Avoidance

- **Sub-AS traversed are carried as part of AS-path**
- **AS-sequence and AS path length**
- **Confederation boundary**
- **AS-sequence should be skipped during MED comparison**

Confederations: AS-Sequence

Cisco.com



Route Propagation Decisions

- **Same as with “normal” BGP:**
  - **From peer in same sub-AS → only to external peers**
  - **From external peers → to all neighbors**
- **“External peers” refers to**
  - **Peers outside the confederation**
  - **Peers in a different sub-AS**
    - **Preserve LOCAL_PREF, MED and NEXT_HOP**

Confederations (cont.)

Cisco.com

- **Example (cont.):**

```
BGP table version is 78, local router ID is 141.153.17.1
Status codes: s suppressed, d damped, h history, * valid, >
best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network         Next Hop      Metric LocPrf Weight Path
*> 10.0.0.0     141.153.14.3    0     100       0    (65531) 1 i
*> 141.153.0.0 141.153.30.2    0     100       0    (65530) i
*> 144.10.0.0  141.153.12.1    0     100       0    (65530) i
*> 199.10.10.0 141.153.29.2    0     100       0    (65530) 1 i
```

More points about confederations

- Can ease “absorbing” other ISPs into you ISP – e.g., if one ISP buys another (can use local-as feature to do a similar thing)
- You can use route-reflectors with confederation sub-AS to reduce the sub-AS iBGP mesh

Confederations: Benefits

Cisco.com

- Solves iBGP mesh problem
- Packet forwarding not affected
- Can be used with route reflectors
- Policies could be applied to route traffic between sub-AS's

Confederations: Caveats

- Minimal number of sub-AS
- Sub-AS hierarchy
- Minimal inter-connectivity between sub-AS's
- Path diversity
- Difficult migration

  BGP reconfigured into sub-AS

  must be applied across the network

RRs or Confederations

	Internet Connectivity	Multi-Level Hierarchy	Policy Control	Scalability	Migration Complexity
Confederations	Anywhere in the Network	Yes	Yes	Medium	Medium to High
Route Reflectors	Anywhere in the Network	Yes	Yes	Very High	Very Low

Most new service provider networks now deploy Route Reflectors from Day One

BGP Scaling Techniques

Troubleshooting BGP

Before We Begin…

- **My assumptions**

  Operational experience with BGP

  Intermediate to advanced knowledge of the protocol

- **What can you expect to get from this presentation?**

  Learn how to use show commands and debugs to troubleshoot BGP problems

  Go through various real world examples

Agenda

Cisco.com

- Peer Establishment
- Missing Routes
- Inconsistent Route Selection
- Loops and Convergence Issues

Peer Establishment

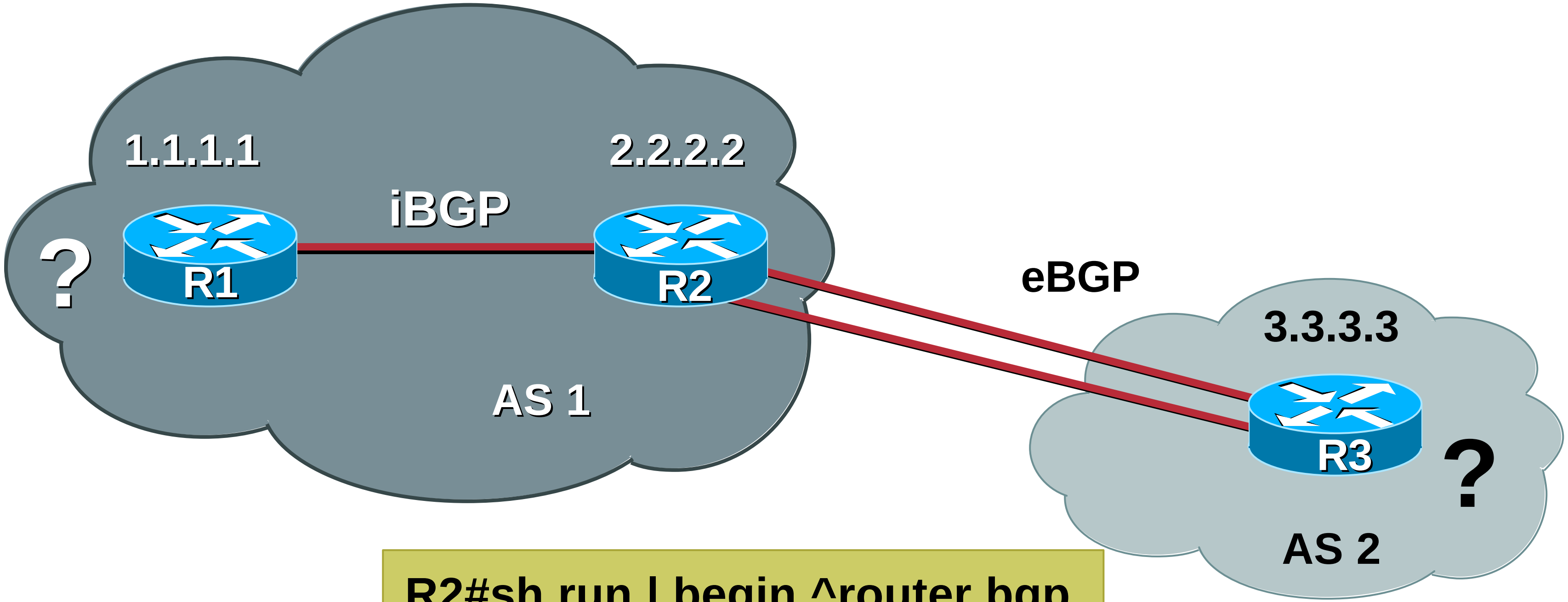
- **Routers establish a TCP session**
  - **Port 179—Permit in ACLs**
  - **IP connectivity (route from IGP)**
- **OPEN messages are exchanged**
  - **Peering addresses must match the TCP session**
  - **Local AS configuration parameters**

Common Problems

Cisco.com

- **Sessions are not established**
  - **No IP reachability**
  - **Incorrect configuration**
- **Peers are flapping**
  - **Layer 2 problems**

Peer Establishment—Diagram




```
R2#sh run | begin ^router bgp
router bgp 1
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 1
 neighbor 3.3.3.3 remote-as 2
```

Peer Establishment—Symptoms

Cisco.com

```
R2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 1
BGP table version is 1, main routing table version 1
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State
1.1.1.1         4    1        0       0        0    0    0 never    Active
3.3.3.3         4    2        0       0        0    0    0 never    Idle
```

- **Both peers are having problems**

  State may change between Active, Idle and Connect

Peer Establishment

- Is the Local AS configured correctly?
- Is the remote-as assigned correctly?
- Verify with your diagram or other documentation!

Local AS

```
R2#
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 3.3.3.3 remote-as 2
```

iBGP Peer

eBGP Peer

Peer Establishment—iBGP

- Assume that IP connectivity has been checked
- Check TCP to find out what connections we are accepting

```
R2#show tcp brief all
TCB            Local Address          Foreign Address        (state)
005F2934   *.179                      3.3.3.3.*               LISTEN
0063F3D4  *.179                       1.1.1.1.*               LISTEN
```

We Are Listening for TCP Connections for Port 179 for the Configured Peering Addresses Only!

```
R2#debug ip tcp transactions
TCP special event debugging is on
R2#
TCP: sending RST, seq 0, ack 2500483296
TCP: sent RST to 4.4.4.4:26385 from 2.2.2.2:179
```

Remote Is Trying to Open the Session from 4.4.4.4 Address…

Peer Establishment—iBGP

What about Us?

```
R2#debug ip bgp
BGP debugging is on
R2#
BGP: 1.1.1.1 open active, local address 4.4.4.5
BGP: 1.1.1.1 open failed: Connection refused by remote host
```

We Are Trying to Open the Session from 4.4.4.5 Address…

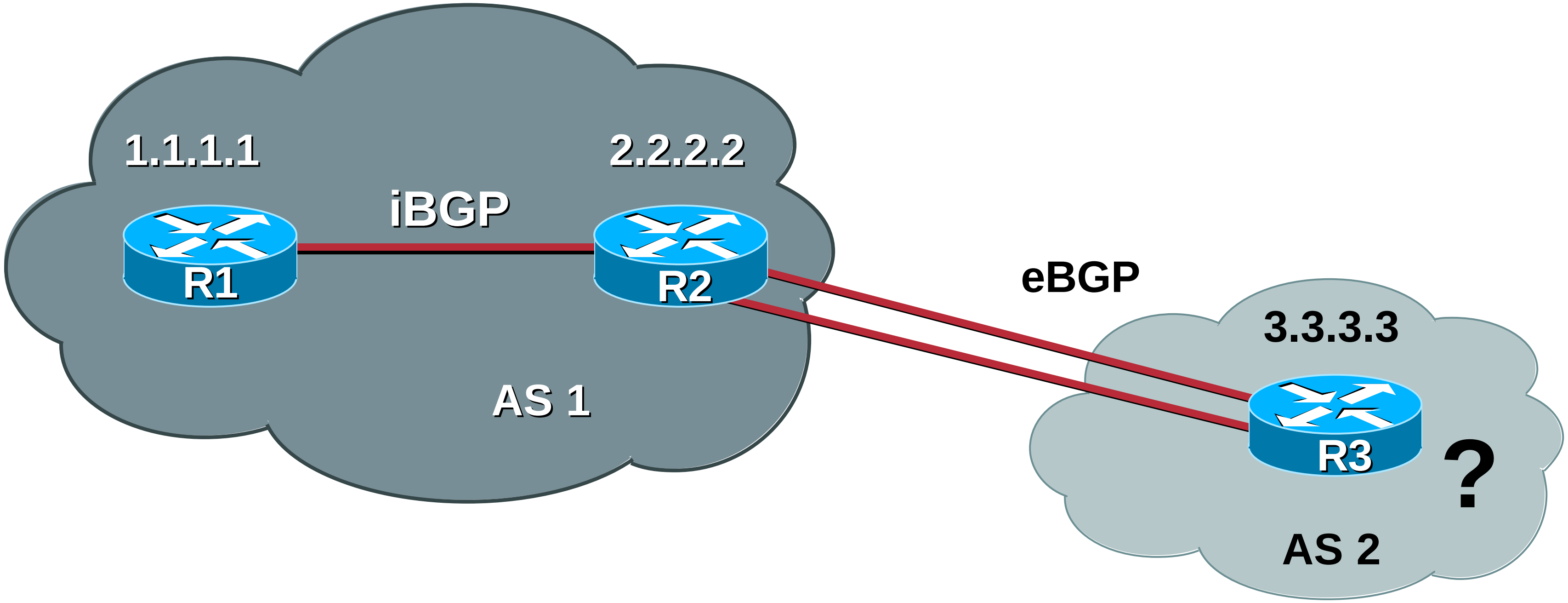
```
R2#sh ip route 1.1.1.1
Routing entry for 1.1.1.1/32
  Known via "static", distance 1, metric 0 (connected)
  * directly connected, via Serial1
      Route metric is 0, traffic share count is 1

R2#show ip interface brief | include Serial1
Serial1                 4.4.4.5        YES manual   up     up
```

Peer Establishment—iBGP

- Source address is the outgoing interface towards the destination but peering in this case is using loopback interfaces!
- Force both routers to source from the correct interface
- Use "update-source" to specify the loopback when loopback peering

```
R2#
router bgp 1
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 3.3.3.3 remote-as 2
 neighbor 3.3.3.3 update-source Loopback0
```

Peer Establishment—Diagram



- R1 is established now
- The eBGP session is still having trouble!

Peer Establishment—eBGP

- Trying to load-balance over multiple links to the eBGP peer
- Verify IP connectivity

    Check the routing table

    Use ping/trace to verify two way reachability

```
R2#ping 3.3.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

- Routing towards destination correct, but…

Peer Establishment—eBGP

```
R2#ping ip
Target IP address: 3.3.3.3
Extended commands [n]: y
Source address or interface: 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

- **Use extended pings to test loopback to loopback connectivity**
- **R3 does not have a route to our loopback, 2.2.2.2**

Peer Establishment—eBGP

Cisco.com

- Assume R3 added a route to 2.2.2.2
- Still having problems…

```
R2#sh ip bgp neigh 3.3.3.3
BGP neighbor is 3.3.3.3,  remote AS 2, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:04, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0
  Index 2, Offset 0, Mask 0x4
  0 accepted prefixes consume 0 bytes
  Prefix advertised 0, suppressed 0, withdrawn 0
  Connections established 0; dropped 0
  Last reset never
  External BGP neighbor not directly connected.
  No active TCP connection
```

Peer Establishment—eBGP


```
R2#
router bgp 1
 neighbor 3.3.3.3 remote-as 2
 neighbor 3.3.3.3 ebgp-multihop 255
 neighbor 3.3.3.3 update-source Loopback0
```

- eBGP peers are normally directly connected

  By default, TTL is set to 1 for eBGP peers

  If not directly connected, specify ebgp-multihop

- At this point, the session should come up

Peer Establishment—eBGP


```
R2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 1

Neighbor    V   AS MsgRcvd MsgSent   TblVer   InQ OutQ Up/Down   State/PfxRcd
3.3.3.3     4    2      10      26        0     0    0 never     Active
```

- **Still having trouble!**

  Connectivity issues have already been checked and corrected

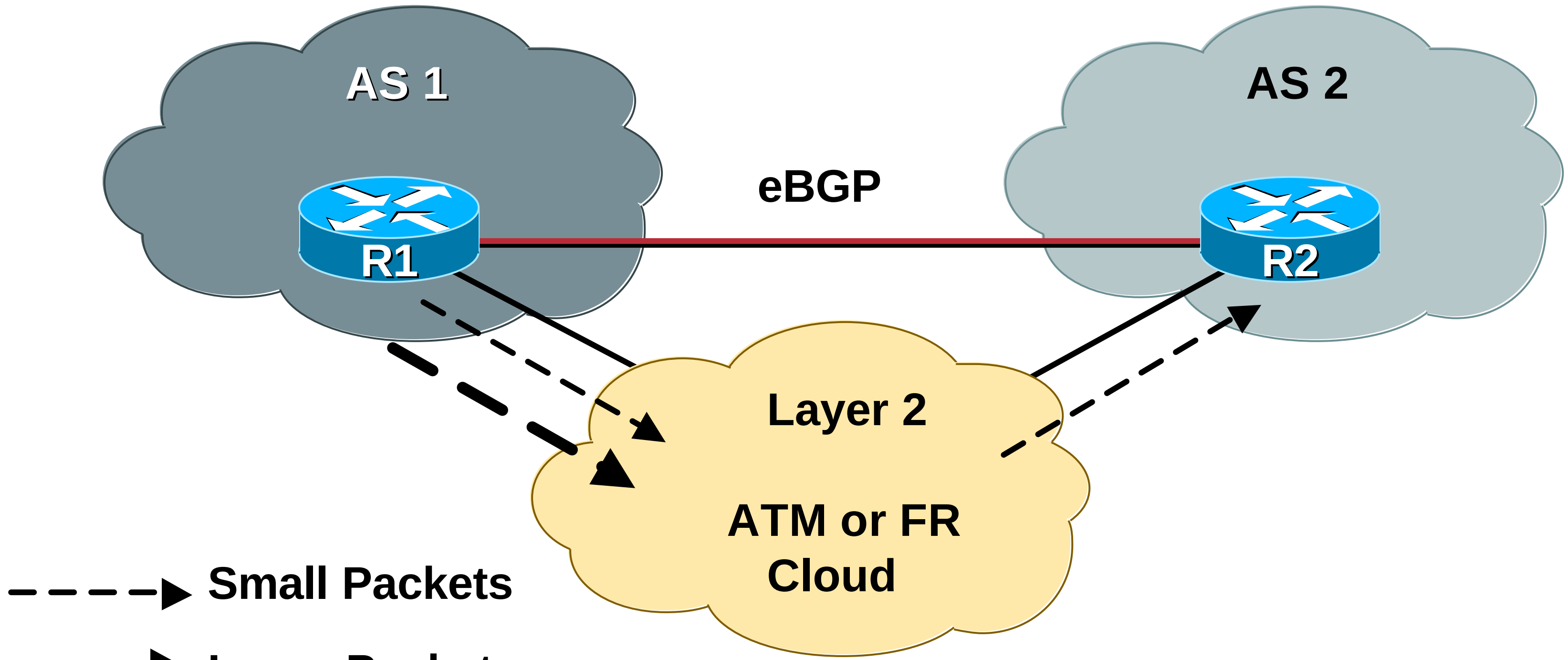
Peer Establishment—eBGP

Cisco.com

```
R2#debug ip bgp events
14:06:37: BGP: 3.3.3.3 open active, local address 2.2.2.2
14:06:37: BGP: 3.3.3.3 went from Active to OpenSent
14:06:37: BGP: 3.3.3.3 sending OPEN, version 4
14:06:37: BGP: 3.3.3.3 received NOTIFICATION 2/2
        (peer in wrong AS) 2 bytes 0001
14:06:37: BGP: 3.3.3.3 remote close, state CLOSEWAIT
14:06:37: BGP: service reset requests
14:06:37: BGP: 3.3.3.3 went from OpenSent to Idle
14:06:37: BGP: 3.3.3.3 closing
```

- **If an error is detected, a notification is sent and the session is closed**
- **R3 is configured incorrectly**
  - **Has "neighbor 2.2.2.2 remote-as 10"**
  - **Should have "neighbor 2.2.2.2 remote-as 1"**
- **After R3 makes this correction the session comes up**

Flapping Peer—Diagram



- Small packets are ok
- Large packets are lost in the cloud
- BGP session flaps

Flapping Peer

- Enable “bgp log-neighbor-changes” so you get a log message when a peer flaps
- R1 and R2 are peering over ATM cloud

```
R2#
%BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down BGP
  Notification sent
%BGP-3-NOTIFICATION: sent to neighbor 1.1.1.1 4/0
  (hold time expired) 0 bytes
R2#show ip bgp neighbor 1.1.1.1 | include Last reset
 Last reset 00:01:02, due to BGP Notification sent,
  hold time expired
```

- We are not receiving keepalives from the other side!

Flapping Peer

- **Let's take a look at our peer!**

```
R1#show ip bgp sum
BGP router identifier 172.16.175.53, local AS number 1
BGP table version is 10167, main routing table version 10167
10166 network entries and 10166 paths using 1352078 bytes of memory
1 BGP path attribute entries using 60 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 10166/300 prefixes, 10166/0 paths, scan interval 15 secs

Neighbor     V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2      4    2      53     284    10167    0   97 00:02:15        0
```

```
R1#show ip bgp summary | begin Neighbor
Neighbor     V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2      4    2      53     284    10167    0   98 00:03:04        0
```

- **Hellos are stuck in OutQ behind update packets!**
- **Notice that the MsgSent counter has not moved**

Flapping Peer


```
R1#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/21/24 m
```

```
R1#ping ip
Target IP address: 2.2.2.2
Repeat count [5]:
Datagram size [100]: 1500
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

- **Normal pings work but a ping of 1500 fails?**

Flapping Peer

- **Things to check**

  MTU values

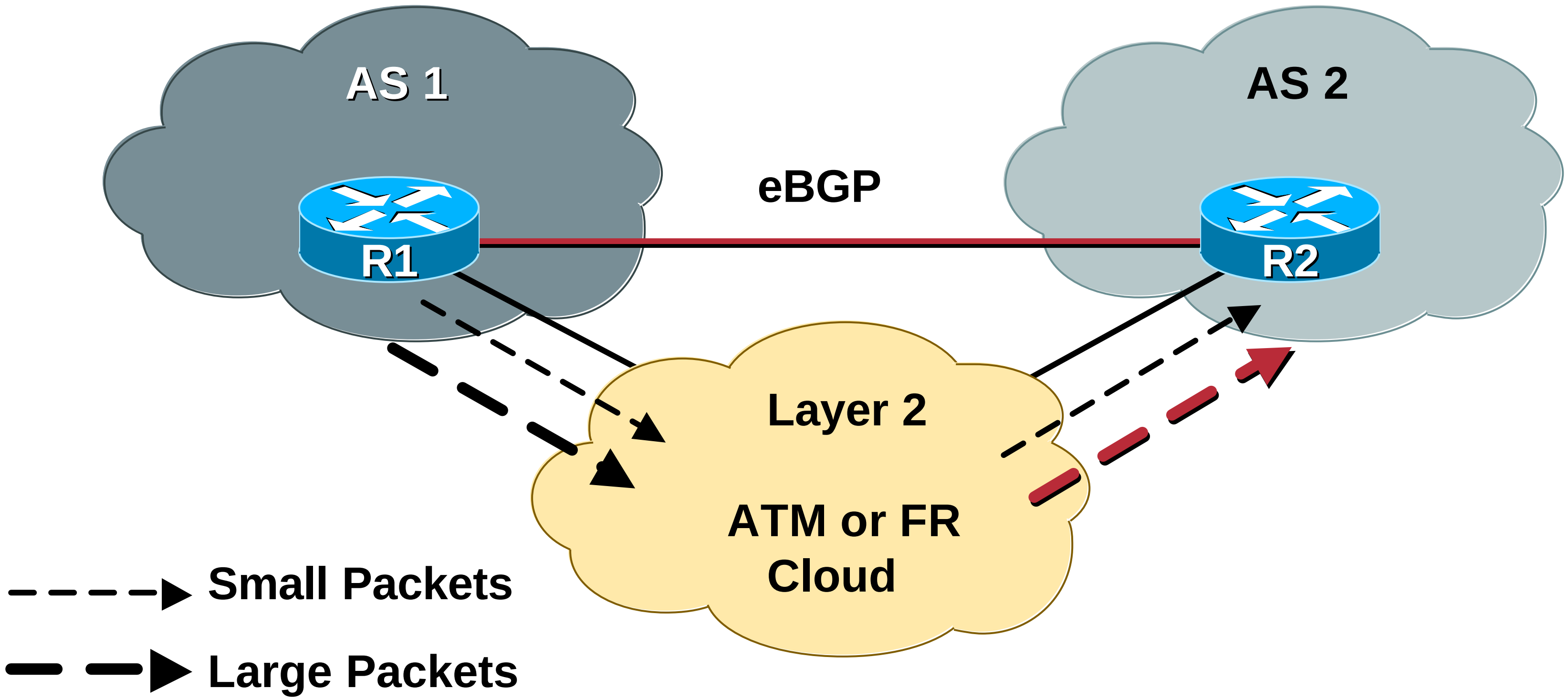
  Traffic shaping

  Rate-limiting parameters

- **Looks like a Layer 2 problem**
- **At this point we have verified that BGP is not at fault**
- **Next step is to troubleshoot layer 2…**

Flapping Peer—Diagram

Cisco.com



- **Large packets are ok now**
- **BGP session is stable!**

Troubleshooting Tips

- Extended ping/traceroute allow you to verify

  Loopback to loopback IP connectivity

  TTL issues

- "show ip bgp summary"

  Displays the state of all peers

- "show ip bgp neighbor"

  Gives a lot of information regarding the peer

Troubleshooting Tips

- **"debug ip bgp"**

  Should give you a good hint as to why a peer will not establish

- **"debug ip bgp events"**

  Displays state transitions for peers

- **"show ip bgp neighbor | include Last reset"**

  Will show you the last reset reason for all peers

Agenda

- Peer Establishment
- Missing Routes
- Inconsistent Route Selection
- Loops and Convergence Issues

Quick Review

- **Once the session has been established, UPDATEs are exchanged**
    - **All the locally known routes**
    - **Only the bestpath is advertised**
- **Incremental UPDATE messages are exchanged afterwards**

Quick Review

- **Bestpath received from eBGP peer**
  - **Advertise to all peers**
- **Bestpath received from iBGP peer**
  - **Advertise only to eBGP peers**
  - **A full iBGP mesh must exist**

Missing Routes—Agenda

- Route Origination
- UPDATE Exchange
- Filtering

Route Origination—Example I

- *All examples are with “auto-summary” enabled
- Basic network statement

```
R1# show run | begin bgp
 network 6.0.0.0
```

- BGP is not originating the route???

```
R1# show ip bgp | include 6.0.0.0
R1#
```

- Do we have a component route?

```
R1# show ip route 6.0.0.0 255.0.0.0 longer
R1#
```

Route Origination—Example I

- As soon as the RIB has a component route

```
R1# show ip route 6.0.0.0 255.0.0.0 longer
       6.0.0.0/32 is subnetted, 1 subnets
S      6.6.6.6 [1/0] via 20.100.1.6
```

- Bingo, BGP originates the route!!

```
R1# show ip bgp | include 6.0.0.0
*> 6.0.0.0  0.0.0.0        0    32768 i
```

Route Origination—Example II

- Network statement with mask

```
R1# show run | include 200.200.0.0
 network 200.200.0.0 mask 255.255.252.0
```

- BGP is not originating the route???

```
R1# show ip bgp | include 200.200.0.0
R1#
```

- Do we have the exact route?

```
R1# show ip route 200.200.0.0 255.255.252.0
% Network not in table
```

Route Origination—Example II

- Nail down routes you want to originate

```
ip route 200.200.0.0 255.255.252.0 Null 0  254
```

- Check the RIB

```
R1# show ip route 200.200.0.0 255.255.252.0
       200.200.0.0/22 is subnetted, 1 subnets
S      200.200.0.0 [1/0] via Null 0
```

- BGP originates the route!!

```
R1# show ip bgp | include 200.200.0.0
*> 200.200.0.0/220.0.0.0           0    32768
```

Route Origination—Example III

- **Trying to originate an aggregate route**

```
aggregate-address 7.7.0.0 255.255.0.0 summary-only
```

- **The RIB has a component but BGP does not create the aggregate???**

```
R1# show ip route 7.7.0.0 255.255.0.0 longer
       7.0.0.0/32 is subnetted, 1 subnets
C       7.7.7.7 [1/0] is directly connected, Loopback 0
```

```
R1# show ip bgp | i 7.7.0.0
R1#
```

Route Origination—Example III

- Remember, to have a BGP aggregate you need a BGP component, not a RIB (Routing Information Base, a.k.a. the routing table) component

```
R1# show ip bgp 7.7.0.0 255.255.0.0 longer
R1#
```

- Once BGP has a component route we originate the aggregate

```
network 7.7.7.7 mask 255.255.255.255


R1# show ip bgp 7.7.0.0 255.255.0.0 longer
*> 7.7.0.0/16  0.0.0.0              32768 i
s> 7.7.7.7/32  0.0.0.0       0      32768 i
```

- s means this component is suppressed due to the “summary-only” argument

Troubleshooting Tips

Cisco.com

- “auto-summary” rules [default]

  Network statement—must have component route (RIB)

  Network/Mask statement—must have exact route (RIB)

- “no auto-summary” rules

  Always need an exact route (RIB)

- aggregate-address looks in the BGP table, not the RIB

- “show ip route x.x.x.x y.y.y.y longer”

  Great for finding RIB component routes

- “show ip bgp x.x.x.x y.y.y.y longer”

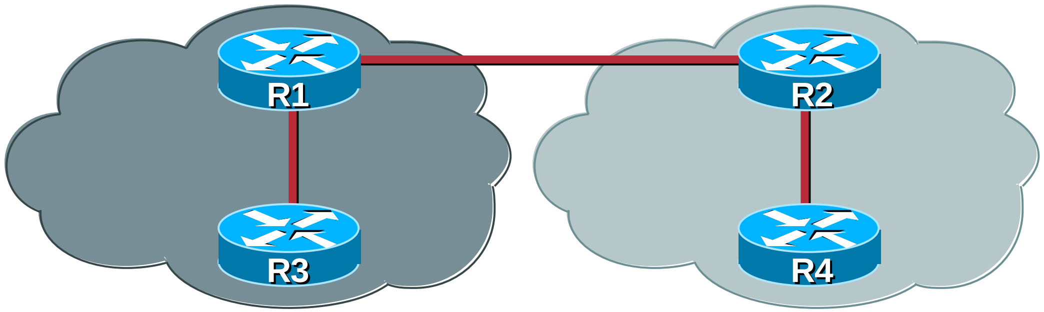
  Great for finding BGP component routes

Missing Routes

- Route Origination
- UPDATE Exchange
- Filtering

Missing Routes—Example I

- Two RR clusters
- R1 is a RR for R3
- R2 is a RR for R4
- R4 is advertising 7.0.0.0/8
- R2 has the route but R1 and R3 do not?



Missing Routes—Example I

- **First, did R2 advertise the route to R1?**

```
R2# show ip bgp neighbors 1.1.1.1 advertised-routes
BGP table version is 2, local router ID is 2.2.2.2
   Network          Next Hop            Metric LocPrf Weight Path
*>i7.0.0.0          4.4.4.4             0      100        0 I
```

- **Did R1 receive it?**

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

Missing Routes—Example I

- **Time to debug!!**

```
access-list 100 permit ip host 7.0.0.0 host 255.0.0.0
R1# debug ip bgp update 100
```

- **Tell R2 to resend his UPDATEs**

```
R2# clear ip bgp 1.1.1.1 soft out
```

- **R1 shows us something interesting**

```
*Mar  1 21:50:12.410: BGP(0): 2.2.2.2 rcv UPDATE w/ attr:
nexthop 4.4.4.4, origin i, localpref 100, metric 0,
originator 100.1.1.1, clusterlist 2.2.2.2, path , community
, extended community

*Mar  1 21:50:12.410: BGP(0): 2.2.2.2 rcv UPDATE about
7.0.0.0/8 -- DENIED due to: ORIGINATOR is us;
```

- **Cannot accept an update with our Router-ID as the ORIGINATOR_ID. Another means of loop detection in BGP**

Missing Routes—Example I

Cisco.com

- R1 and R4 have the same Router-ID

```
R1# show ip bgp summary | include identifier.
BGP router identifier 100.1.1.1, local AS number 100.
```

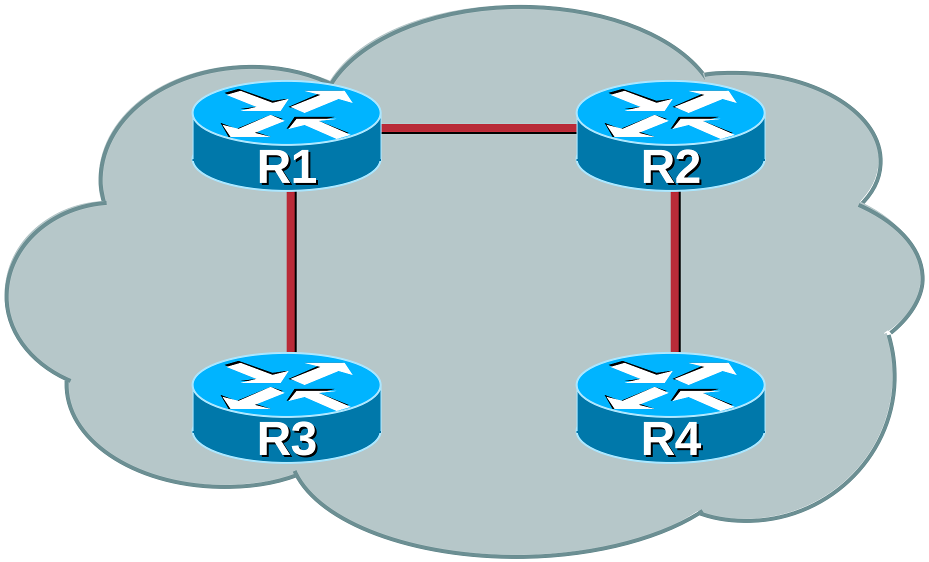
```
R4# show ip bgp summary | include identifier.
BGP router identifier 100.1.1.1, local AS number 100.
```

- Can be a problem in multicast networks; for RP (Rendezvous Point) purposes the same address may be assigned to multiple routers
- Specify a unique Router-ID

```
R1#show run | include router-id
 bgp router-id 1.1.1.1
R4# show run | include router-id
 bgp router-id 4.4.4.4
```

Missing Routes—Example II

- One RR cluster
- R1 and R2 are RRs
- R3 and R4 are RRCs
- R4 is advertising 7.0.0.0/8
  - R2 has it
  - R1 and R3 do not




```
R1#show run | include cluster
 bgp cluster-id 10
R2#show run | include cluster
 bgp cluster-id 10
```

Missing Routes—Example II

- **Same steps as last time!**
- **Did R2 advertise it to R1?**

```
R2# show ip bgp neighbors 1.1.1.1 advertised-routes
BGP table version is 2, local router ID is 2.2.2.2
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*>i7.0.0.0          4.4.4.4                  0    100      0 i
```

- **Did R1 receive it?**

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

Missing Routes—Example II

- **Time to debug!!**

```
access-list 100 permit ip host 7.0.0.0 host 255.0.0.0

R1# debug ip bgp update 100
```

- **Tell R2 to resend his UPDATEs**

```
R2# clear ip bgp 1.1.1.1 soft out
```

- **R1 shows us something interesting**

```
*Mar  3 14:28:57.208: BGP(0): 2.2.2.2 rcv UPDATE w/ attr: nexthop
4.4.4.4, origin i, localpref 100, metric 0, originator 4.4.4.4,
clusterlist 0.0.0.10, path , community , extended community

*Mar  3 14:28:57.208: BGP(0): 2.2.2.2 rcv UPDATE about 7.0.0.0/8 --
DENIED due to: reflected from the same cluster;
```

- **Remember, all RRCs must peer with all RRs in a cluster; allows R4 to send the update directly to R1**

Troubleshooting Tips

- **"show ip bgp neighbor x.x.x.x advertised-routes"**

  Lets you see a list of NLRI that you sent a peer

  Note: The attribute values shown are taken from the BGP table; attribute modifications by outbound route-maps will not be shown

- **"show ip bgp neighbor x.x.x.x routes"**

  Displays routes x.x.x.x sent to us that made it through our inbound filters

- **"show ip bgp neighbor x.x.x.x received-routes"**

  Can only use if "soft-reconfig inbound" is configured

  Displays all routes received from a peer, even those that were denied

Troubleshooting Tips

- **"clear ip bgp x.x.x.x soft in"**
  - Ask x.x.x.x to resend his UPDATEs to us
- **"clear ip bgp x.x.x.x soft out"**
  - Tells BGP to resend UPDATEs to x.x.x.x
- **"debug ip bgp update"**
  - Always use an ACL to limit output
  - Great for troubleshooting "Automatic Denies"
- **"debug ip bgp x.x.x.x update"**
  - Allows you to debug updates to/from a specific peer
  - Handy if multiple peers are sending you the same prefix

Missing Routes

- Route Origination
- UPDATE Exchange
- Filtering

Update Filtering

- Type of filters
  - Prefix filters
  - AS_PATH filters
  - Community filters
  - Route-maps
- Applied incoming and/or outgoing

Missing Routes—Update Filters

- Determine which filters are applied to the BGP session

  show ip bgp neighbors x.x.x.x

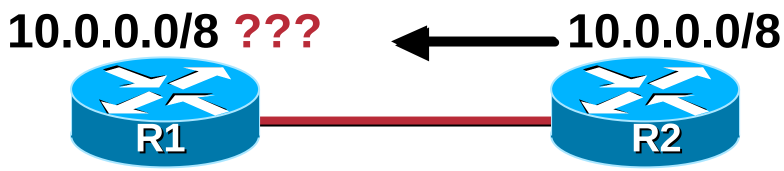
  show run | include neighbor x.x.x.x

- Examine the route and pick out the relevant attributes

  show ip bgp x.x.x.x

- Compare the attributes against the filters

Missing Routes—Update Filters



- Missing 10.0.0.0/8 in R1 (1.1.1.1)
- Not received from R2 (2.2.2.2)

```
R1#show ip bgp neigh 2.2.2.2 routes

Total number of prefixes 0
```

Missing Routes—Update Filters

Cisco.com

- R2 originates the route
- Does not advertise it to R1

```
R2#show ip bgp neigh 1.1.1.1 advertised-routes
   Network          Next Hop            Metric LocPrf Weight Path
```

```
R2#show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, 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
```

Missing Routes—Update Filters

- Time to check filters!
- ^ matches the beginning of a line
- $ matches the end of a line
- ^$ means match any empty AS_PATH
- Filter "looks" correct

```
R2#show run | include neighbor 1.1.1.1
 neighbor 1.1.1.1 remote-as 3
 neighbor 1.1.1.1 filter-list 1 out

R2#sh ip as-path 1
  AS path access list 1
    permit ^$
```

Missing Routes—Update Filters


```
R2#show ip bgp filter-list 1

R2#show ip bgp regexp ^$
BGP table version is 1661, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.0         0.0.0.0              0         32768 i
```

- Nothing matches the filter-list???
- Re-typing the regexp gives the expected output

Missing Routes—Update Filters

- **Copy and paste the entire regexp line from the configuration**

R2#show ip bgp regexp ^$

Nothing matches again! Let's use the up arrow key to see where the cursor stops

R2#show ip bgp regexp ^$ █

End of Line Is at the Cursor

- **There is a trailing white space at the end**
- **It is considered part of the regular expression**

Missing Routes—Update Filters

- Force R2 to resend the update after the filter-list correction
- Then check R1 to see if he has the route

```
R2#clear ip bgp 1.1.1.1 soft out

R1#show ip bgp 10.0.0.0
% Network not in table
```

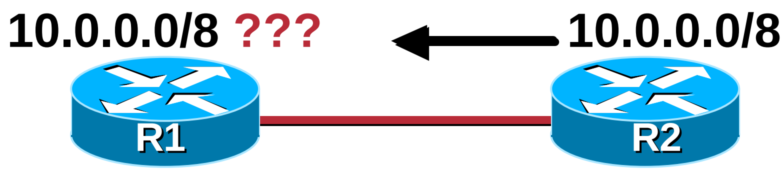
- R1 still does not have the route
- Time to check R1's inbound policy for R2

Missing Routes—Update Filters

Cisco.com

```
R1#show run | include neighbor 2.2.2.2
 neighbor 2.2.2.2 remote-as 12
 neighbor 2.2.2.2 route-map POLICY in
R1#show route-map POLICY
route-map POLICY, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100 101
    as-path (as-path filter): 1
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
R1#show access-list 100
Extended IP access list 100
    permit ip host 10.0.0.0 host 255.255.0.0
R1#show access-list 101
Extended IP access list 101
    permit ip 200.1.0 0.0.0.255 host 255.255.255.0
R1#show ip as-path 1
AS path access list 1
    permit ^12$
```

Missing Routes—Update Filters



- Confused? Let’s run some debugs

```
R1#show access-list 99
Standard IP access list 99
    permit 10.0.0.0

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

R1#
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 10.0.0.0/8 -- DENIED due to: route-map;
```

Missing Routes—Update Filters


```
R1#sh run | include neighbor 2.2.2.2
 neighbor 2.2.2.2 remote-as 12
 neighbor 2.2.2.2 route-map POLICY in
R1#sh route-map POLICY
route-map POLICY, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100 101
    as-path (as-path filter): 1
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
R1#sh access-list 100
Extended IP access list 100
    permit ip host 10.0.0.0 host 255.255.0.0
R1#sh access-list 101
Extended IP access list 101
    permit ip 200.1.1.0 0.0.0.255 host 255.255.255.0
R1#sh ip as-path 1
AS path access list 1
    permit ^12$
```

Missing Routes—Update Filters

Cisco.com

- **Wrong mask! Needs to be /8 and the ACL allows a /16 only!**

  Extended IP access list 100

  permit ip host 10.0.0.0 host 255.255.0.0

- **Should be**

  Extended IP access list 100

  permit ip host 10.0.0.0 host 255.0.0.0

- **Use prefix-list instead, more difficult to make a mistake**

  ip prefix-list my_filter permit 10.0.0.0/8

- **What about ACL 101?**

  Multiple matches on the same line are ORed

  Multiple matches on different lines are ANDed

- **ACL 101 does not matter because ACL 100 matches which satisfies the OR condition**

Troubleshooting Tips

- "show ip as-path-access-list"

  Displays the filter

- "show ip bgp filter-list"

  Displays BGP paths that match the filter

- "show ip bgp regexp"

  Displays BGP paths that match the as-path regular expression; handy for troubleshooting filter-list issues

Troubleshooting Tips

- "show ip community-list"

  Displays the filter

- "show ip bgp community-list"

  Displays BGP paths that match the filter

- "show ip prefix-list"

  Displays the filter

  Prefix-list are generally easier to use than ACLs

- "show ip bgp prefix-list"

  Displays BGP paths that match the filter

Troubleshooting Tips

- "show route-map"

  Displays the filter

- "show ip bgp route-map"

  Displays BGP paths that match the filter

- "show access-list"

  Displays the filter

- debug ip bgp update ACL

  After going through the config, debug!

  Don't forget the ACL

Agenda

Cisco.com

- Peer Establishment
- Missing Routes
- Inconsistent Route Selection
- Loops and Convergence Issues

Inconsistent Route Selection

- Two common problems with route selection
  - Inconsistency
  - Appearance of an incorrect decision
- RFC 1771 defines the decision algorithm
- Every vendor has tweaked the algorithm
  - http://www.cisco.com/warp/public/459/25.shtml
- Route selection problems can result from oversights by RFC 1771

Inconsistent—Example I

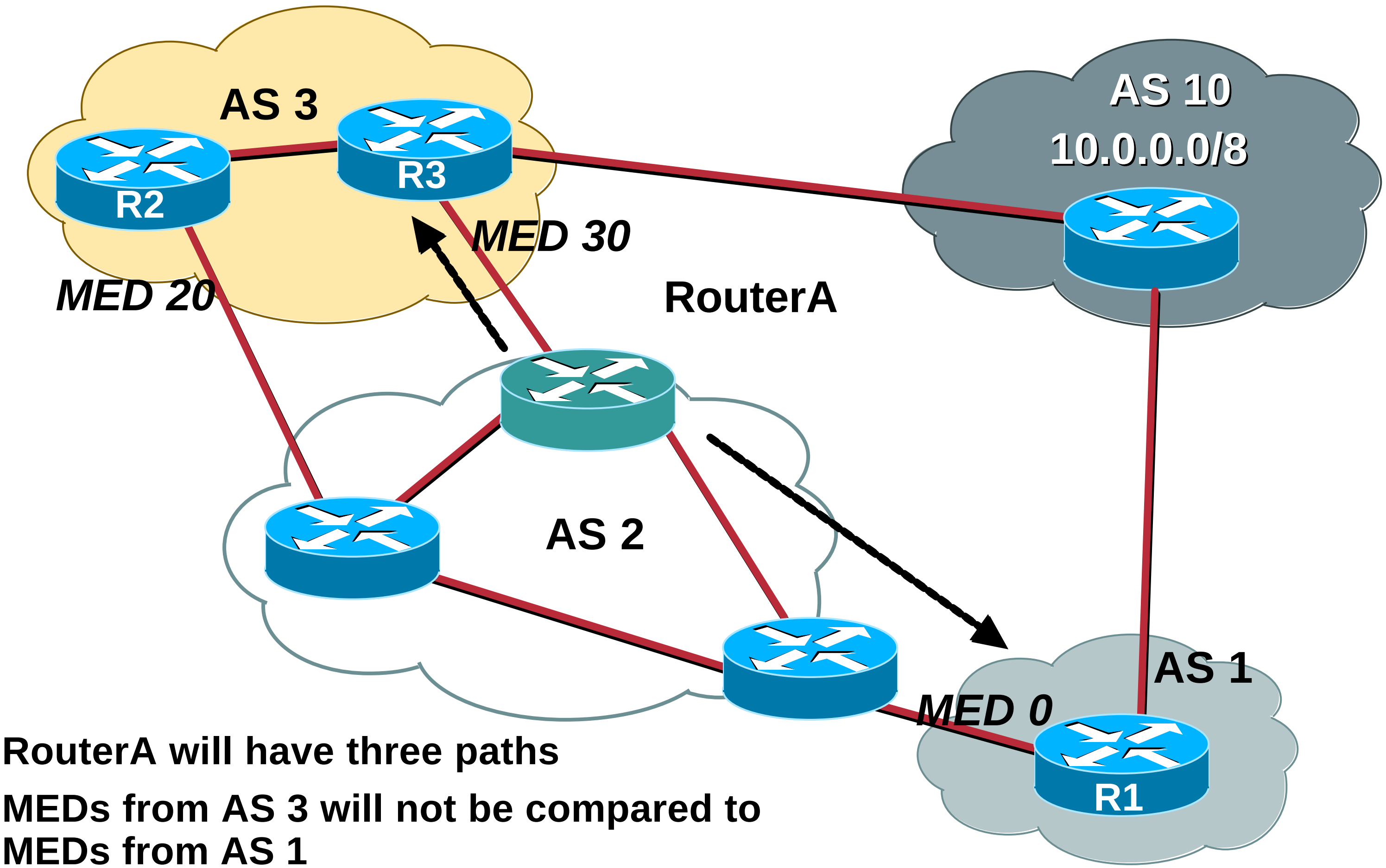
- **RFC says that MED is not always compared**
- **As a result, the ordering of the paths can effect the decision process**
- **By default, the prefixes are compared in order of arrival (most recent to oldest)**
  - **Use bgp deterministic-med to order paths consistently**
  - **The bestpath is recalculated as soon as the command is entered**
  - **Enable in all the routers in the AS**

Inconsistent—Example I

Cisco.com

- Inconsistent route selection may cause problems
  - Routing loops
  - Convergence loops—i.e. the protocol continuously sends updates in an attempt to converge
  - Changes in traffic patterns
- Difficult to catch and troubleshoot
- It is best to avoid the problem in the first place
  - bgp deterministic-med

Symptom I—Diagram



- RouterA will have three paths
- MEDs from AS 3 will not be compared to MEDs from AS 1
- RouterA will sometimes select the path from R1 as best and but may also select the path from R3 as best

Inconsistent—Example I

Cisco.com

```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #3, advertised over iBGP, eBGP)
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

- **Initial State**

  Path 1 beats Path 2—Lower MED

  Path 3 beats Path 1—Lower Router-ID

Inconsistent—Example I


```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #3, advertised over iBGP, eBGP)
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external, best
```

- **1.1.1.1 bounced so the paths are re-ordered**

    Path 1 beats Path 2—Lower Router-ID

    Path 3 beats Path 1—External vs Internal

Deterministic MED—Operation

- The paths are ordered by Neighbor AS
- The bestpath for each Neighbor AS group is selected
- The overall bestpath results from comparing the winners from each group
- The bestpath will be consistent because paths will be placed in a deterministic order

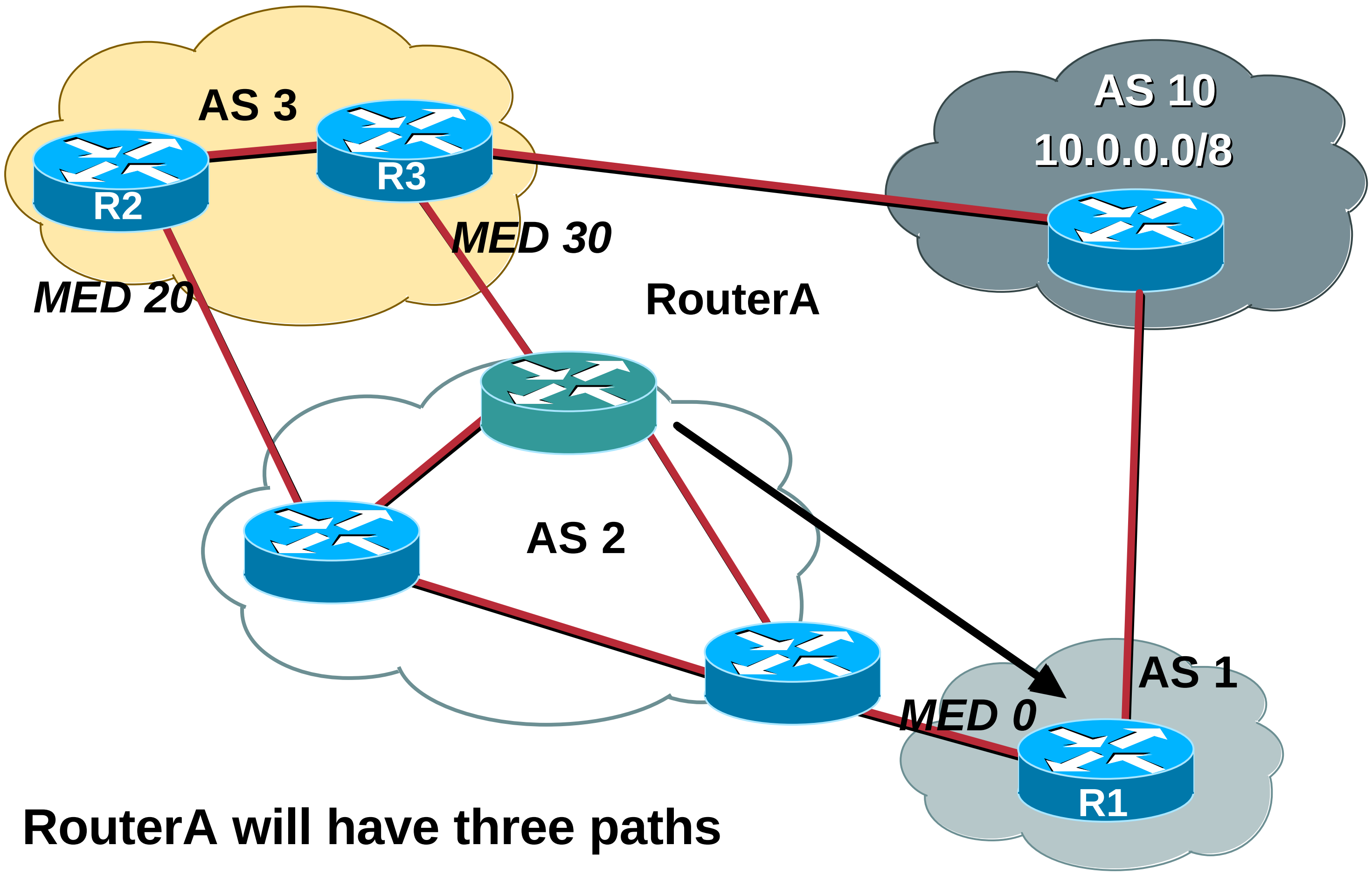
Deterministic MED—Result


```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #1, advertised over iBGP, eBGP)
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external
```

- **Path 1 is best for AS 1**
- **Path 2 beats Path 3 for AS 3—Lower MED**
- **Path 1 beats Path 2—Lower Router-ID**

Solution—Diagram

Cisco.com



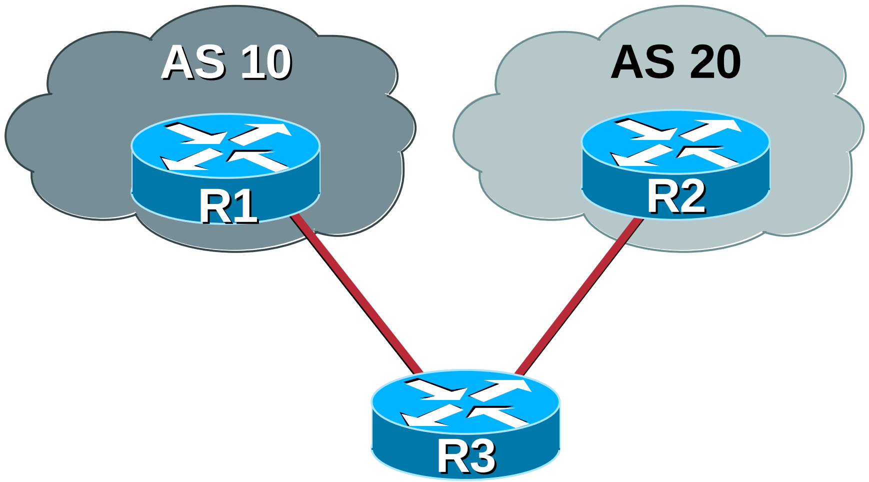
- **RouterA will have three paths**
- **RouterA will consistently select the path from R1 as best!**

Deterministic MED—Summary

- Always use "bgp deterministic-med"
- Need to enable throughout entire network at roughly the same time
- If only enabled on a portion of the network routing loops and/or convergence problems may become more severe
- As a result, default behavior cannot be changed so the knob must be configured by the user

Inconsistent—Example II

- The bestpath changes every time the peering is reset




```
R3#show ip bgp 7.0.0.0
BGP routing table entry for 7.0.0.0/8, version 15
  10 100
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, external
  20 100
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Inconsistent—Example II

Cisco.com

```
R3#show ip bgp 7.0.0.0
BGP routing table entry for 7.0.0.0/8, version 17
Paths: (2 available, best #2)
  Not advertised to any peer
  20 100
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 0, localpref 100, valid, external
  10 100
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, external, best
```

- **The "oldest" external is the bestpath**
  - **All other attributes are the same**
  - **Stability enhancement!!—CSCdk12061—Integrated in 12.0(1)**
- **"bgp bestpath compare-router-id" will disable this enhancement—CSCdr47086—Integrated in 12.0(11)S and 12.1(3)**

Inconsistent—Example III

Cisco.com

```
R1#sh ip bgp 11.0.0.0
BGP routing table entry for 11.0.0.0/8, version 10
  100
    1.1.1.1 from 1.1.1.1
      Origin IGP, localpref 120, valid, internal
  100
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 0, localpref 100, valid, external, best
```

- **Path 1 has higher localpref but path 2 is better???**
- **This appears to be incorrect…**

Inconsistent—Example III

- Path is from an internal peer which means the path must be synchronized by default
- Check to see if sync is on or off

```
R1# show run | include sync
R1#
```

- Sync is still enabled, check for IGP path:

```
R1# show ip route 11.0.0.0
% Network not in table
```

- CSCdr90728 “BGP: Paths are not marked as not synchronized”—Fixed in 12.1(4)
- Path 1 is not synchronized
- Router made the correct choice

Troubleshooting Tips

- **“show run | include sync”**

    Quick way to see if synchronization is enabled

- **“show run | include bgp”**

    Will show you what bestpath knobs you have enabled (bgp deterministic-med, bgp always-compare-med, etc.)

- **“show ip bgp x.x.x.x”**

    Go through the decision algorithm step-by-step

    Understand why the bestpath is the best

Agenda

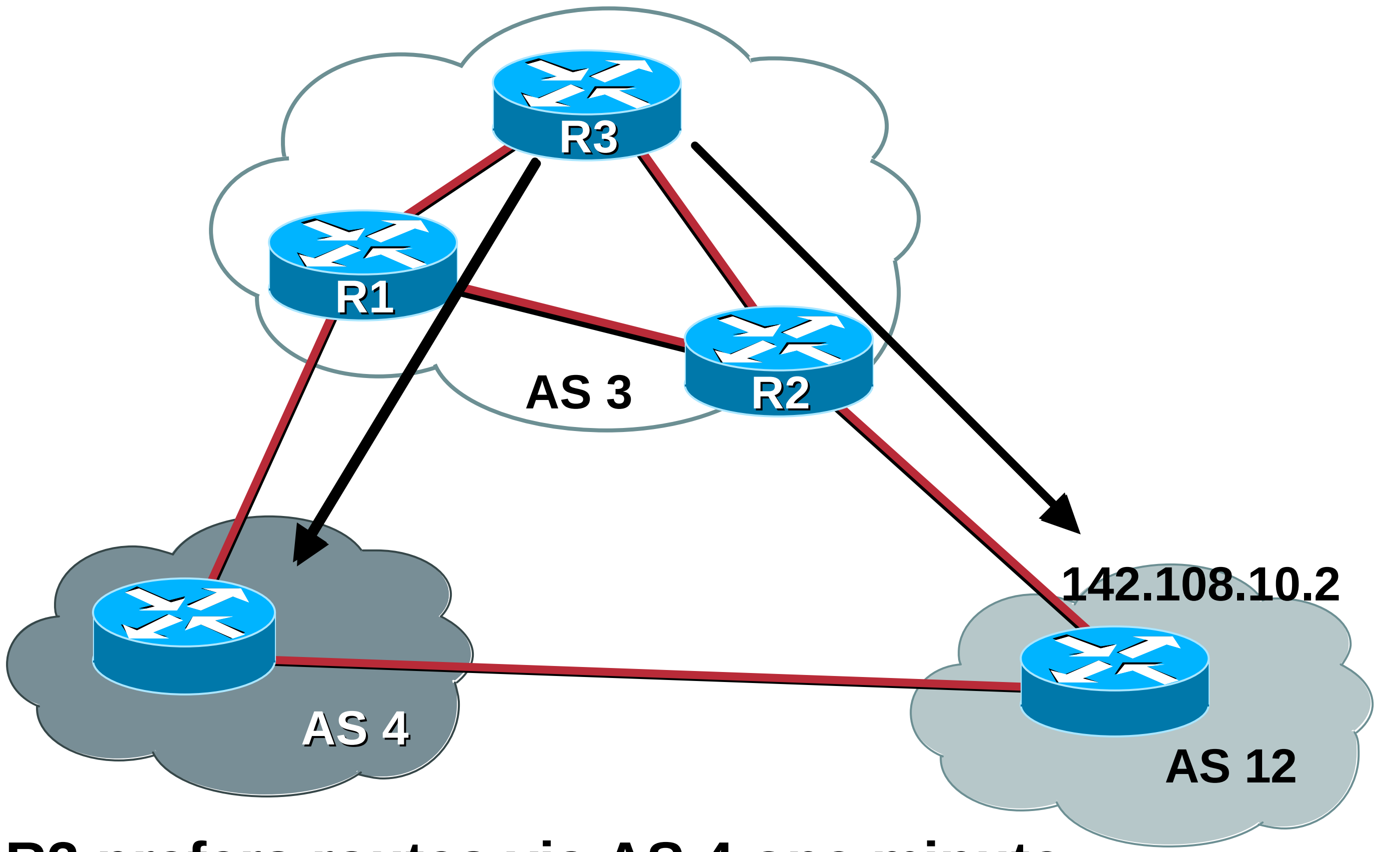
Cisco.com

- Peer Establishment
- Missing Routes
- Inconsistent Route Selection
- Loops and Convergence Issues

Route Oscillation

- One of the most common problems!
- Every minute routes flap in the routing table from one nexthop to another
- With full routes the most obvious symptom is high CPU in "BGP Router" process

Route Oscillation—Diagram



- R3 prefers routes via AS 4 one minute
- BGP scanner runs then R3 prefers routes via AS 12
- The entire table oscillates every 60 seconds

Route Oscillation—Symptom


```
R3#show ip bgp summary
BGP router identifier 3.3.3.3, local AS number 3
BGP table version is 502, main routing table version 502
267 network entries and 272 paths using 34623 bytes of memory

R3#sh ip route summary | begin bgp
bgp 3          4         6            520        1400
  External: 0 Internal: 10 Local: 0
internal       5                                 5800
Total          10        263          13936      43320
```

- **Watch for:**

  Table version number incrementing rapidly

  Number of networks/paths or external/internal routes changing

Route Oscillation—Troubleshooting

- Pick a route from the RIB that has changed within the last minute
- Monitor that route to see if it changes every minute

```
R3#show ip route 156.1.0.0
Routing entry for 156.1.0.0/16
  Known via "bgp 3", distance 200, metric 0
 Routing Descriptor Blocks:
  * 1.1.1.1, from 1.1.1.1, 00:00:53 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2, BGP network version 474
```

```
R3#show ip bgp 156.1.0.0
BGP routing table entry for 156.1.0.0/16, version 474
Paths: (2 available, best #1)
  Advertised to non peer-group peers:
    2.2.2.2
  4 12
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, internal, best
  12
    142.108.10.2 (inaccessible) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal
```

Route Oscillation—Troubleshooting

- Check again after bgp_scanner runs
- bgp_scanner runs every 60 seconds and validates reachability to all nexthops

```
R3#sh ip route 156.1.0.0
Routing entry for 156.1.0.0/16
  Known via "bgp 3", distance 200, metric 0
    Routing Descriptor Blocks:
  * 142.108.10.2, from 2.2.2.2, 00:00:27 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 478

R3#sh ip bgp 156.1.0.0
BGP routing table entry for 156.1.0.0/16, version 478
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    1.1.1.1
  4 12
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, internal
  12
    142.108.10.2 from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation—Troubleshooting

- Lets take a closer look at the nexthop

```
R3#show ip route 142.108.10.2
Routing entry for 142.108.0.0/16
  Known via "bgp 3", distance 200, metric 0
 Routing Descriptor Blocks:
  * 142.108.10.2, from 2.2.2.2, 00:00:50 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 476

R3#show ip bgp 142.108.10.2
BGP routing table entry for 142.108.0.0/16, version 476
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    1.1.1.1
  4 12
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, internal
  12
    142.108.10.2 from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation—Troubleshooting

Cisco.com

- BGP nexthop is known via BGP
- Illegal recursive lookup
- Scanner will notice and install the other path in the RIB

```
R3#sh debug
  BGP events debugging is on
  BGP updates debugging is on
  IP routing debugging is on
R3#
BGP: scanning routing tables
BGP: nettable_walker 142.108.0.0/16 calling revise_route
RT: del 142.108.0.0 via 142.108.10.2, bgp metric [200/0]
BGP: revise route installing 142.108.0.0/16 -> 1.1.1.1
RT: add 142.108.0.0/16 via 1.1.1.1, bgp metric [200/0]
RT: del 156.1.0.0 via 142.108.10.2, bgp metric [200/0]
BGP: revise route installing 156.1.0.0/16 -> 1.1.1.1
RT: add 156.1.0.0/16 via 1.1.1.1, bgp metric [200/0]
```

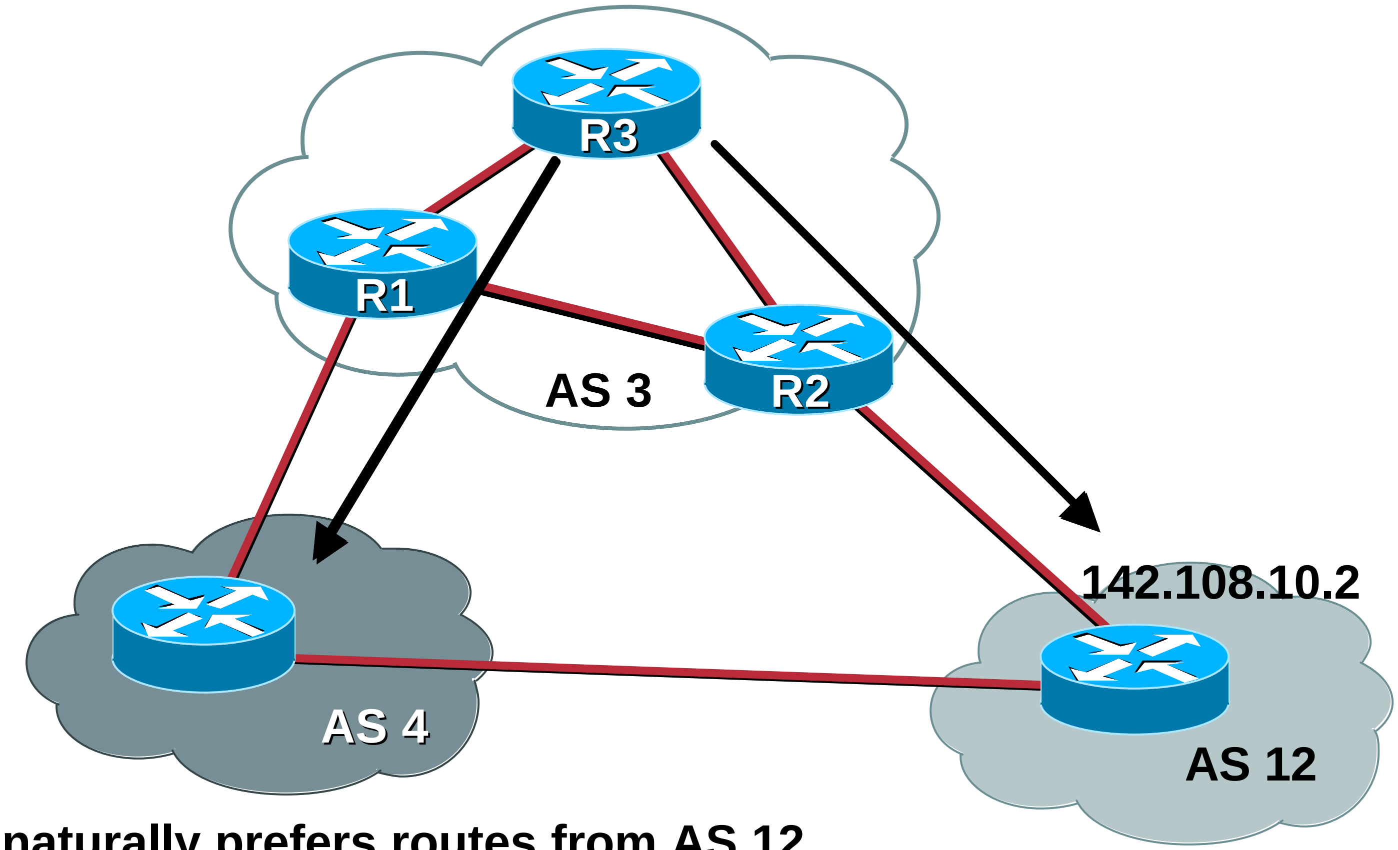
Route Oscillation—Troubleshooting

Cisco.com

- Route to the nexthop is now valid
- Scanner will detect this and re-install the other path
- Routes will oscillate forever

```
R3#
BGP: scanning routing tables
BGP:  ip nettable_walker 142.108.0.0/16 calling revise_route
RT: del 142.108.0.0 via 1.1.1.1, bgp metric [200/0]
BGP: revise route installing 142.108.0.0/16 -> 142.108.10.2
RT: add 142.108.0.0/16 via 142.108.10.2, bgp metric [200/0]
BGP: nettable_walker 156.1.0.0/16 calling revise_route
RT: del 156.1.0.0 via 1.1.1.1, bgp metric [200/0]
BGP: revise route installing 156.1.0.0/16 -> 142.108.10.2
RT: add 156.1.0.0/16 via 142.108.10.2, bgp metric [200/0]
```

Route Oscillation—Step by Step



- R3 naturally prefers routes from AS 12
- R3 does not have an IGP route to 142.108.10.2 which is the next-hop for routes learned via AS 12
- R3 learns 142.108.0.0/16 via AS 4 so 142.108.10.2 becomes reachable

Route Oscillation—Step by Step

- R3 then prefers the AS 12 route for 142.108.0.0/16 whose next-hop is 142.108.10.2
- This is an illegal recursive lookup
- BGP detects the problem when scanner runs and flags 142.108.10.2 as inaccessible
- Routes through AS 4 are now preferred
- The cycle continues forever…

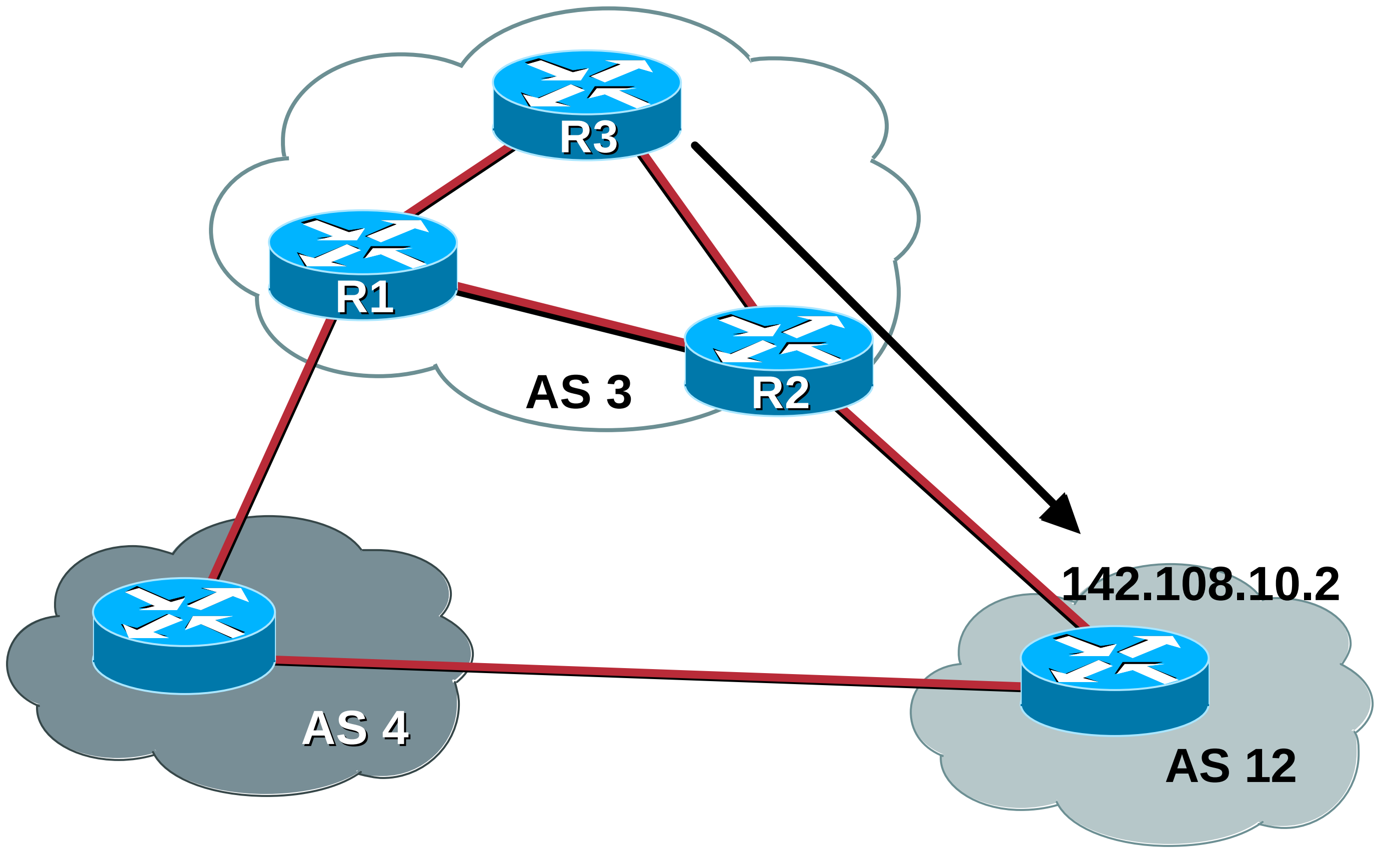
Route Oscillation—Solution

- iBGP preserves the next-hop information from eBGP
- To avoid problems

  Use “next-hop-self” for iBGP peering

  Make sure you advertise the next-hop prefix via the IGP

Route Oscillation—Solution

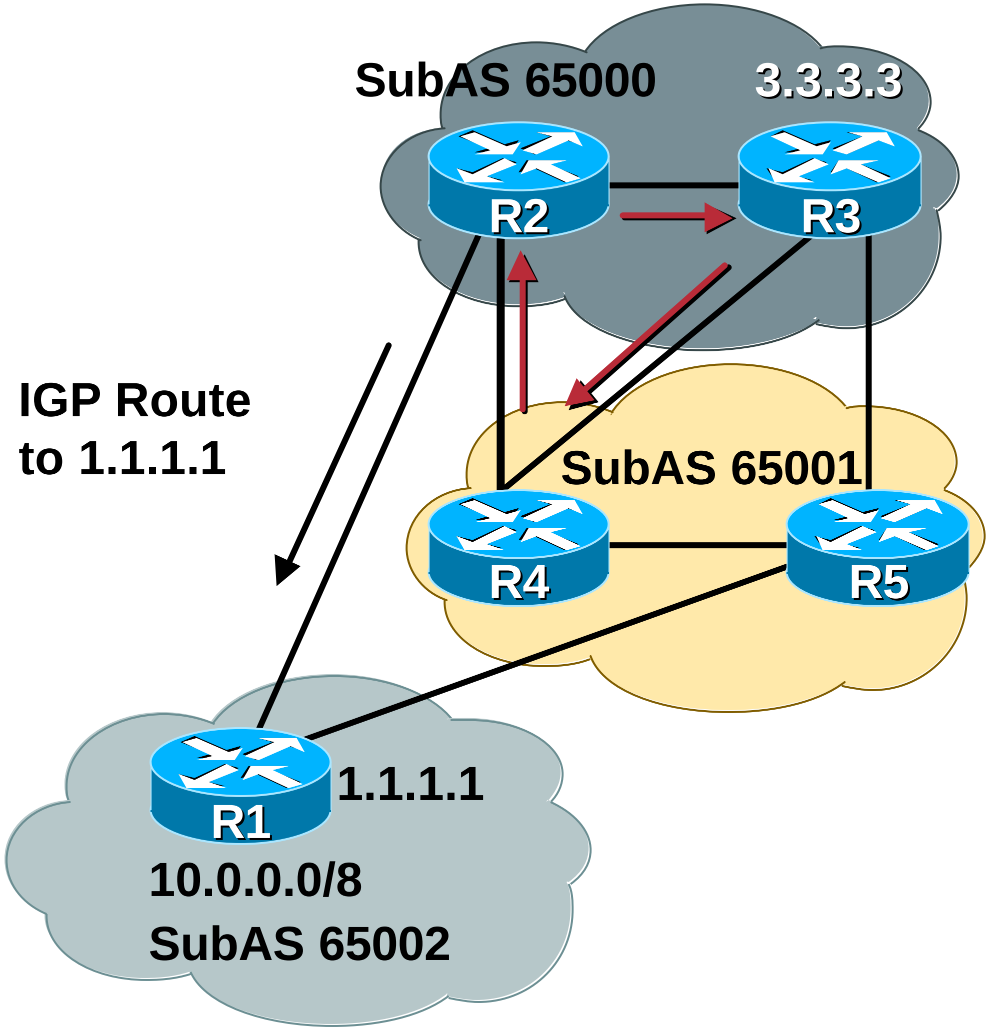


- R3 now has IGP route to AS 12 next-hop or R2 is using next-hop-self
- R3 now prefers routes via AS 12 all the time
- No more oscillation!!

Routing Loop

R5# traceroute 10.1.1.1

1 30.100.1.1
2 20.20.20.4 - R3
3 30.1.1.26 - R4
4 30.1.1.17 - R2
5 20.20.20.4 - R3
6 30.1.1.26 - R4
7 30.1.1.17 - R2
8 20.20.20.4
9 30.1.1.26
10 30.1.1.17



- Traffic loops between R3, R4, and R2

Routing Loop

- **First capture a "show ip route" from the three problem routers**
- **R3 is forwarding traffic to 1.1.1.1 (R1)**

```
R3# show ip route 10.1.1.1
Routing entry for 10.0.0.0/8
  Known via "bgp 65000", distance 200, metric 0
  Routing Descriptor Blocks:
    1.1.1.1, from 5.5.5.5, 01:46:43 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0, BGP network version 0
  * 1.1.1.1, from 4.4.4.4, 01:46:43 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0, BGP network version 0
```

Routing Loop

- R4 is also forwarding to 1.1.1.1 (R1)

```
R4# show ip route 10.1.1.1
Routing entry for 10.0.0.0/8
  Known via "bgp 65001", distance 200, metric 0
  Routing Descriptor Blocks:
  * 1.1.1.1, from 5.5.5.5, 01:47:02 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

Routing Loop

- R2 is forwarding to 3.3.3.3? (R3)

```
R2# show ip route 10.1.1.1
Routing entry for 10.0.0.0/8
  Known via "bgp 65000", distance 200, metric 0
Routing Descriptor Blocks:
  * 3.3.3.3, from 3.3.3.3, 01:47:00 ago
        Route metric is 0, traffic share count is 1
        AS Hops 0, BGP network version 3
```

- Very odd that the NEXT_HOP is in the middle of the network

Routing Loop

Cisco.com

- **Verify BGP paths on R2**

```
R2#show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 3
Paths: (4 available, best #1)
  Advertised to non peer-group peers:
    1.1.1.1 5.5.5.5 4.4.4.4
  (65001 65002)
    3.3.3.3 (metric 11) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, confed-internal,
best
 (65002)
    1.1.1.1 (metric 50) from 1.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, confed-external
```

- **R3 path is better than R1 path because of IGP cost to the NEXT_HOP**
- **R3 is advertising the path to us with a NEXT_HOP of 3.3.3.3 ???**

Routing Loop

Cisco.com

- **What is R3 advertising?**

```
R3# show ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 3
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  5.5.5.5 2.2.2.2
  (65001 65002)
    1.1.1.1 (metric 5031) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, confed-external, best, multipath
  (65001 65002)
    1.1.1.1 (metric 5031) from 5.5.5.5 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, valid, confed-external, multipath
```

- **Hmmm, R3 is using multipath to load-balance**

```
R3#show run | i maximum
 maximum-paths 6
```

Routing Loop

- “maximum-paths” tells the router to reset the NEXT_HOP to himself
  - R3 sets NEXT_HOP to 3.3.3.3
- Forces traffic to come to him so he can load-balance
- Is typically used for multiple eBGP sessions to an AS
  - Be careful when using in Confederations!!
- Need to make R2 prefer the path from R1 to prevent the routing loop
  - Make IGP metric to 1.1.1.1 better than IGP metric to 4.4.4.4

Troubleshooting Tips

- High CPU in “Router BGP” is normally a sign of a convergence problem
- Find a prefix that changes every minute

  show ip route | include , 00:00

- Troubleshoot/debug that one prefix

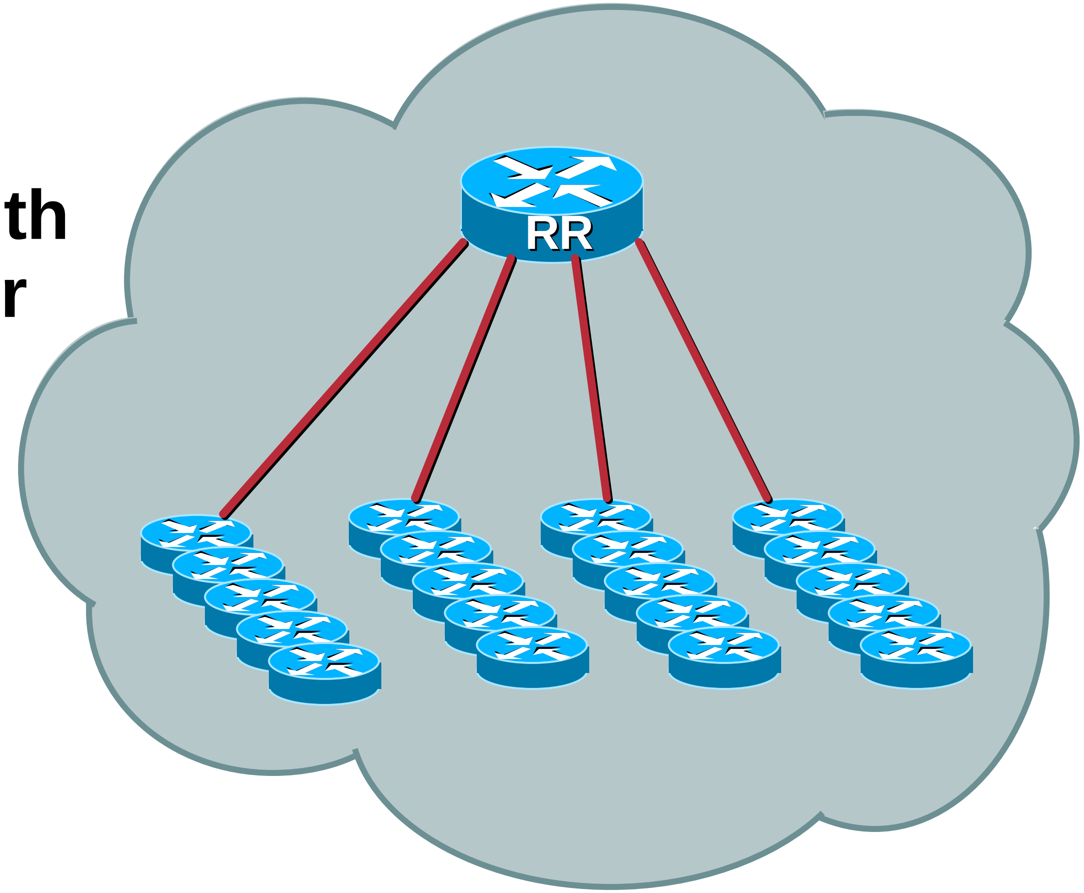
Troubleshooting Tips

- **BGP routing loop?**
  - First, check for IGP routing loops to the BGP NEXT_HOPs
- **BGP loops are normally caused by**
  - Not following physical topology in RR environment
  - Multipath with confederations
  - Lack of a full iBGP mesh
- **Get the following from each router in the loop path**
  - show ip route x.x.x.x
  - show ip bgp x.x.x.x
  - show ip route NEXT_HOP

Convergence Problems

Cisco.com

- Route reflector with 250 route reflector clients
- 100k routes
- BGP will not converge



Convergence Problems

Cisco.com

- Have been trying to converge for 10 minutes
- Peers keep dropping so we never converge?

```
RR# show ip bgp summary
Neighbor      V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
20.3.1.160    4   100      10    5416     9419    0    0 00:00:12 Closing
20.3.1.161    4   100      11    4418     8055    0  335 00:10:34        0
20.3.1.162    4   100      12    4718     8759    0  128 00:10:34        0
20.3.1.163    4   100       9    3517        0    1    0 00:00:53 Connect
20.3.1.164    4   100      13    4789     8759    0  374 00:10:37        0
20.3.1.165    4   100      13    3126        0    0  161 00:10:37        0
20.3.1.166    4   100       9    5019     9645    0    0 00:00:13 Closing
20.3.1.167    4   100       9    6209     9218    0  350 00:10:38        0
```

- Check the log to find out why

```
RR#show log | i BGP
*May  3 15:27:16: %BGP-5-ADJCHANGE: neighbor 20.3.1.118 Down— BGP Notification sent
*May  3 15:27:16: %BGP-3-NOTIFICATION: sent to neighbor 20.3.1.118 4/0 (hold time expired) 0 bytes
*May  3 15:28:10: %BGP-5-ADJCHANGE: neighbor 20.3.1.52 Down— BGP Notification sent
*May  3 15:28:10: %BGP-3-NOTIFICATION: sent to neighbor 20.3.1.52 4/0 (hold time expired) 0 bytes
```

Convergence Problems

Cisco.com

- We are either missing hellos or our peers are not sending them
- Check for interface input drops

```
RR# show interface gig 2/0 | include input drops
Output queue 0/40, 0 drops; input queue 0/75, 72390 drops
RR#
```

- 72k drops will definitely cause a few peers to go down
- We are missing hellos because the interface input queue is very small
- A rush of TCP Acks from 250 peers can fill 75 spots in a hurry
- Increase the size of the queue

```
RR# show run interface gig 2/0
interface GigabitEthernet 2/0
 ip address 7.7.7.156 255.255.255.0
 hold-queue 2000 in
```

Convergence Problems

- **Let's start over and give BGP another chance**

```
RR# clear ip bgp *
RR#
```

- **No more interface input drops**

```
RR# show interface gig 2/0 | include input drops
Output queue 0/40, 0 drops; input queue 0/2000, 0 drops
RR#
```

- **Our peers are stable!!**

```
RR# show log | include BGP
RR#
```

Convergence Problems

- BGP converged in 25 minutes
- Still seems like a long time
- What was TCP doing?

```
RR#show tcp stat | begin Sent:
Sent: 1666865 Total, 0 urgent packets
      763 control packets (including 5 retransmitted)
      1614856 data packets (818818410 bytes)
      39992 data packets (13532829 bytes) retransmitted
      6548 ack only packets (3245 delayed)
      1 window probe packets, 2641 window update packets
```

```
RR#show ip bgp neighbor | include max data segment
Datagrams (max data segment is 536 bytes):
```

Convergence Problems

Cisco.com

- 1.6 Million packets is high
- 536 is the default MSS (max segment size) for a TCP connection
- Very small considering the amount of data we need to transfer

```
RR#show ip bgp neighbor | include max data segment
Datagrams (max data segment is 536 bytes):
Datagrams (max data segment is 536 bytes):
```

- Enable path mtu discovery
- Sets MSS to max possible value

```
RR#show run | include tcp
ip tcp path-mtu-discovery
RR#
```

Convergence Problems

- Restart the test one more time

```
RR# clear ip bgp *
RR#
```

- MSS looks a lot better

```
RR#show ip bgp neighbor | include max data segment
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
```

Convergence Problems

- TCP sent 1 million fewer packets
- Path MTU discovery helps reduce overhead by sending more data per packet

```
RR# show tcp stat | begin Sent:
Sent: 615415 Total, 0 urgent packets
      0 control packets (including 0 retransmitted)
      602587 data packets (818797102 bytes)
      9609 data packets (7053551 bytes) retransmitted
      2603 ack only packets (1757 delayed)
      0 window probe packets, 355 window update packets
```

- BGP converged in 15 minutes!
- A respectable time for 250 peers and 100k routes

Summary/Tips

- Use ACLs when enabling debug commands
- Enable bgp log-neighbor-changes
- Use bgp deterministic-med
- If the entire table is having problem pick one prefix and troubleshoot it

References

- **TAC BGP pages—Very nice**

  http://www.cisco.com/cgi-bin/Support/PSP/psp_view.pl?p=Internetworking:BGP

- **BGP Case Studies**

  http://www.cisco.com/warp/public/459/bgp-toc.html

- **Internet Routing Architectures**

  http://www.ciscopress.com/book.cfm?series=1&book=155

- **Standards**

  RFC 1771, 1997, etc…

  http://www.rfc-editor.org/rfcsearch.html

  http://search.ietf.org/search/brokers/internet-drafts/query.html

CISCO SYSTEMS

EMPOWERING THE INTERNET GENERATION

BGP New Features

Assumptions

Cisco.com

- **BGP operational experience**
  - **Basic configuration**
  - **Show commands**
  - **Clear commands**
- **Understand the attributes**
- **Understand the decision algorithm**
- **Know what a route-map and peer-group are**

Agenda

- New Features
- Multipath
- Graceful Restart
- Protocol Issues
- Convergence and Scalability

New Features—Agenda

Cisco.com

- **Policy Configuration and Maintenance**

  policy-lists

  route-map continue

  peer-templates

  update-groups

- **Cost Community**
- **Improved Counters**
- **“bgp suppress-inactive”**

policy-list

- Match policy-list make route-maps easier to maintain/configure
- Macro for a route-map
- Release in 12.0(22)S—CSCdv41129
- Example:

```
!
ip policy-list common-match
 match as-path 10
 match ip address 100
!
route-map bar permit 10
 match ip policy-list common-match
 set community 100:200
```

Continue Statement

- continue statement for route-maps
- Provides the ability to jump to a specific step within the current route-map
- 12.0(24)S—CSCdx90201

```
! Old way
route-map foo-old permit 10
 match ip address 1
 set community 100:57
 set as-path prepend 100 100
!
route-map foo-old permit 20
 match ip address 2
 set community 100:58
 set as-path prepend 100 100
!
```

```
! New way
route-map foo-new permit 10
 match ip address 1
 set community 100:57
 continue 30
!
route-map foo-new permit 20
 match ip address 2
 set community 100:58
 continue 30
!
route-map foo-new permit 30
 set as-path prepend 100 100
```

Policy Configuration

Cisco.com

- **Peer-groups are used to group peers with common outgoing policy**

  No exceptions in the outgoing policy are allowed

- **The main benefits of peer-groups are:**

  UPDATE replication: only one UPDATE message is created per peer-group—it is then sent to each individual member

  Configuration grouping: all the members of a peer-group MUST have the same outgoing policy

- **Any deviation from the peer-group's outgoing policy causes the peer not to be able to be a part of the peer-group**

  Results in longer configuration files

BGP Peer Templates

- **Used to group common configurations**

  Uses peer-group-like syntax

- **Hierarchical policy configuration mechanism**

  A peer-template may be used to provide policy configurations to an individual neighbor, a peer-group or another peer-template

  The more specific user takes precedence if policy overlaps

    individual neighbor > peer-group > peer-template

BGP Peer Templates

- 12.0(24)S
- Two types of templates
- Session template

  Can inherit from one session-template

  Used to configure AFI (address-family-identifier) independent parameters

  remote-as, ebgp-multihop, passwords, etc.

- Peer/policy template

  Can inherit from multiple peer/policy templates

  Used to configure AFI dependant parameters

  Filters, next-hop-self, route-reflector-client, etc.

Session Template

Cisco.com

```
router bgp 100
 !
 template peer-session all-sessions
  version 4
  timers 10 30
 exit-peer-session
 !
 template peer-session iBGP-session
  remote-as 100
  password 7
    022F021B12091A61484B0A0B1C07064B180C2338642C26
    272B1D
  description iBGP peer
  update-source Loopback0
  inherit peer-session all-sessions
 exit-peer-session
!
 template peer-session eBGP-session
  description eBGP peer
  ebgp-multihop 2
  inherit peer-session all-sessions
 exit-peer-session
!
```

```
 !
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 inherit peer-session iBGP-session
 neighbor 1.1.1.2 inherit peer-session iBGP-session
 neighbor 1.1.1.3 inherit peer-session iBGP-session
 neighbor 10.1.1.1 remote-as 1442
 neighbor 10.1.1.1 inherit peer-session eBGP-session
 neighbor 10.1.1.2 remote-as 6445
 neighbor 10.1.1.2 inherit peer-session eBGP-session
 no auto-summary
!
```

- **1.1.1.1 → 1.1.1.3 are configured with commands from all-sessions and iBGP-session**
- **10.1.1.1 → 10.1.1.2 are configured with commands from all-sessions and eBGP-session**

Policy Template

Cisco.com

```
router bgp 100
 template peer-policy all-peers
  prefix-list deny-martians in
  prefix-list deny-martians out
 exit-peer-policy
 !
 template peer-policy external-policy
  remove-private-as
  maximum-prefix 1000
  inherit peer-policy all-peers 10
 exit-peer-policy
 !
 template peer-policy full-routes-customer
  route-map full-routes out
  inherit peer-policy external-policy 10
 exit-peer-policy
 !
 template peer-policy partial-routes-customer
  route-map partial-routes out
  inherit peer-policy external-policy 10
 exit-peer-policy
 !
 template peer-policy internal-policy
  send-community
  inherit peer-policy all-peers 10
 exit-peer-policy
 !
 template peer-policy RRC
  route-reflector-client
  inherit peer-policy internal-policy 10
 exit-peer-policy
```

```
neighbor 1.1.1.1 inherit peer-policy internal-policy
neighbor 1.1.1.2 inherit peer-policy RRC
neighbor 1.1.1.3 inherit peer-policy RRC
neighbor 10.1.1.1 inherit peer-policy full-routes-customer
neighbor 10.1.1.2 inherit peer-policy partial-routes-customer
```

Policy Template


```
!
template peer-policy foo
 filter-list 100 out
 prefix-list foo-filter out
 inherit peer-policy all-peers 10
exit-peer-policy
!
template peer-policy bar
 prefix-list bar-filter out
exit-peer-policy
!
template peer-policy seq_example
 inherit peer-policy bar 20
 inherit peer-policy foo 10
exit-peer-policy
!
neighbor 10.1.1.3 remote-as 200
neighbor 10.1.1.3 inherit peer-policy seq_example
```

```
Router#show ip bgp neighbors 10.1.1.3 policy
 Neighbor: 10.1.1.3, Address-Family: IPv4
   Unicast
 Inherited polices:
  prefix-list deny-martians in
  prefix-list bar-filter out
  filter-list 100 out
Router#
```

- **A policy template can inherit from multiple templates**
- **Seq # determines priority if overlapping policies**
  - **Higher seq # has priority**

BGP Update Groups

- 12.0(24)S
- The problem: peer-groups help BGP scale but customers do not always use peer-groups, especially with eBGP peers
- The solution: treat peers with a common outbound policy as if they are in a peer-group
- An “update group” is a group of peers with a common outbound policy which will be converged as if they are in a peer-group

BGP Update Groups

- What "neighbor" commands determine a common outbound policy?
  - Outbound filters (route-maps, as-path ACLs, etc.)
  - Internal vs. external peer
  - min-advertisement-interval
  - ORF (Outbound Route Filtering)
  - route-reflector-client
  - next-hop-self
  - etc…
- "neighbor x.x.x.x default-originate" is the only exception
- Inbound policy does not matter

BGP Update Groups

- **Example**

```
router bgp 100
 neighbor 10.1.1.1 remote 200
 neighbor 10.1.1.1 route-map full-routes out
 …
 neighbor 10.1.1.30 remote-as 3453
 neighbor 10.1.1.30 route-map full-routes out
 neighbor 10.2.1.1 remote-as 25332
 neighbor 10.2.1.1 route-map customer-routes out
 …
 neighbor 10.2.1.5 remote-as 6344
 neighbor 10.2.1.5 route-map customer-routes out
```

BGP Update Groups

- “full-routes” peers are in one update-group
- “customer-routes” peers are in another
- New command—show ip bgp replication
- Displays summary of each update-group
    - # of members
    - # of updates formatted (MsgFmt) and replicated (MsgRepl)

```
Router#show ip bgp replication
BGP Total Messages Formatted/Enqueued : 0/0

     Index      Type  Members           Leader   MsgFmt  MsgRepl  Csize  Qsize
         1  external       30         10.1.1.1        0        0      0      0
         2  external        5         10.2.1.1        0        0      0      0
```

BGP Update Groups

Cisco.com

- "show ip bgp update-group"
- Peers with "`route-map customer-routes out`" are in update-group #2

```
Router#show ip bgp update-group 10.2.1.1
BGP version 4 update-group 2, external, Address Family: IPv4 Unicast
  BGP Update version : 0, messages 0/0
  Route map for outgoing advertisements is customer-routes
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
  Minimum time between advertisement runs is 30 seconds
  Has 5 members (* indicates the members currently being sent updates):
   10.2.1.1   10.2.1.2    10.2.1.3     10.2.1.4
   10.2.1.5
```

BGP Custom Decision Algorithm

Cisco.com

- **12.0(24)S**
- **The BGP uses the path attributes and other criteria (BGP ID, for example) to select a best path**

  Not all the attributes/metrics are used (or even significant) during the selection

  The decision process doesn't provide flexibility to assign locally significant criteria, except at pre-determined points (LOCAL_PREF, for example)

  Other changes require complex policy configurations and/or IGP metric modifications (which affect all the paths)
- **A flexible, locally significant metric is needed to address the specific policies of an AS**

BGP Custom Decision Algorithm

Solution

- **Operation specified in an upcoming IETF draft: *“BGP Cost Community”* (draft-retana-bgp-custom-decision-00.txt)**
- **The *Cost Community* is a non-transitive extended community that can be inserted at any point(s) into the BGP selection process**

Allows for custom selection process rules!!

Cost Community

Format

```
+--------------------------------+
| Type TBD (2 octets)            |
+--------------------------------+
| Point of Insertion (1 octet)   |
+--------------------------------+
| Community-ID (1 octet)         |
+--------------------------------+
| Cost (4 octet)                 |
+--------------------------------+
```

Value of the Path Attribute <u>after</u> which this Community Should Be Considered during the Best Path Selection Process

Multiple Communities May Be Used

Locally Significant Cost; Lower Cost Is Preferred; Default Value Is 0x7fffffff

BGP Custom Decision Algorithm

Operation

- **The Cost and Point of Insertion are assigned by the local administrator**

  All *Cost Communities* should be advertised throughout the local AS

  The Cost is considered at the Point of Insertion specified

- **Paths that do not contain the *Cost Community* (for a particular Point of Insertion) are considered to have the highest possible value**

- **Should only be used if a consistent best path selection implementation is deployed in the local AS**

BGP Improved Counters

- 12.0(24)S
- Global
  - More accurate “bytes consumed” counters
  - # of multipath prefixes
- per-peer/per-address-family
  - Sent/Rcvd #s for all message types
  - Per inbound/outbound filter #s for number of prefixes denied
  - # of automatically denied prefixes
  - # of bestpaths/multipaths recevied
  - # of explicit/implicit withdraws sent/rcvd
  - # of routes advertised
- “show ip traffic” now includes BGP data

BGP Improved Counters

Cisco.com

- **show ip bgp summary**

Router#show ip bgp summ

BGP router identifier 1.1.1.5, local AS number 100

BGP table version is 40, main routing table version 40

9 network entries using 1000 bytes of memory

15 paths using 413 bytes of memory

6 multipath network entries and 12 multipath paths

[snip]

BGP using 1413 total bytes of memory

BGP Improved Counters

Cisco.com

- *—will only be displayed when non-zero
- show ip bgp neighbor—per peer counters

```
Message statistics:
    InQ depth is 0
    OutQ depth is 0
                                 Sent          Rcvd
    Opens:                          1             1
    Notifications:                  0             0
    Updates:                        1             2
    Keepalives:                     9             9
    Route Refresh:                  0             0
*   Unrecognized:                 n/a             0
    Total:                         11            12
```

BGP Improved Counters

Cisco.com

- **show ip bgp neighbor—per peer/per-afi**

```
For address family: IPv4 Unicast
                                 Sent       Rcvd
  Prefix activity:               ----       ----
    Prefixes Current:               7          2 (Consumes 72 bytes)
    Prefixes Total:                 7          2
    Implicit Withdraw:              0          0
    Explicit Withdraw:              0          0
    Used as bestpath:             n/a          2
    Used as multipath:            n/a          0
*   Saved (soft-reconfig):        n/a          0
*   History paths:                n/a          0
```

BGP Improved Counters


```
                                    Outbound  Inbound
  Local Policy Denied Prefixes:     --------  -------
*   route-map:                             0        0
*   filter-list:                           0        0
*   prefix-list                            0        0
*   Ext Community:                       n/a        0
*   AS_PATH too long:                    n/a        0
*   AS_PATH loop:                        n/a        0
*   AS_PATH confed info:                 n/a        0
*   AS_PATH contains AS 0:               n/a        0
*   NEXT_HOP Martian:                    n/a        0
*   NEXT_HOP non-local:                  n/a        0
*   NEXT_HOP is us:                      n/a        0
*   CLUSTER_LIST loop:                   n/a        0
*   ORIGINATOR loop:                     n/a        6
```

BGP Improved Counters

```
*   unsuppress-map:                     0        n/a
*   advertise-map:                      0        n/a
*   VPN Imported prefix:                0        n/a
*   Well-known Community:               0        n/a
*   SOO loop:                           0        n/a
*   Bestpath from this peer:            2        n/a
*   Suppressed due to dampening:        0        n/a
*   Bestpath from iBGP peer:            0        n/a
*   Incorrect RIB for CE:               0        n/a
*   BGP distribute-list:                0        n/a
    Total:                              2          6
  Number of NLRIs in the update sent: max 7, min 0
```

BGP Suppress Inactive—12.2T

- RFC 1771 says that a route should only be advertised if successfully installed in the RIB
- Successful installation—either the BGP route or a route with a matching next-hop is installed
- “bgp suppress-inactive” knob is available to enforce this rule
- Used for strict RFC compliance

BGP Suppress Inactive—12.2T

- New show command
- "show ip bgp rib-failure"
- Will display all prefixes that were not installed in the RIB and why
- If "bgp suppress-inactive" is enabled, will display if the NH matches

Agenda

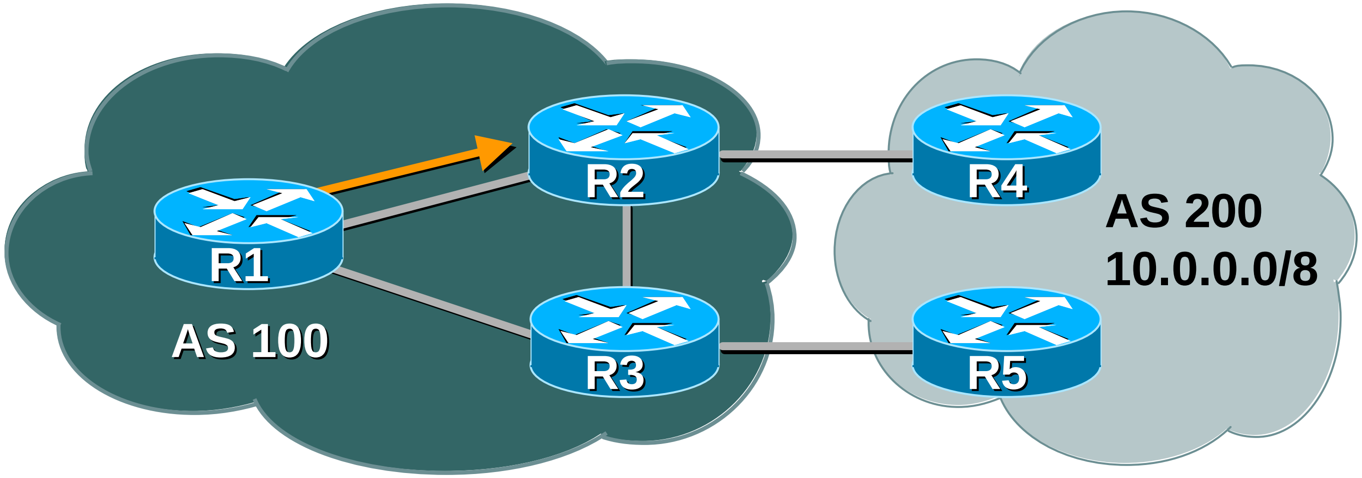
Cisco.com

- New Features
- Multipath
- Graceful Restart
- Protocol Issues
- Convergence and Scalability

Multipath Review

- Previously only supported for eBGP peers in the same Neighbor AS
- Multiple eBGP paths can be flagged as *multipath* as long as the paths are similar
- Similar means that all relevant BGP attributes are a tie and that there is no significant difference between the paths
  - If paths 1 and 2 both have a local-pref of 200, MED of 300, etc…but the router-IDs are different then paths 1 and 2 are eligible for multipath
- These paths are installed in the RIB/FIB to load-balance outbound traffic
- Multipath is the correct approach to a difficult problem but not terribly useful because it can only be used in one specific topology
  - iBGP multipath and link-BW will help correct this

iBGP Multipath



- R1 has two paths for 10.0.0.0/8
- Both paths are identical in terms of localpref, med, IGP cost to next-hop, etc.
- Router-ID, peer-address, etc are different but these are arbitrary in terms of selecting a best path
- R1 will select one path as best and send all traffic for 10.0.0.0/8 towards one of the exit points

iBGP Multipath

- **Flag multiple iBGP paths as 'multipath'**

  Each path must have a unique NEXT_HOP

- **All multipaths are inserted the RIB/FIB**
- **Number of multipaths can be controlled**

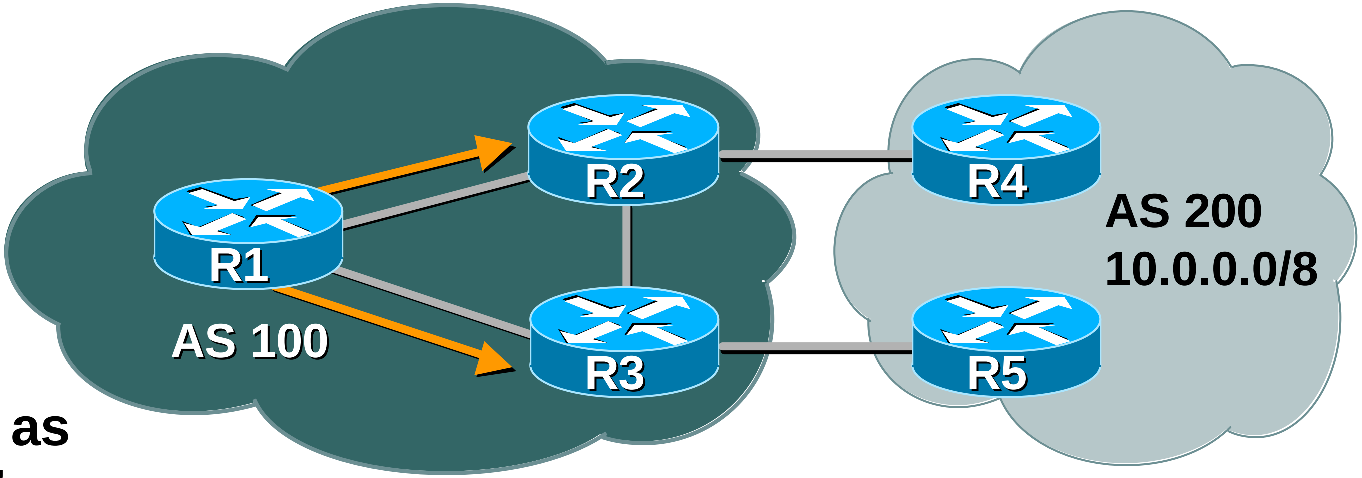
  maximum-paths ibgp <1-6>

- **The bestpath as determined by the decision algorithm will be advertised to our peers**
- **Each BGP next-hop is resolved and mapped to available IGP paths**
- **CSCdp72929—BGP: support iBGP multipath**

  12.0(22)S, 12.2(2)

iBGP Multipath

- R1 has two paths for 10.0.0.0/8
- Both paths are flagged as "multipath"



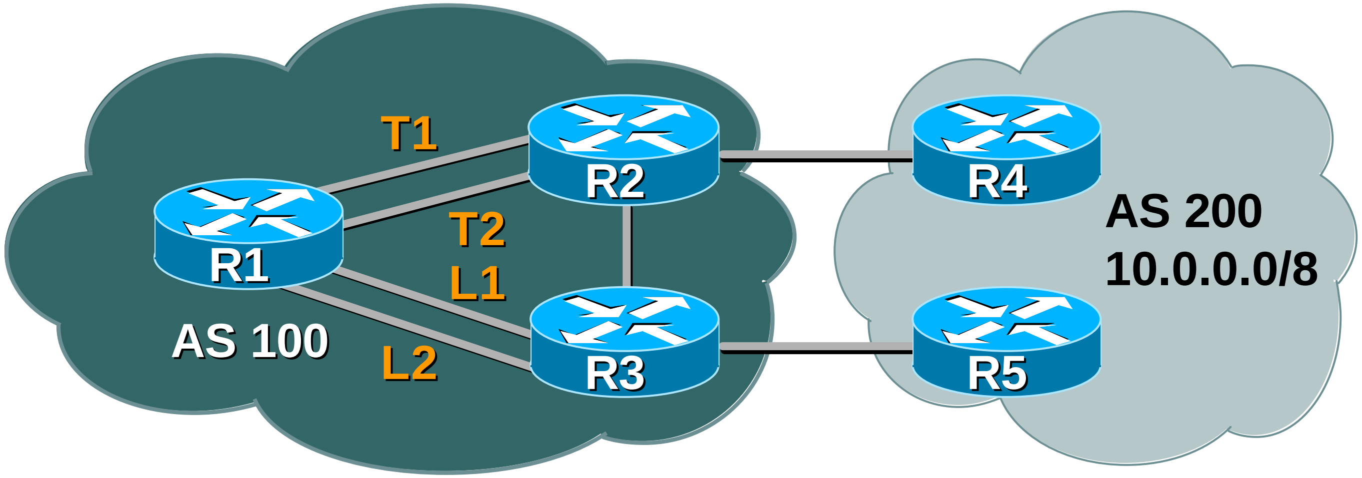

```
R1#sh ip bgp 10.0.0.0
  200
    20.20.20.3 from 20.20.20.3 (3.3.3.3)
    Origin IGP, metric 0, localpref 100, valid, internal, multipath
  200
    20.20.20.2 from 20.20.20.2 (2.2.2.2)
    Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
```

iBGP Multipath

- These two paths are installed in the RIB/FIB
- Traffic is load-balanced across the two paths/exit points

```
R1#sh ip route 10.0.0.0
Routing entry for 10.0.0.0/8
  * 20.20.20.3, from 20.20.20.3, 00:00:09 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    20.20.20.2, from 20.20.20.2, 00:00:09 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1

R1#show ip cef 10.0.0.0
10.0.0.0/8, version 237, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.3, 0 dependencies, recursive
    traffic share 1
    next hop 20.20.20.3, FastEthernet0/0 via 20.20.20.3/32
    valid adjacency
  via 20.20.20.2, 0 dependencies, recursive
    traffic share 1
    next hop 20.20.20.2, FastEthernet0/0 via 20.20.20.2/32
    valid adjacency
```

iBGP Multipath



- **What about iBGP multipath and IGP load-balancing together?**
- **R1 will pick one IGP path to R2 and one IGP path to R3**

  10.0.0.0 via T1, L1

Link BW—Ext Community

Cisco.com

- Latest ext-community draft defines a new extended community that can be used to indicate the BW of the link used to exit the AS
- Useful data to have if you want to load-balance traffic based on the BW of the outbound link

  Great for the customer that has a T3 and a T1 and wants to load-balance evenly across them

- “Link-BW” is ext-community type 0x0004
- CSCdr46701—BGP/VPN: support link-bandwidth-attribute—support unequal load balancing

  12.2

Link BW—Ext Community

- In conjunction with iBGP and eBGP multipath, link-BW can be used to influence the traffic share for each multipath
- End result is that you can do unequal cost load-balancing based on the BW of the exit point
- Configuration tasks:
  1. Configure bandwidth of DMZ links
  2. Tell BGP to include the link-BW attribute for all routes learned from an eBGP peer

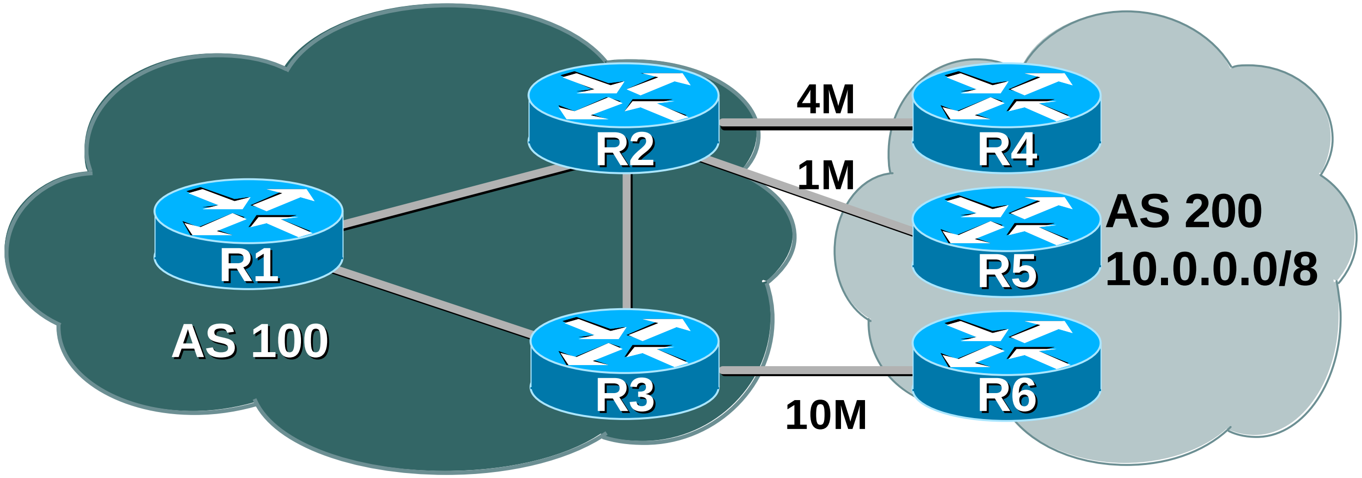
     neighbor x.x.x.x dmzlink-bw
  3. Tell BGP to send extended communities to iBGP peers

     neighbor x.x.x.x send-community [extended|both]
  4. Tell any router that implements eBGP/iBGP multipath to use the link-BW information to influence traffic share ratios

     bgp dmzlink-bw

Link BW—Ext Community

Cisco.com



- R2 wants to do unequal cost load-balancing over the 4M and 1M link
- R1 wants to do unequal cost load-balancing over the total amount of bandwidth for each exit point
  - 5M for R2
  - 10M for R3
- Configure R1 and R2 for multipath
- Configure R2 and R3 to send communities to R1
- Configure R2 and R3 to include Link-BW for routes learned from R4, R5, and R6

Link BW—Ext Community

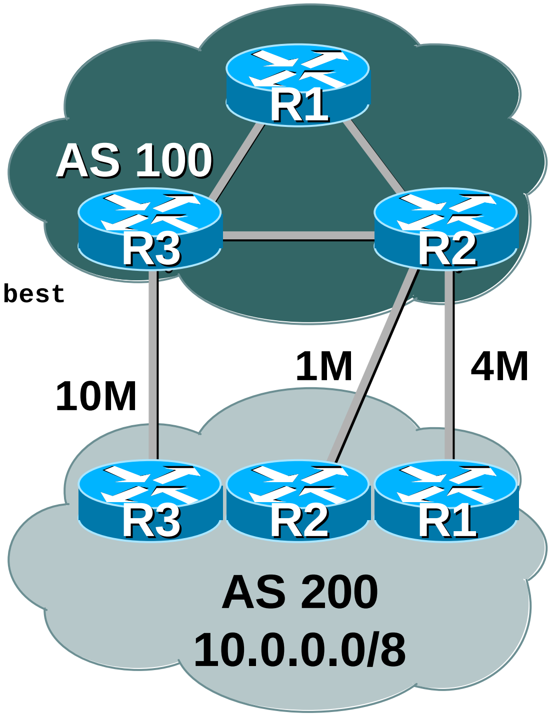
```
R1#
router bgp 100
 bgp dmzlink-bw
 maximum-paths ibgp 6
```

```
R2#
router bgp 100
 bgp dmzlink-bw
 maximum-paths 6
 neighbor 1.1.1.1 send-community extended
 neighbor 4.4.4.4 dmzlink-bw
 neighbor 5.5.5.5 dmzlink-bw
```

```
R3#
router bgp 100
 neighbor 1.1.1.1 send-community extended
 neighbor 6.6.6.6 dmzlink-bw
```

Link BW—Ext Community


```
R1#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 24
Paths: (2 available, best #1)
Multipath: iBGP
  Not advertised to any peer
  200
    20.20.20.2 from 20.20.20.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link bandwidth 10000000 kbit
  200
    20.20.20.3 from 20.20.20.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community: 0x0:0:0
      DMZ-Link bandwidth 5000000 kbit

R1#sh ip route 10.0.0.0
Routing entry for 10.0.0.0/8
  Known via "bgp 100", distance 200, metric 0
  Tag 200, type internal
  Last update from 20.20.20.3 00:11:17 ago
  Routing Descriptor Blocks:
  * 20.20.20.2, from 20.20.20.2, 00:11:17 ago
      Route metric is 0, traffic share count is 2
      AS Hops 1
    20.20.20.3, from 20.20.20.3, 00:11:17 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
```



Link BW—Ext Community

- **Link-BW is propagated to iBGP peers only and is stripped from paths before sending the paths to eBGP peers**
- **When doing eBGP-multipath, the bandwidth that is advertised to iBGP peers is the sum of DMZ-link bandwidth of all eBGP multipaths**

Agenda

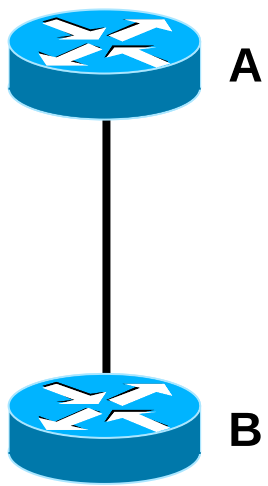
- New Features
- Multipath
- Graceful Restart
- Protocol Issues
- Convergence and Scalability

Non-Stop Forwarding

- What NSF solves
- An overview of how NSF works
- Where NSF is available

Non-Stop Forwarding

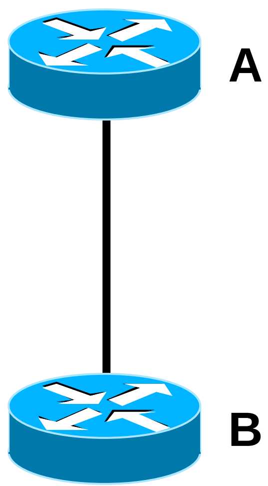
- Router A loses its control plane for some period of time
- It will take some time for Router B to recognize this failure, and react to it



Non-Stop Forwarding

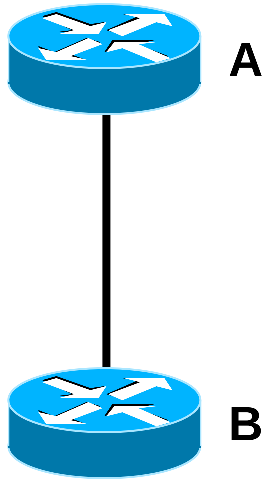
Cisco.com

- During the time that A has failed, and B has not detected the failure, B will continue forwarding traffic through A
- This traffic will be dropped



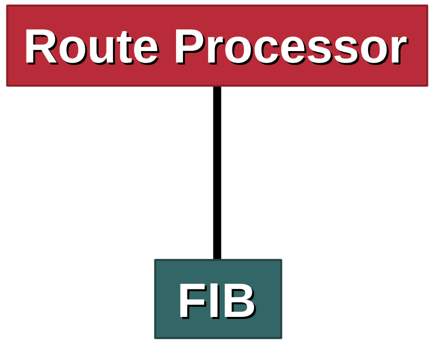
Non-Stop Forwarding

- **NSF reduces or eliminates the traffic dropped while A's control plane is down**
- **Some mechanism to recover forwarding information at the control plane must be used in conjunction with NSF, such as routing protocols graceful restart**



Non-Stop Forwarding

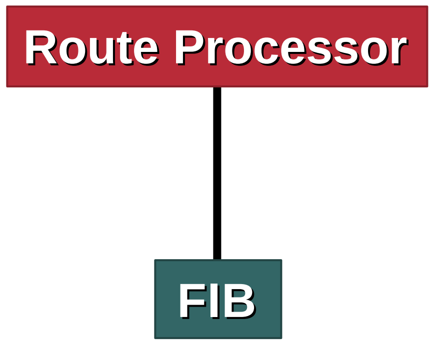
- **When the control plane fails, the FIB maintains its current state**
- **The switching components in the router continue forwarding information based on the last good known FIB information**



Non-Stop Forwarding

Cisco.com

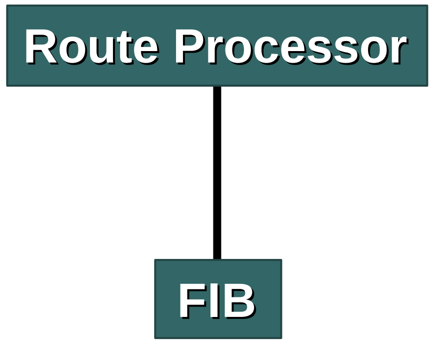
- **As the control plane restarts, various techniques are used to recover the information needed to rebuild the forwarding information**
- **While this information is rebuilt, the router continues switching packets based on the last known good forwarding information**



Non-Stop Forwarding

Cisco.com

- When the control plane has finished rebuilding the information required, it signals the FIB that convergence is complete
- The old information stored in the FIB, from before the restart, is now cleaned out, and forwarding continues based on the new information only



Non-Stop Forwarding

- May be used for planned and unplanned events
- Switchover must be completed before dead/hold timer expires
  - Peers will reset the adjacency and re-route the traffic after that time
- Transient routing loops or black holes may be introduced if the network topology changes before the FIB is updated

Non-Stop Forwarding

- Supported on the Cisco 10000
- Supported on the GSR
- Supported on the 6500
- Supported on the Cisco 7500, with the caveat that inserting a new standby RSP will cause some traffic loss, and switching from the primary to standby RSP will cause a microcode reload on the line cards

2 January 2003

NSF—Routing Protocol Requirements

Cisco.com

- Switchover must be completed before dead/hold timer expires

    Peers will reset the adjacency and re-route the traffic after that time

- FIB must remain unchanged during switchover

    Current routes marked as “dirty” during restart; “cleaned” once convergence is complete

- Adjacencies must not be reset when switchover is complete

    Protocol state is not maintained

- Peers of restarting router should also be NSF-aware

    Needed to take full advantage of NSF

NSF—Operation

- **Once the switchover is complete…**
  - **Routing protocol is restarted**
  - **Signal sent to neighbors indicating the process is restarting**
    - **Avoids adjacencies from being reset**
  - **Exchange of routing information (re-sync)**
    - **Route selection is done once re-sync is complete**
  - **FIB is updated**
    - **Any remaining "dirty" routes must be removed**
- **Transient routing loops or black holes may be introduced if the network topology changes before the FIB is updated**

BGP Graceful Restart

- **The goal of BGP Graceful Restart is to allow one peer to restart**
- **Peers of a restarting speaker should not route around the restarting speaker**
- **The tables of the restarting speaker should be rebuilt from existing routing information**

BGP Graceful Restart

- Operation specified in an IETF draft:
  "Graceful Restart Mechanism for BGP"
  (draft-ietf-idr-restart-XX.txt)

- End-of-RIB marker

  Indicates the completion of the initial routing update after the session is established

    UPDATE with empty withdrawn NLRI

    MP_UNREACH_NLRI used for other address families

BGP Graceful Restart

Cisco.com

- **Graceful Restart capability**

  Used by a BGP speaker to indicate its ability to preserve its forwarding state during BGP restart; it can also be used to convey its intention to generate the End-of-RIB marker after completion of the initial routing update

  If no <AFI, Sub-AFI> is specified, then it just signals the intent of generating the End-of-RIB market

  Capability code: 64

BGP Graceful Restart


```
+-----------------------------------------------+
| Restart Flags (4 bits)                        |
+-----------------------------------------------+
| Restart Time in seconds (12 bits)             |
+-----------------------------------------------+
| Address Family Identifier (16 bits)           |
+-----------------------------------------------+
| Subsequent Address Family Identifier (8 bits) |
+-----------------------------------------------+
| Flags for Address Family (8 bits)             |
+-----------------------------------------------+
| ...                                           |
+-----------------------------------------------+
| Address Family Identifier (16 bits)           |
+-----------------------------------------------+
| Subsequent Address Family Identifier (8 bits) |
+-----------------------------------------------+
| Flags for Address Family (8 bits)             |
+-----------------------------------------------+
```

BGP Graceful Restart

Graceful Restart Capability Fields

- **Restart flags: the most significant bit is defined as the restart state bit; when set (value 1) it indicates that the BGP speaker has restarted, and its peer should advertising routing information to it right away**
- **Restart time: estimated time (in seconds) it will take for the BGP session to be re-established after a restart**
- **Flags for address family: the most significant bit is defined as the forwarding state bit; when set (value 1), it indicates that the forwarding state has been preserved for the <AFI, Sub-AFI>**

BGP Graceful Restart

Special Operational/Deployment Considerations

- **Restarting router**

  Best path selection should be deferred until the End-of-RIB marker is received form all the peers, except peers that are restarting as well

- **Receiving router (restarting router's peer)**

  A new TCP connection opened by an existing peer should be interpreted as an indication of a restarting peer

- **All iBGP peers should be NSF-aware to reduce the risk of unwanted routing loops or black holes**

- **The IGPs must also be NSF-capable**

BGP Graceful Restart

- **BGP Graceful Restart is supported in 12.0(22)S**

Agenda

- New Features
- Multipath
- Graceful Restart
- Protocol Issues
- Convergence and Scalability

Protocol Issues—Agenda

- **Minimum Route Advertisement Interval**
- **NEXT_HOP Reachability**
- **Route Dampening**
- **Deterministic MED**
- **MED Oscillation**

minRouteAdvertisementInterval

"MinRouteAdvertisementInterval determines the minimum amount of time that must elapse between advertisement of routes to a particular destination from a single BGP speaker."

Draft-ietf-idr-bgp4-13

Section 9.2.3.1

minRouteAdvertisementInterval

Cisco.com

- *Studies show the effects of the minRouteAdvertisementInterval on BGP convergence
- In a nutshell

  Keeping the timer per peer instead of per prefix has some negative effects

  The default MinAdvInterval of 30 seconds may be too long

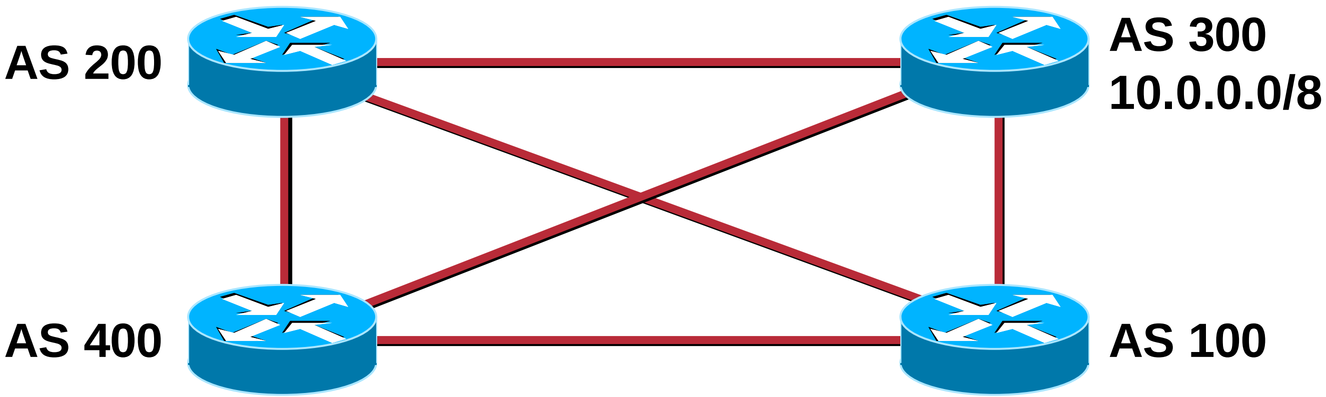
  TX loop detection should be implemented

    Using an outbound filter to prevent advertising routes to a peer that will deny them due to AS_PATH loop detection

*"An Experimental Study of Internet Routing Convergence"
—Labovitz, Ahuja, Bose, Jahanian

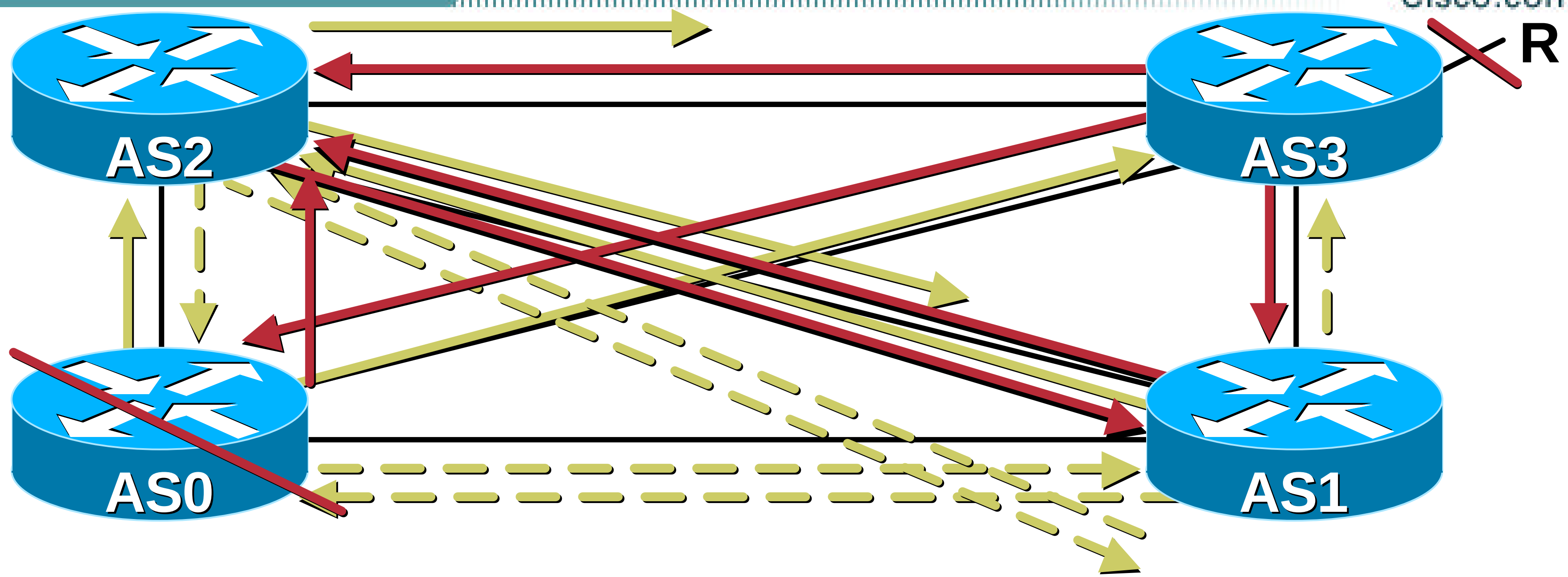
minRouteAdvertisementInterval

Cisco.com



- **Topology used to perform internal testing to study the effects when flapping the 10.0.0.0/8 prefix**
- **Convergence time, number of messages sent, number of denied messages, etc…are all monitored**

BGP Convergence Example
—Slide "Borrowed" from Labovitz Presentation



~~* B R via 3~~
~~* B R via 13~~
~~B R via 23~~

AS0

~~* B R via 3~~
~~* B R via 03~~
~~* B R via 203~~

AS1

~~* B R via 3~~
~~* B R via 013~~
~~B R via 103~~

AS2

Min Adv Interval—Variables

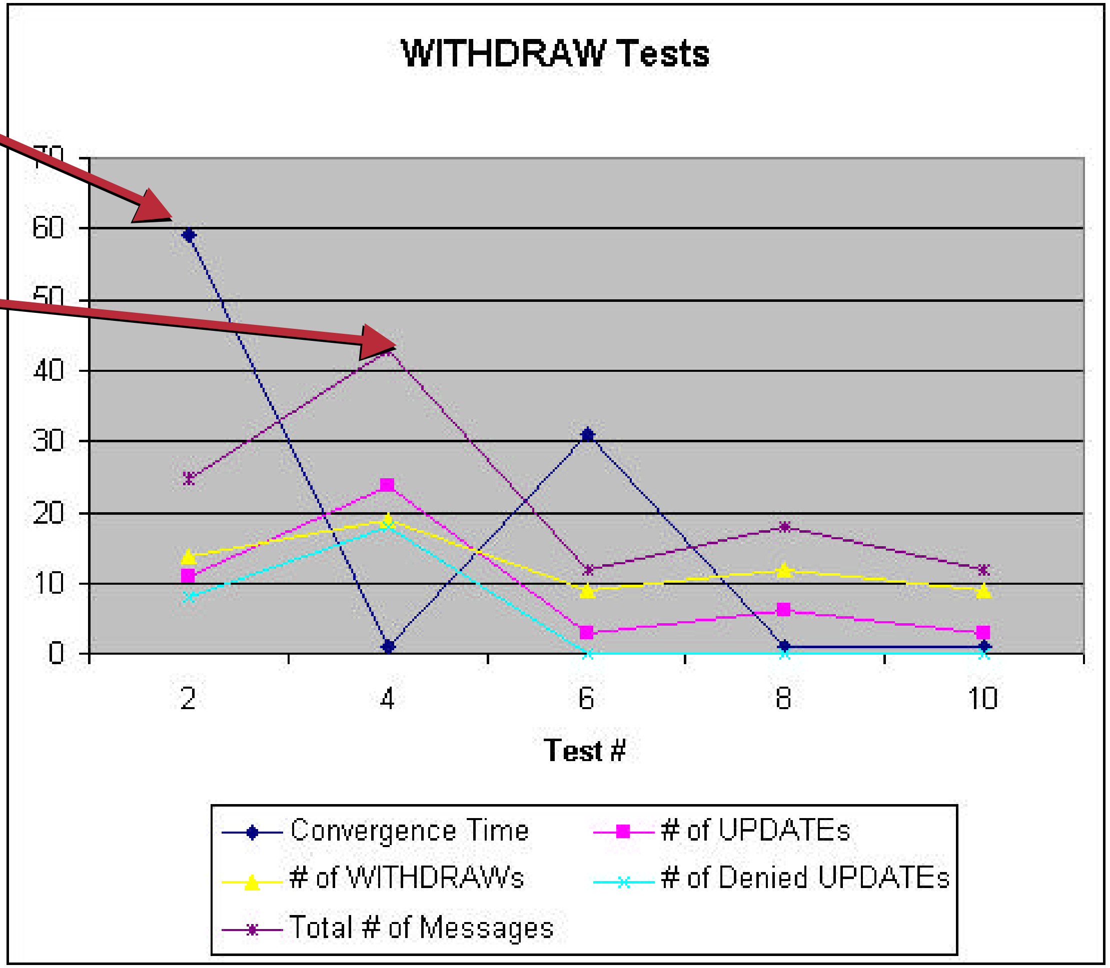
- **Min adv interval—0 seconds, 1 second, and 30 seconds**
- **Message type—advertisement (UPDATE) or WITHDRAW**
- **TX loop detection—either on or off; refers to using an outbound filter to prevent advertising routes to a peer that will be denied due to AS_PATH loop detection; example: if peer A is in AS 100 do not send A any routes that have AS 100 in the AS_PATH**

minRouteAdvertisementInterval—Test Matrix

	Message Type.	Timer (Sec)	TX Loop Detection	# Msgs Total	Denied UPDATES	Conv. (Sec)
Test 1	UPDATE	30		9		< 1
Test 2	WITHDRAW	30		25	8	59
Test 3	UPDATE	0		9		< 1
Test 4	WITHDRAW	0		43	18	< 1
Test 5	UPDATE	30	X	9		< 1
Test 6	WITHDRAW	30	X	12		31
Test 7	UPDATE	0	X	9		< 1
Test 8	WITHDRAW	0	X	18		< 1
Test 9	UPDATE	1	X	9		< 1
Test 10	WITHDRAW	1	X	12		< 1

minRouteAdvertisementInterval—Conclusions

- Default behavior takes almost 1 minute to converge
- Using a MinAdvInterval of 0 results results in a flurry of messages (43) for a single route-flap (see test 4)
- Using TX loop detection reduces the number of messages sent (see tests 6, 8, and 10)
- Best results are in test 10 which uses TX loop detection with Min Adv Interval of 1 second



minRouteAdvertisementInterval—Conclusions

- **Sending UPDATEs that will be denied unnecessarily triggers timer**
- **Setting the timer to 0 causes a flurry of messages**

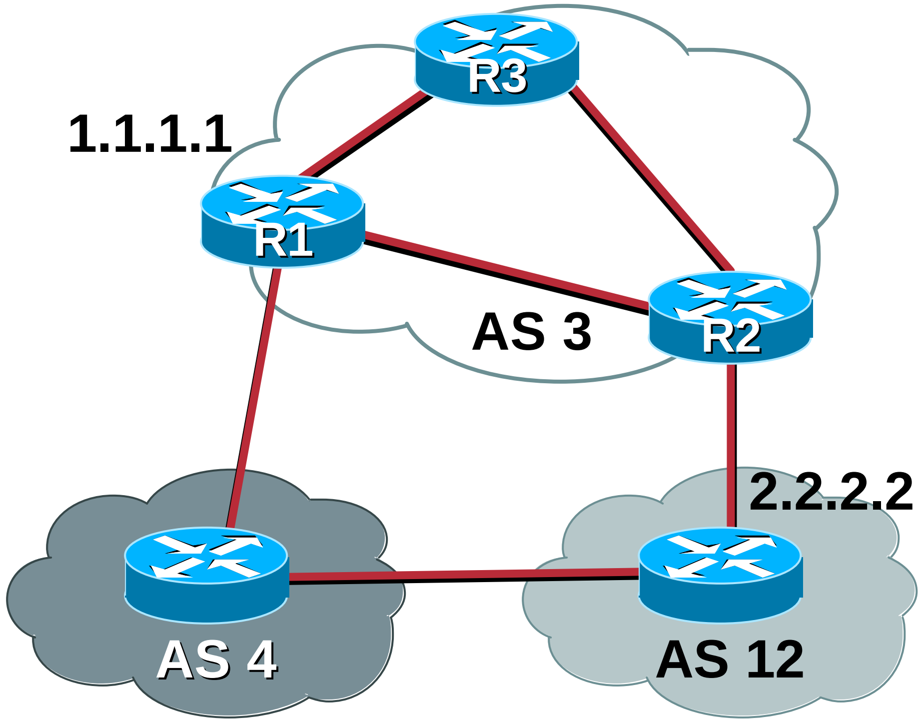
NEXT_HOP Reachability

- The NEXT_HOP must be reachable for the BGP path to be valid
  - Reachability should be provided by the IGP
- Other route characteristics also important for best path selection
  - IGP metric to NEXT_HOP
- Change in the reachability characteristics of the NEXT_HOP (availability, cost) may impair the ability to forward traffic and/or cause black holes or routing loops
  - BGP depends on the underlying IGP to provide fast and consistent notification of any change

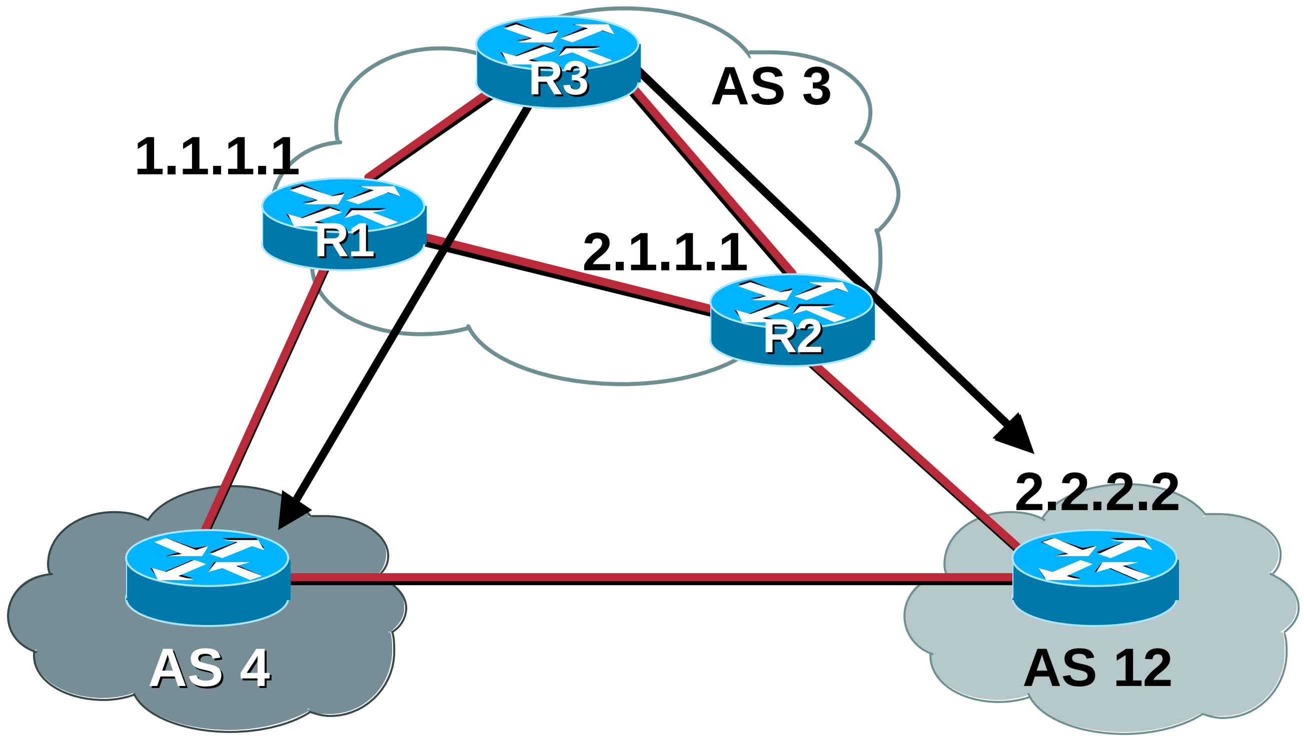
NEXT_HOP Reachability

Cisco.com

- R1 and R2 advertise routes to R3 with NEXT_HOPs of 1.1.1.1 and 2.2.2.2
- R3 must have a route to these two addresses
- Black holes and severe route flapping can occur if R3 does not have a proper route to both NEXT_HOPs



NEXT_HOP Route Oscillation—Symptoms



- R3 prefers routes via AS 4 one minute
- BGP scanner runs then R3 prefers routes via AS 12
- The entire table oscillates every 60 seconds

Route Oscillation—Symptom


```
R3#show ip bgp summary
BGP router identifier 3.3.3.3, local AS number 3
BGP table version is 502, main routing table version 502
267 network entries and 272 paths using 34623 bytes of memory

R3#sh ip route summary | begin bgp
bgp 3          4          6              520        1400
  External: 0 Internal: 10 Local: 0
internal       5                                     5800
Total          10         263            13936      43320
```

- **Watch for:**

  Table version number incrementing rapidly

  Number of networks/paths or external/internal routes changing

Route Oscillation—Troubleshooting

- Pick a route from the RIB that has changed within the last minute
- Monitor that route to see if it changes every minute

```
R3#show ip route 156.1.0.0
Routing entry for 156.1.0.0/16
  Known via "bgp 3", distance 200, metric 0
 Routing Descriptor Blocks:
  * 1.1.1.1, from 1.1.1.1, 00:00:53 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2, BGP network version 474

R3#show ip bgp 156.1.0.0
BGP routing table entry for 156.1.0.0/16, version 474
Paths: (2 available, best #1)
  Advertised to non peer-group peers:
    2.2.2.2
  4 12
    1.1.1.1 from 1.1.1.1
      Origin IGP, localpref 100, valid, internal, best
  12
    2.2.2.2 (inaccessible) from 2.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal
```

Route Oscillation—Troubleshooting

- Check again after bgp_scanner runs
- bgp_scanner runs every 60 seconds and validates reachability to all nexthops

```
R3#sh ip route 156.1.0.0
Routing entry for 156.1.0.0/16
  Known via "bgp 3", distance 200, metric 0
    Routing Descriptor Blocks:
  * 2.2.2.2, from 2.1.1.1, 00:00:27 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 478

R3#sh ip bgp 156.1.0.0
BGP routing table entry for 156.1.0.0/16, version 478
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    1.1.1.1
  4 12
    1.1.1.1 from 1.1.1.1
      Origin IGP, localpref 100, valid, internal
  12
    2.2.2.2 from 2.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation—Troubleshooting

- Lets take a closer look at the nexthop

```
R3#show ip route 2.2.2.2
Routing entry for 2.0.0.0/8
  Known via "bgp 3", distance 200, metric 0
 Routing Descriptor Blocks:
  * 2.2.2.2, from 2.1.1.1, 00:00:50 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 476

R3#show ip bgp 2.2.2.2
BGP routing table entry for 2.0.0.0/8, version 476
Paths: (2 available, best #2)
  Advertised to non peer-group peers:
    1.1.1.1
  4 12
    1.1.1.1 from 1.1.1.1
      Origin IGP, localpref 100, valid, internal
  12
    2.2.2.2 from 2.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Route Oscillation—Troubleshooting

Cisco.com

- BGP nexthop is known via BGP
- Illegal recursive lookup
- Scanner will notice and install the other path in the RIB

```
R3#sh debug
  BGP events debugging is on
  BGP updates debugging is on
  IP routing debugging is on
R3#
BGP: scanning routing tables
BGP: nettable_walker 2.0.0.0/8 calling revise_route
RT: del 2.0.0.0 via 2.2.2.2, bgp metric [200/0]
BGP: revise route installing 2.0.0.0/8 -> 1.1.1.1
RT: add 2.0.0.0/8 via 1.1.1.1, bgp metric [200/0]
RT: del 156.1.0.0 via 2.2.2.2, bgp metric [200/0]
BGP: revise route installing 156.1.0.0/16 -> 1.1.1.1
RT: add 156.1.0.0/16 via 1.1.1.1, bgp metric [200/0]
```

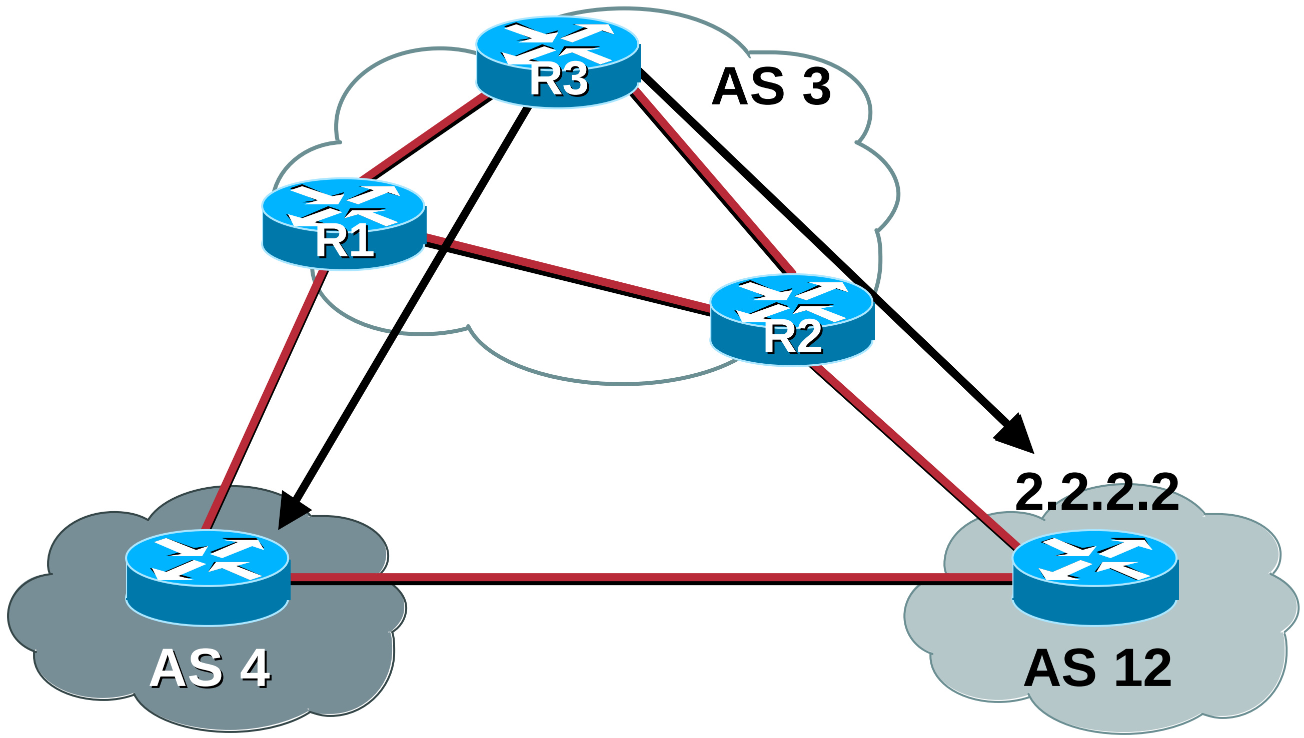
Route Oscillation—Troubleshooting

- Route to the nexthop is now valid
- Scanner will detect this and re-install the other path
- Routes will oscillate forever

```
R3#
BGP: scanning routing tables
BGP:  ip nettable_walker 2.0.0.0/8 calling revise_route
RT: del 2.0.0.0 via 1.1.1.1, bgp metric [200/0]
BGP: revise route installing 2.0.0.0/8 -> 2.2.2.2
RT: add 2.0.0.0/8 via 2.2.2.2, bgp metric [200/0]
BGP: nettable_walker 156.1.0.0/16 calling revise_route
RT: del 156.1.0.0 via 1.1.1.1, bgp metric [200/0]
BGP: revise route installing 156.1.0.0/16 -> 2.2.2.2
RT: add 156.1.0.0/16 via 2.2.2.2, bgp metric [200/0]
```

Route Oscillation—Step by Step

Cisco.com



- R3 naturally prefers routes from AS 12
- R3 does not have an IGP route to 2.2.2.2 which is the next-hop for routes learned via AS 12
- R3 learns 2.0.0.0/8 via AS 4 so 2.2.2.2 becomes reachable

Route Oscillation—Step by Step

- R3 then prefers the AS 12 route for 2.0.0.0/8 whose next-hop is 2.2.2.2
- This is an illegal recursive lookup
- BGP detects the problem when scanner runs and flags 2.2.2.2 as inaccessible
- Routes through AS 4 are now preferred
- The cycle continues forever…

NEXT_HOP Reachability

- **Three solutions**
- **Option 1—carry the R1 and R2 eBGP peering links in the IGP**
  - **Adds extra routes to the IGP**
  - **Carrying customer links adds instability to the IGP**
  - **Most unattractive option**
- **Option 2—do "redistribute connected" and "redistribute static" into BGP on R1 and R2**
  - **Adds a lot of extra routes to BGP; connected subnets of any router with an eBGP peer are now carried in the IGP and BGP**
  - **Carrying customer links adds instability to BGP**
  - **BGP will know how to get to its BGP NEXT_HOPs via BGP; illegal recursive lookups can easily led to severe route churn**
  - **Two recursive lookups have to be done to resolve the outbound interface; traffic forwarding is not effected but troubleshooting multiple recursive lookups becomes complex**
  - **AS carries more NEXT_HOPs than it has exit points; creates extra attribute combinations in the BGP table**

NEXT_HOP Reachability

- **Option 3—do *"neighbor x.x.x.x next-hop-self"* on the iBGP sessions from R1 and R2 to R3**
  - **Adds 0 routes to the IGP**
  - **Adds 0 routes to BGP**
  - **Promotes IGP/BGP stability by leaving customer links out of the picture**
  - **BGP will have an IGP route to BGP NEXT_HOPs; route churn due to illegal recursive lookups is no longer an issue**
  - **NEXT_HOPs accessed via a single recursive lookup which makes troubleshooting easier**
  - **Ideal option**
- **Note: "next-hop-self" to a route-reflector-client will not modify the NEXT_HOP of a reflected route; routes advertised from an eBGP peer to a RRC will be modified**

Dampening

Cisco.com

- **Defined in RFC 2439**
- **Route flap: the bouncing of a path or a change in its characteristics**

  A flap ripples through the entire Internet

  Consumes CPU cycles, causes instability

- **Solution: reduce scope of route flap propagation**

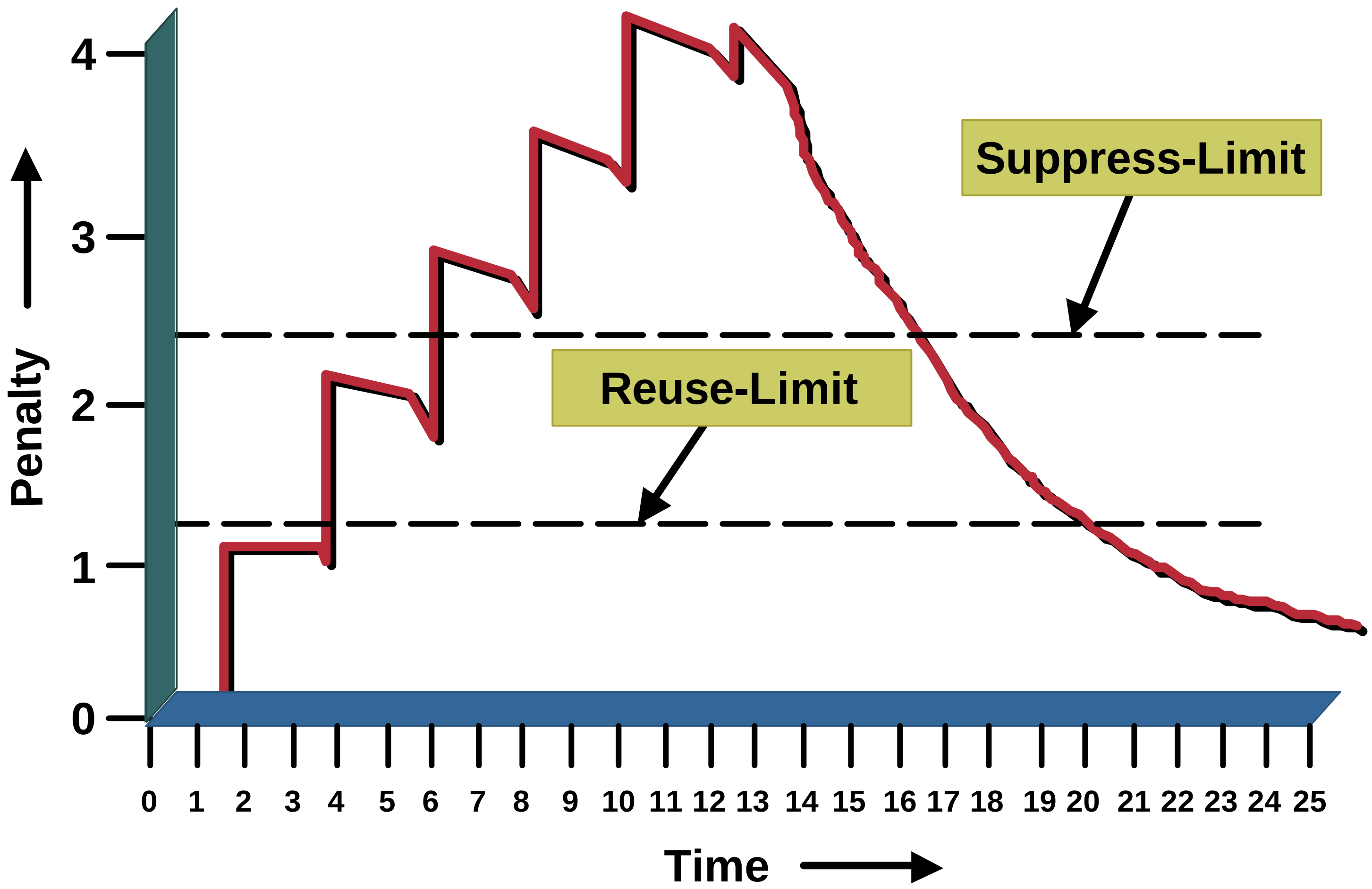
  History predicts future behavior

  Suppress oscillating routes

  Advertise stable suppressed routes

  Only eBGP routes are dampened

Dampening



4
3
2
1
0
Penalty
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

Dampening

- A route can only be suppressed when receiving an advertisement
  - Not when receiving a WITHDRAW
  - Attribute changes count as a flap (1/2)
- In order for a route to be suppressed the following must be true:
  - The penalty must be greater than the suppress-limit
  - An advertisement for the route must be received while the penalty is greater than the suppress-limit
  - A route will not automatically be suppressed if the suppress-limit is 1000 and the penalty reaches 1200; the route will only be suppressed if an advertisement is received while the penalty is decaying from 1200 down to 1000

Dampening—Deployment

- **Configurable parameters:**

  Half-life—the number of minutes it takes for the penalty to decay by 1/2

  Reuse-limit—if a route is suppressed the penalty must decay to this value to be unsuppressed

  Suppress-limit—the penalty must be greater than this threshold when an advertisement is received for a route to be suppressed

  Max-suppress-time—the maximum number of minutes a route may be suppressed

Dampening—Deployment

- **Calculated parameters:**

  Max-penalty—the maximum penalty a route may have that will allow the penalty to decay to reuse-limit within max-suppress-time

  max-penalty = reuse-limit * 2^(max-suppress-time/half-life)

  If half-life is 30, reuse-limit is 800, and max-suppress-time is 60 then the max-penalty would be 3200; if we allowed the penalty to reach 3201 it would be impossible for the penalty to decay to 800 within 60 minutes

Cisco IOS® Will Generate a Warning Message if the Max-Penalty Is above 20,000 or Less than the Suppress-Limit

Dampening—Example

Cisco.com

- **Small suppress window:**

  Half-life of 30 minutes, reuse-limit of 800, suppress-limit of 3000, and max-suppress-time of 60

  Max-penalty is 3200

- **Advertisement must be received while penalty is decaying from 3200 down to 3000 for the route to be suppressed**

  A 3 min 45 second (rough numbers) window exist for an advertisement to be received while decaying from 3200 to 3000

Dampening—Example II

- **No window:**

  Half-life of 30 minutes, reuse-limit of 750, suppress-limit of 3000, and max-suppress-time of 60

  Max-penalty = 750 * 2^(60/30) = 3000

  Here the max-penalty is equal to the suppress-limit

- **The penalty can only go as high as 3000**

  The decay begins immediately, so the penalty will be lower than 3000 by the time an advertisement is received

  A route could consistently flap several times a minute and never be suppressed

Dampening—Example III

- **Medium window**

  Half-life of 15 minutes, reuse-limit of 750, suppress-limit of 3000, and max-suppress-time of 45

  Max-penalty = 750 * 2^(45/15) = 6000

  Provides a 15 minute window

- **RIPE publishes recommendations**

  http://www.ripe.net/ripe/docs/ripe-210.html

Deterministic MED

- RFC says that MED is not always compared
- As a result, the ordering of the paths can effect the decision process
- By default, the prefixes are compared in order of arrival (most recent to oldest)

  Use bgp deterministic-med to order paths consistently

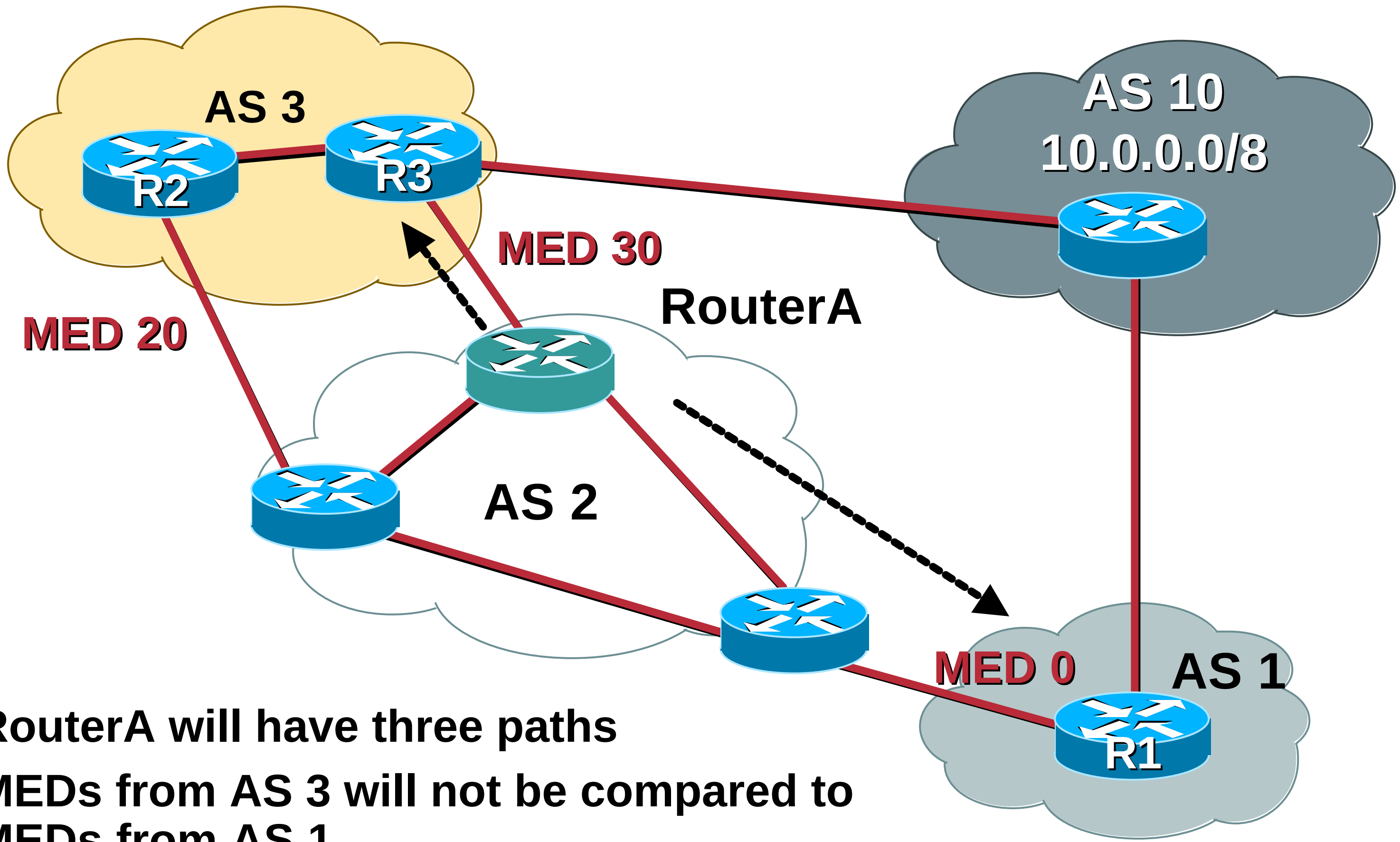
  The bestpath is recalculated as soon as the command is entered

  Enable in all the routers in the AS

Deterministic MED

- **Inconsistent route selection may cause problems**
  - **Routing loops**
  - **Convergence loops—i.e. the protocol continuously sends updates in an attempt to converge**
  - **Changes in traffic patterns**
- **Difficult to catch and troubleshoot**
- **It is best to avoid the problem in the first place**
  - **bgp deterministic-med**

Inconsistent Bestpath—Diagram



- RouterA will have three paths
- MEDs from AS 3 will not be compared to MEDs from AS 1
- RouterA will sometimes select the path from R1 as best and but may also select the path from R3 as best

Inconsistent Bestpath—Diagram

Cisco.com

```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #3, advertised over iBGP, eBGP)
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

- **Initial state**

  Path 1 beats path 2—lower MED

  Path 3 beats path 1—lower router-ID

Inconsistent Bestpath—Diagram


```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #3, advertised over iBGP, eBGP)
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external, best
```

- **1.1.1.1 bounced so the paths are re-ordered**

  Path 1 beats path 2—lower router-ID

  Path 3 beats path 1—external vs. internal

Deterministic MED—Operation

- The paths are ordered by Neighbor AS
- The bestpath for each Neighbor AS group is selected
- The overall bestpath results from comparing the winners from each group
- The bestpath will be consistent because paths will be placed in a deterministic order

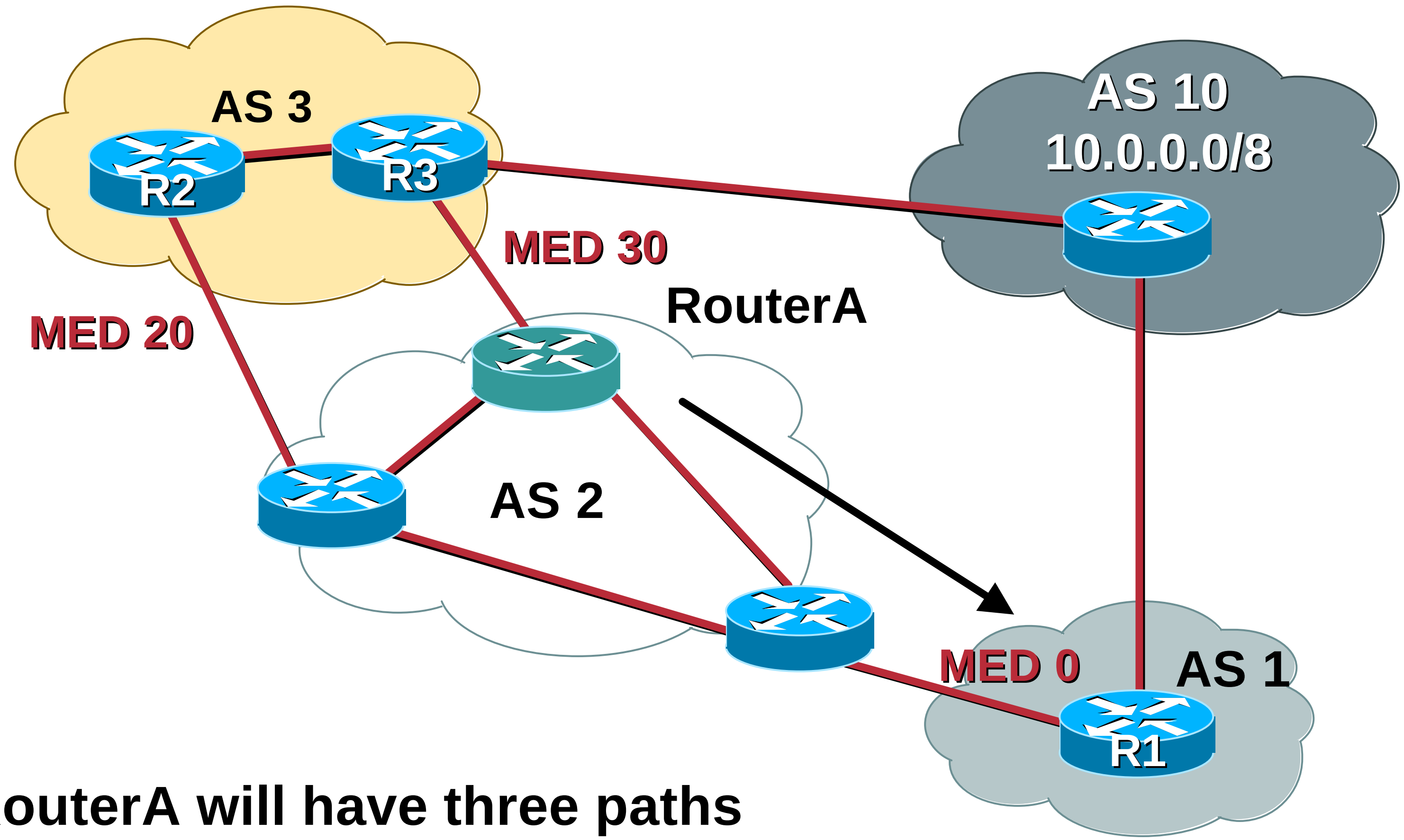
Deterministic MED—Result

Cisco.com

```
RouterA#sh ip bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 40
Paths: (3 available, best #1, advertised over iBGP, eBGP)
  1 10
    1.1.1.1 from 1.1.1.1
      Origin IGP, metric 0, localpref 100, valid, internal, best
  3 10
    2.2.2.2 from 2.2.2.2
      Origin IGP, metric 20, localpref 100, valid, internal
  3 10
    3.3.3.3 from 3.3.3.3
      Origin IGP, metric 30, valid, external
```

- **Path 1 is best for AS 1**
- **Path 2 beats path 3 for AS 3—lower MED**
- **Path 1 beats path 2—lower router-ID**

Solution—Diagram



- **RouterA will have three paths**
- **RouterA will consistently select the path from R1 as best!**

Deterministic MED—Summary

- Always use "bgp deterministic-med"
- Need to enable throughout entire network at roughly the same time
- If only enabled on a portion of the network routing loops and/or convergence problems may become more severe
- As a result, default behavior cannot be changed so the knob must be configured by the user

MED Churn

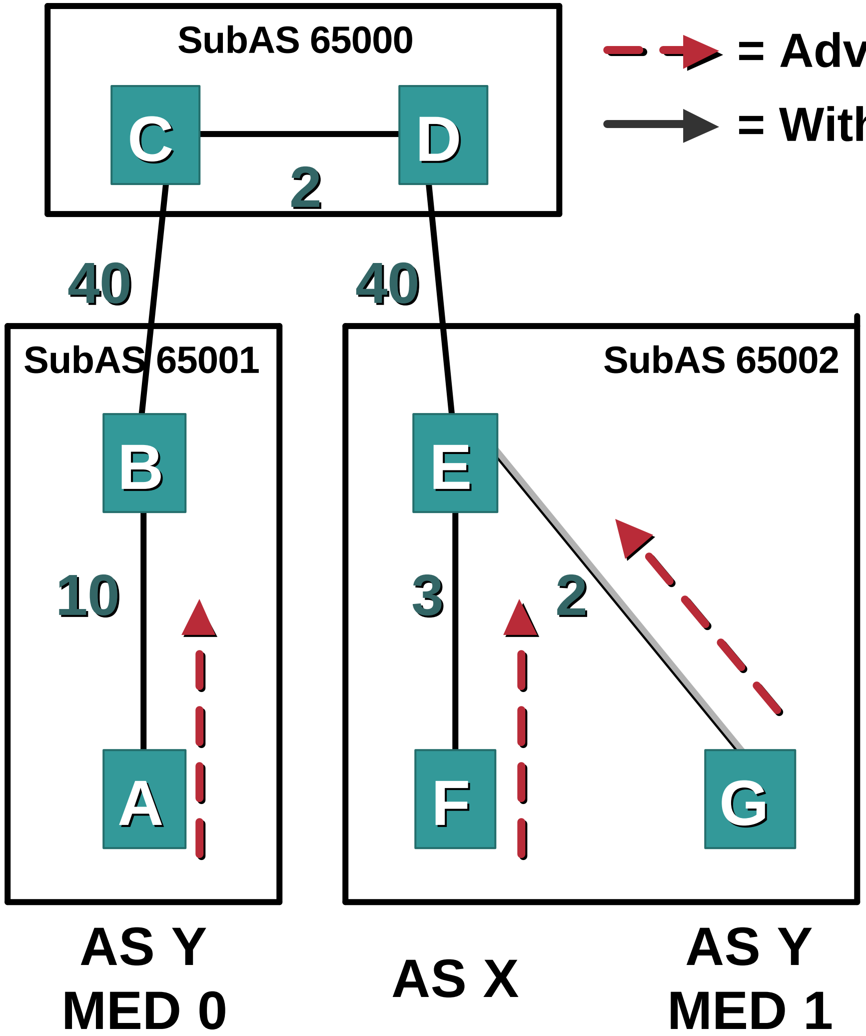
- RFC 3345
- Two types of MED Churn
- Type I—occurs in networks with a single tier of RRs or Sub-ASs

  Can be solved by following deployment guidelines

- Type II—occurs in networks with more than one tier of RRs or Sub-ASs

  Cannot solve without always comparing MEDs

Type II MED Churn—Example <todo – distinguish by other than color>



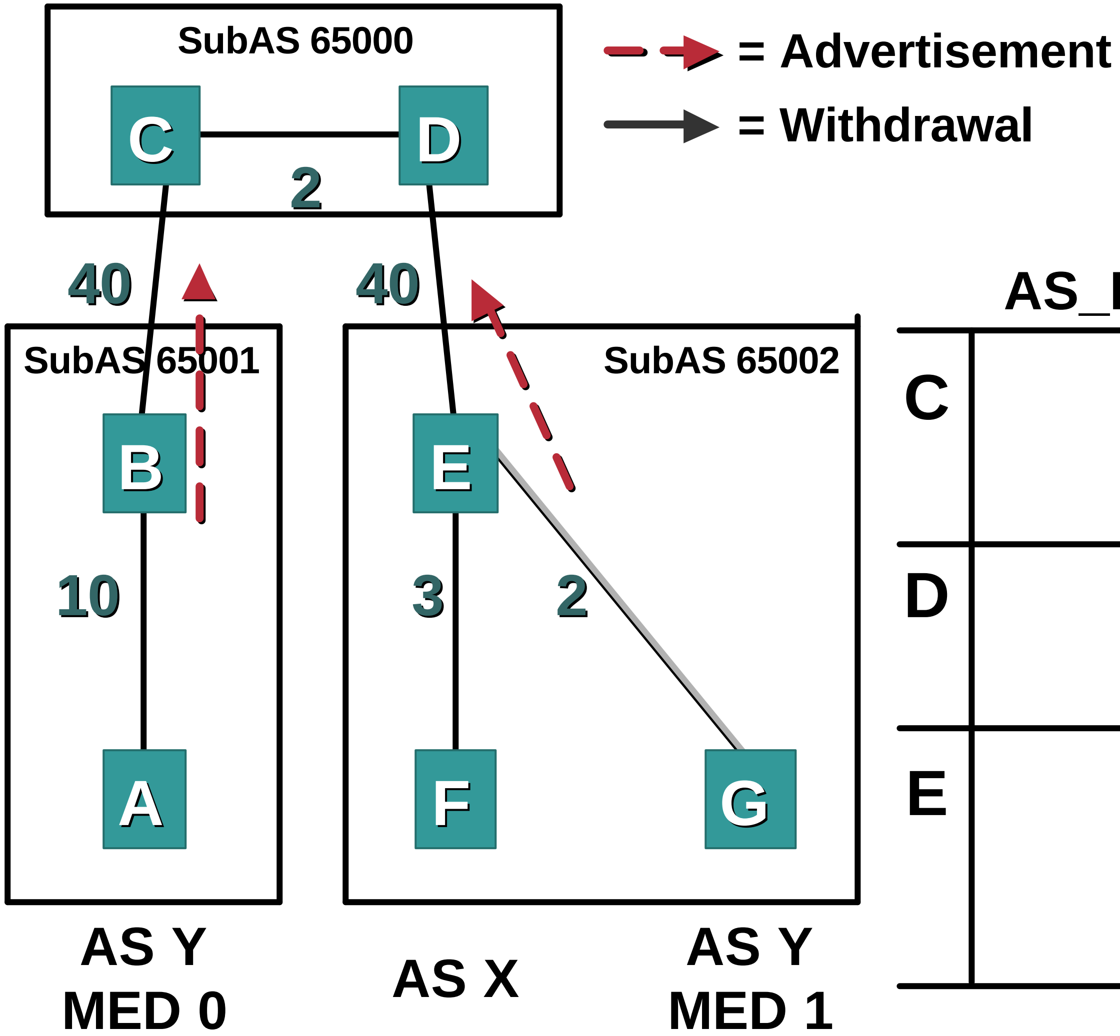
= Advertisement

= Withdrawal

Step 1
—E Selects Y1

	AS_PATH	MED	IGP
C			
D			
E	X		3
	* Y	1	2

Type II MED Churn—Example



Step 2

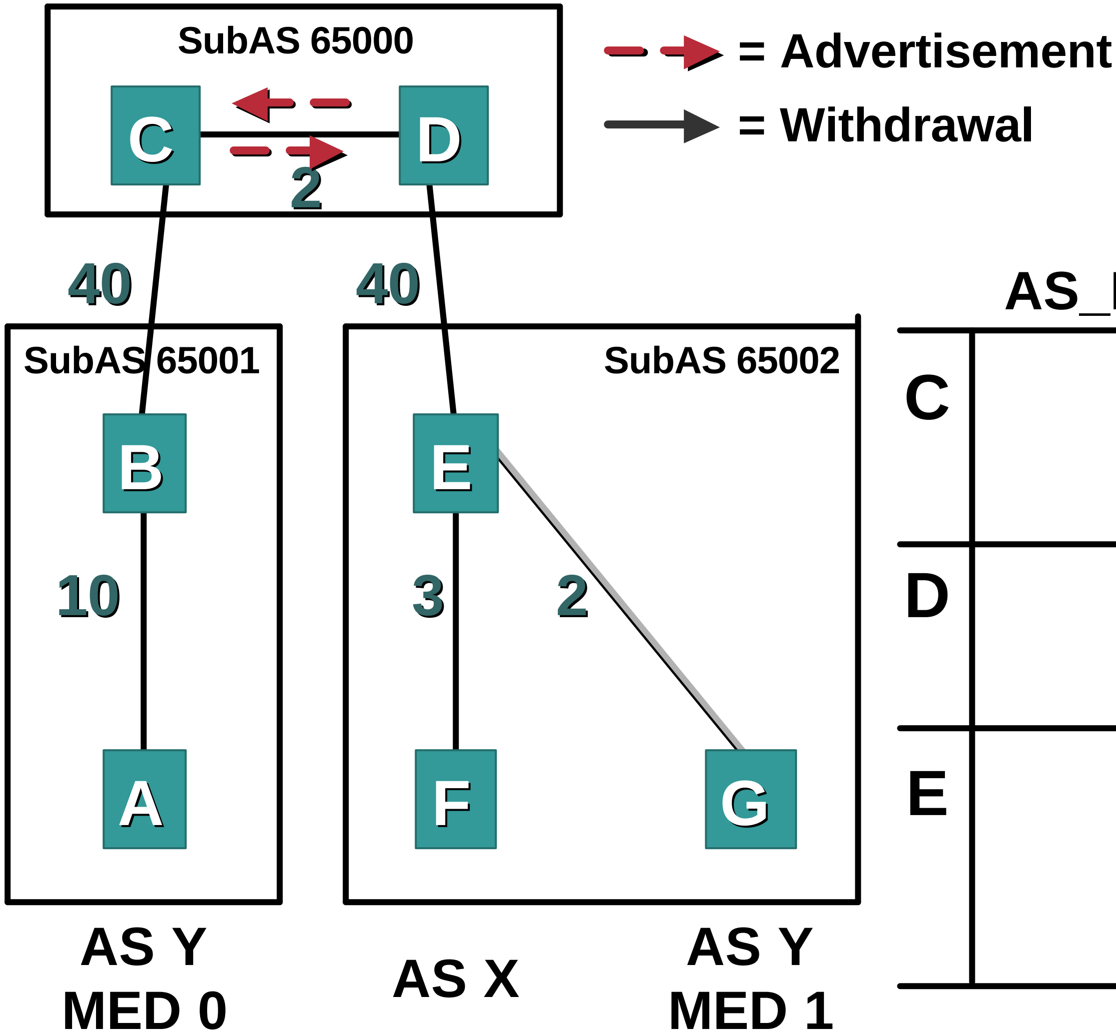
—C Selects Y0

—D Selects Y1

	AS_PATH	MED	IGP
C	* Y	0	50
D	* Y	1	42
E	X		3
	* Y	1	2

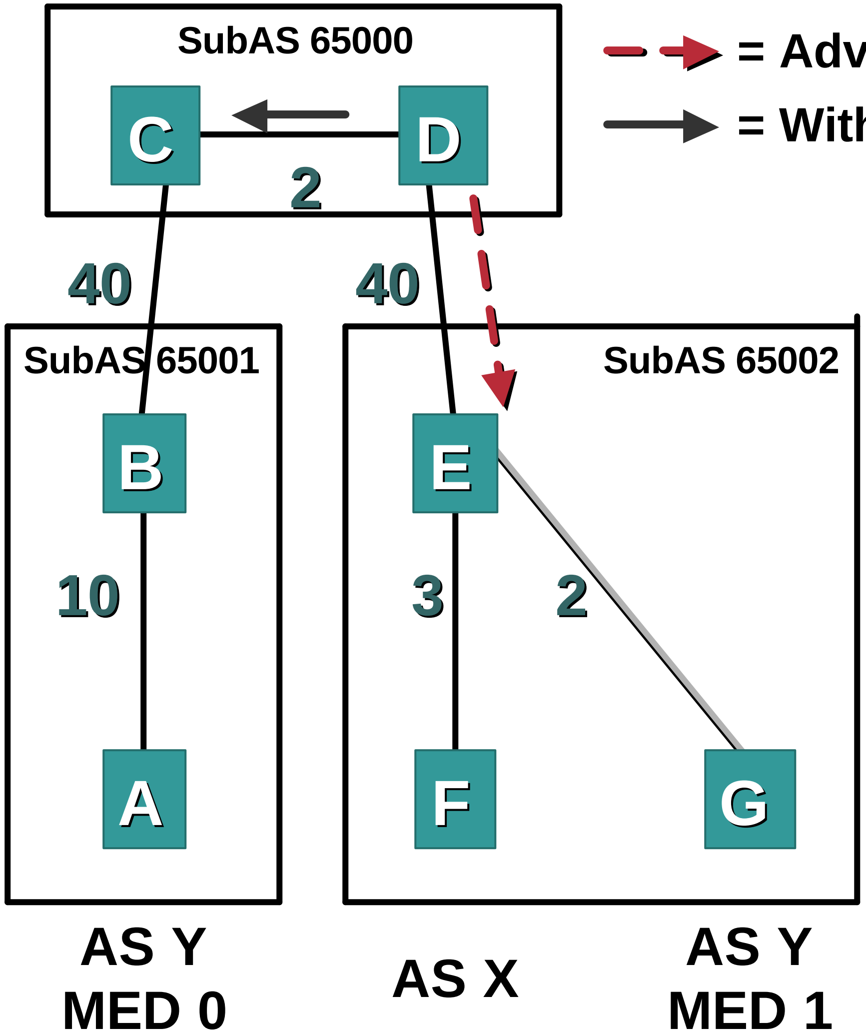
Type II MED Churn—Example

Step 3
—D Selects Y0



	AS_PATH	MED	IGP
C	* Y	0	50
	Y	1	44
D	* Y	0	52
	Y	1	42
E	X		3
	* Y	1	2

Type II MED Churn—Example



= Advertisement

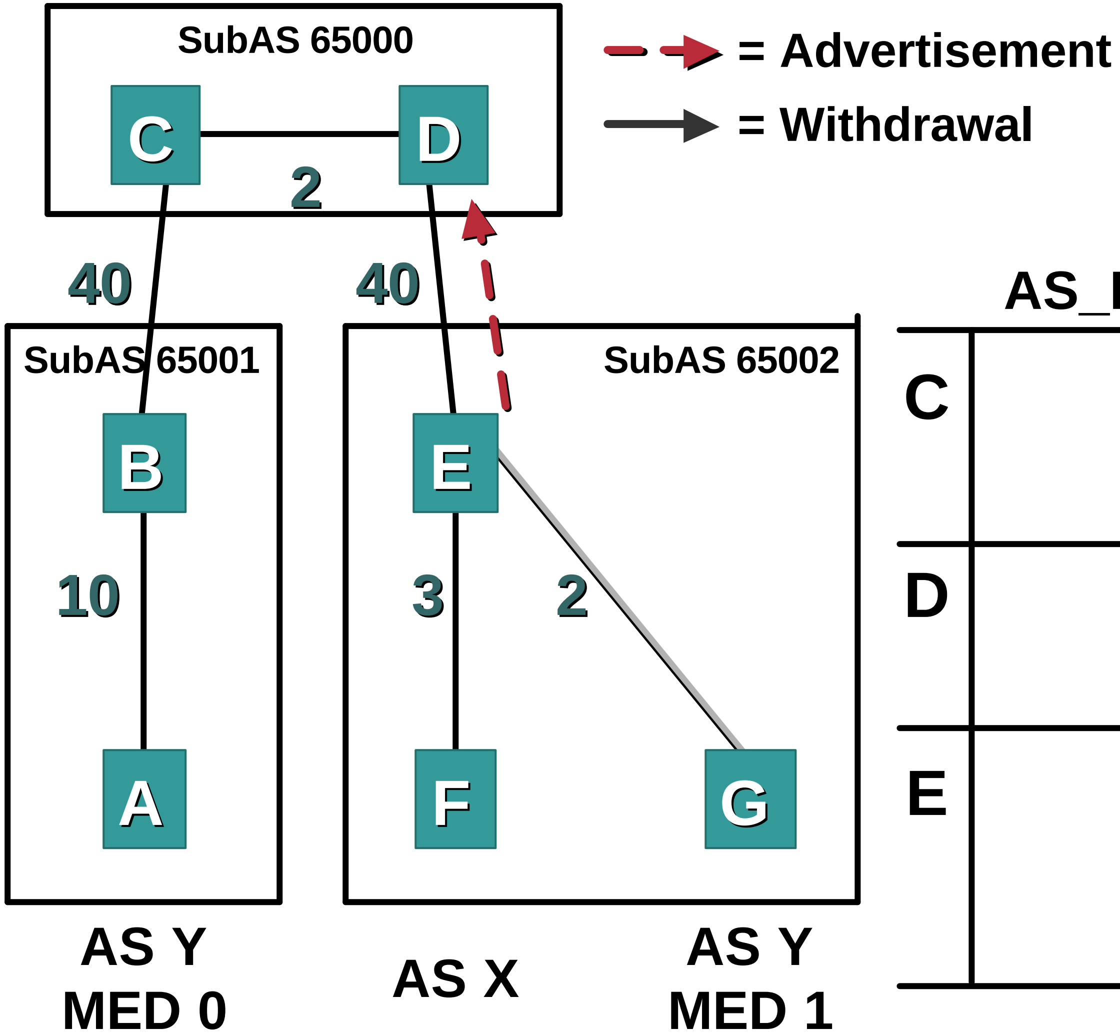
= Withdrawal

Step 4
—E Selects X

	AS_PATH	MED	IGP
C	* Y	0	50
D	* Y	0	52
	Y	1	42
E	* X		3
	Y	0	92
	Y	1	2

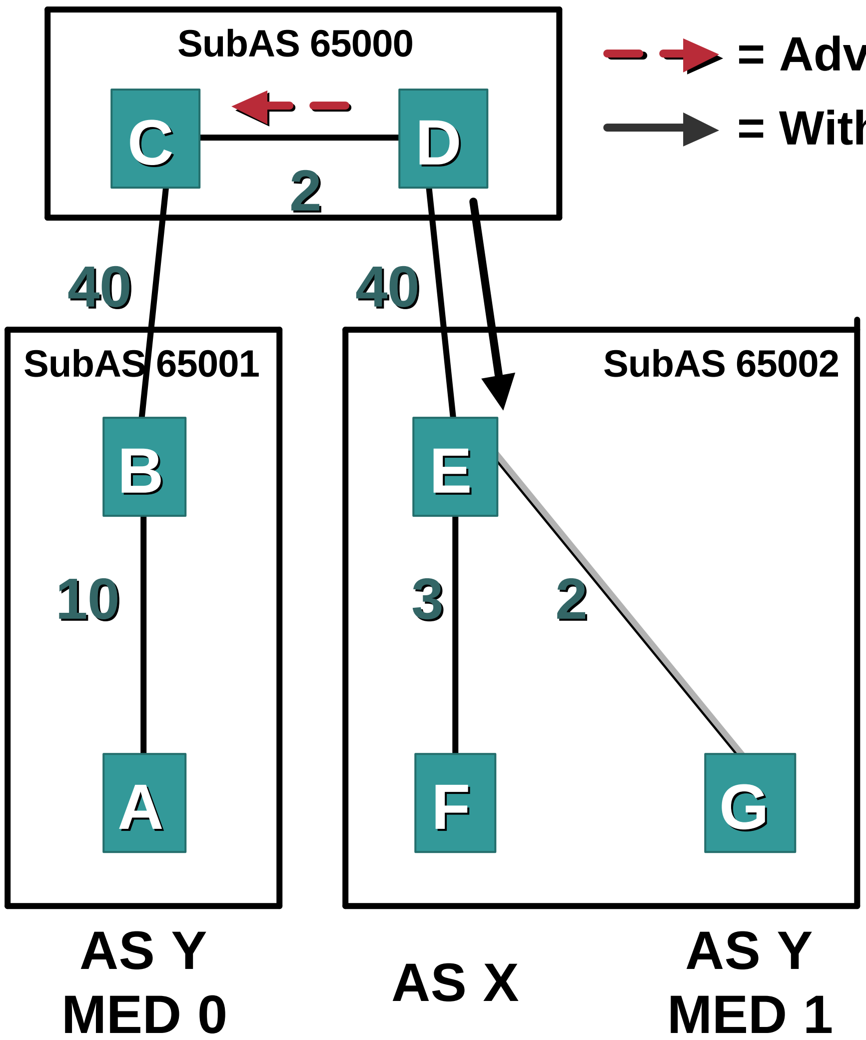
Type II MED Churn—Example

Step 5
—D Selects X



	AS_PATH	MED	IGP
C	* Y	0	50
D	Y	0	52
	* X		43
E	* X		3
	Y	0	92
	Y	1	2

Type II MED Churn—Example



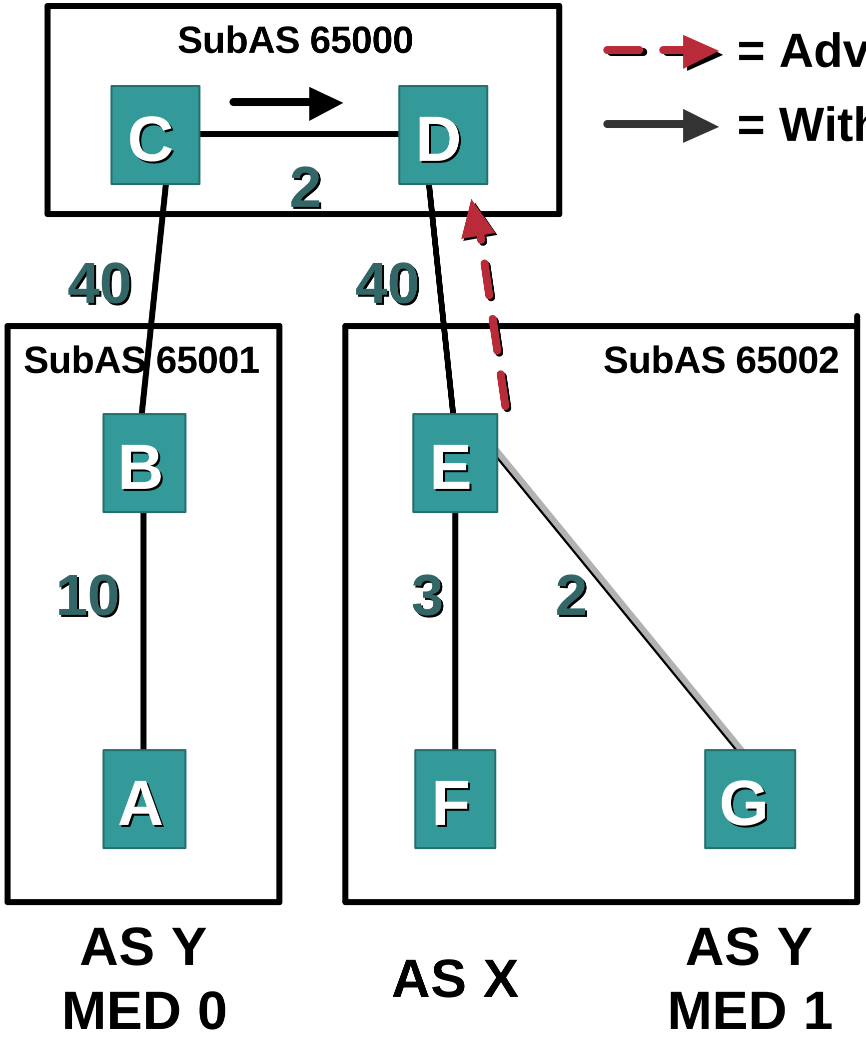
= Advertisement

= Withdrawal

Step 6
—C Selects X
—E Selects Y1

	AS_PATH	MED	IGP
C	Y	0	50
	* X		45
D	Y	0	52
	* X		43
E	X		3
	* Y	1	2

Type II MED Churn—Example



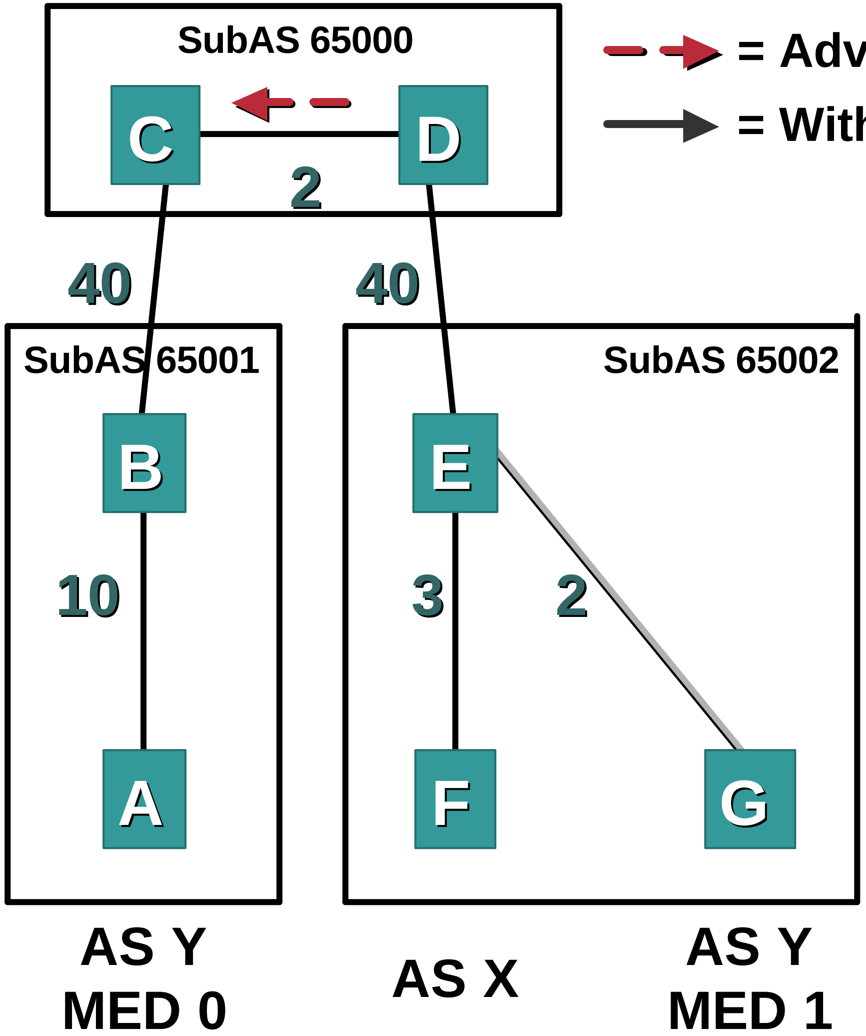
= Advertisement

= Withdrawal

Step 7
—D Selects Y1

	AS_PATH	MED	IGP
C	Y	0	50
	* X		45
D	* Y	1	42
E	X		3
	* Y	1	2

Type II MED Churn—Example



= Advertisement

= Withdrawal

Step 8

—C Selects Y0

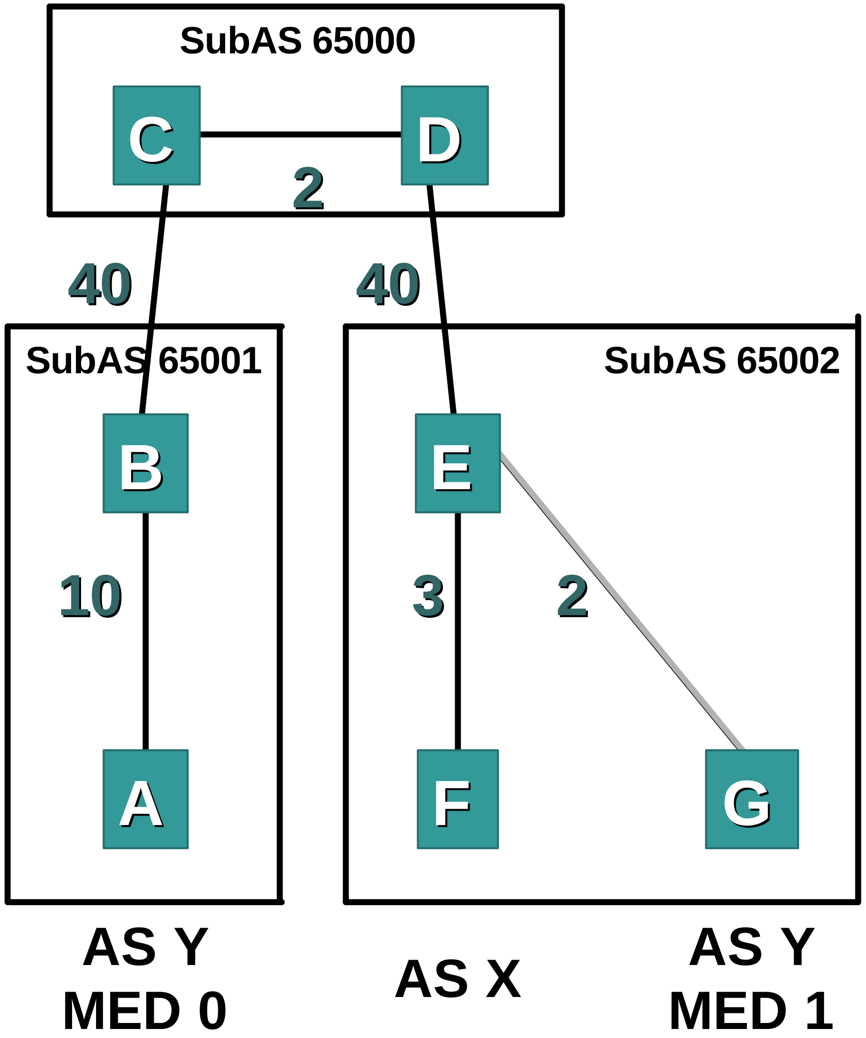
—This Is the Same as Step 2

—BGP Is in a Loop

	AS_PATH	MED	IGP
C	* Y	0	50
	Y	1	44
D	* Y	1	42
E	X		3
	* Y	1	2

Type II MED Churn—Example

Cisco.com



- In a nutshell, the churn happens because E does not always know about the Y0 path but the Y0 path has an effect on what E considers to be his best path
- Without Y0, E considers Y1 as best
- With Y0, E considers X as best
- From C and D's point of view

  Y0 < Y1 < X < Y0 ← this happens because MED is not compared every time

Possible Solutions

- **Solution #1—make sure E has the Y0 path**

  BGP peers will need to advertise multiple paths

  BGP will need a new attribute that will allow a speaker to advertise multiple paths for the same prefix

  A BGP speaker will then need to advertise a best path per "Neighbor AS" group IF that path came from an internal peer; this will force C and D to always advertise Y0 to D

- **Solution #2—eliminate "Y0 < Y1 < X < Y0" problem**

  Always comparing MEDs accomplishes this

Agenda

- New Features
- Multipath
- Graceful Restart
- Protocol Issues
- Convergence and Scalability

Convergence and Scalability

- **Advertising a full Internet table of routes to many peers is the main challenge**

  Router bootup

  clear ip bgp *

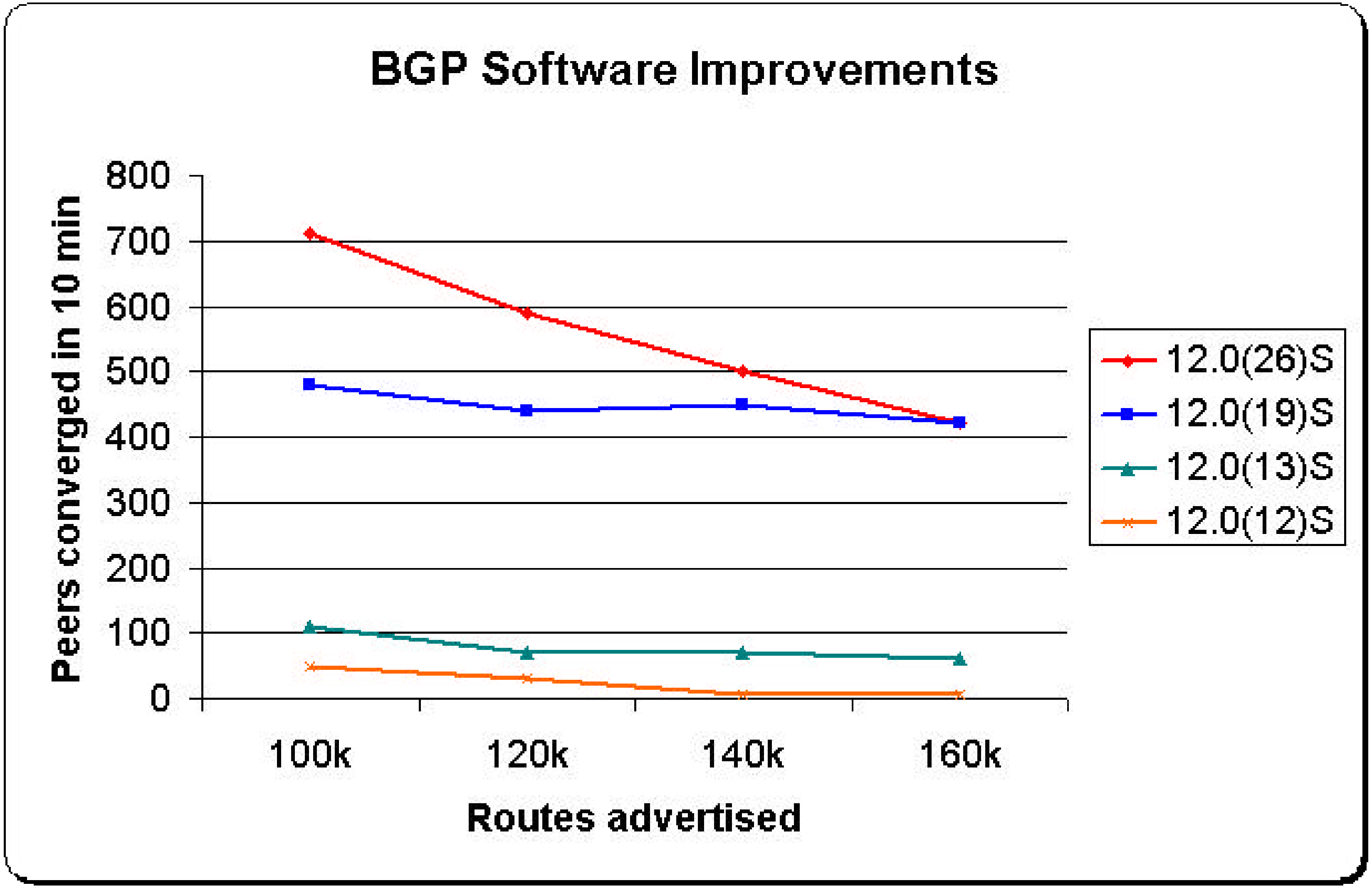
- **Thousands of peers can be supported if we only have to send them hundreds of routes**
- **Hundreds of peers can be supported if we have to send thousands of routes**

Convergence and Scalability

- **Two key ways to improve scalability**
- **Upgrade ☺**
  - **Improved update packing**
  - **Update-groups**
- **Configuration**
  - **Peer-groups**
  - **TCP**
  - **Queues**

Software Improvements

Cisco.com



peer-groups/update-groups

Cisco.com

- **Update-groups were introduced in 12.0(24)S**

  Treats peers with common outbound policy as if they are in a peer-group

  UPDATE generate for those peers is as if they are configured in a peer-group

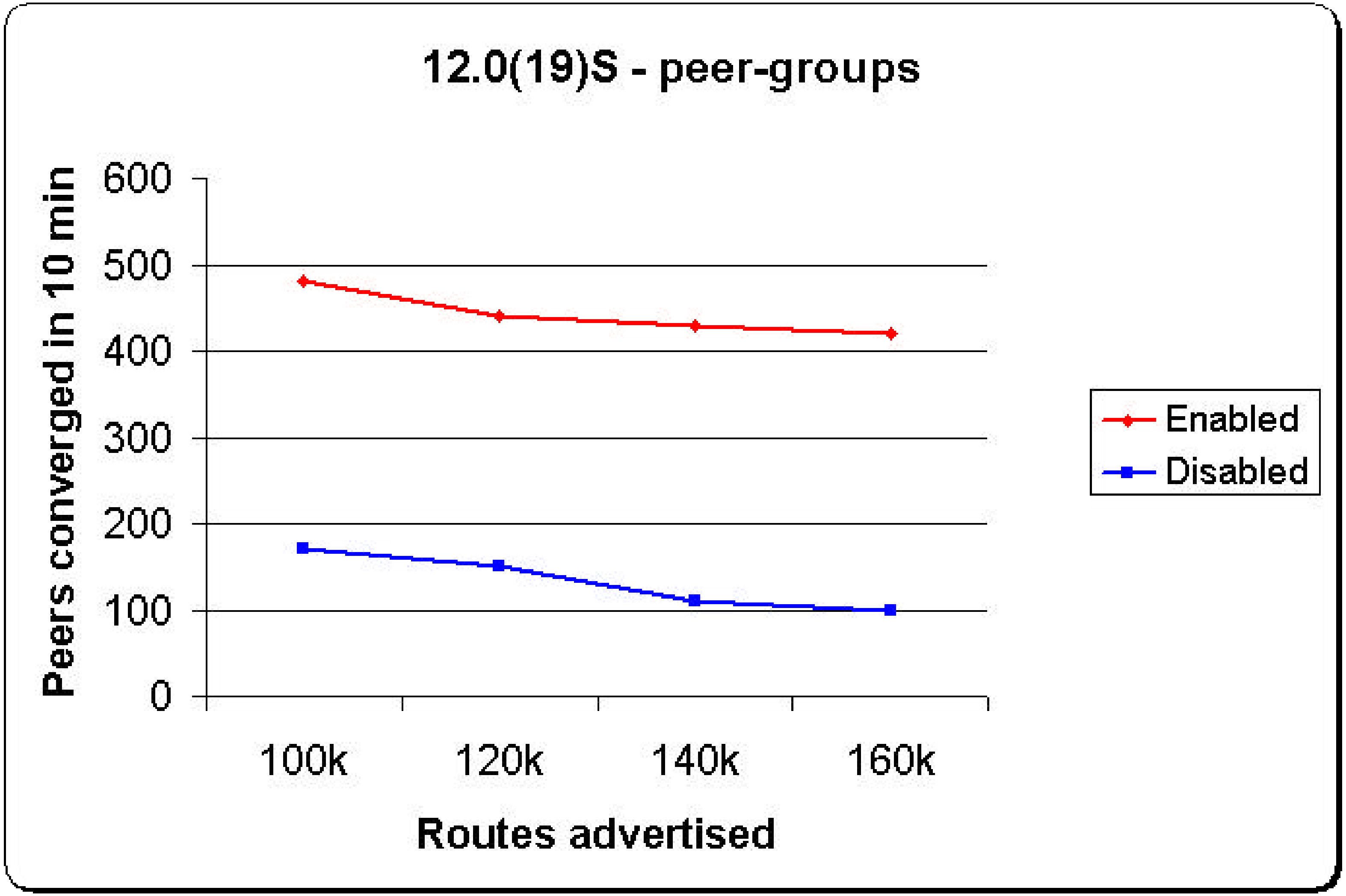
- **UPDATE generation without peer-groups/update-groups**

  The BGP table is walked for every peer, prefixes are filtered through outbound policies, UPDATEs are generated and sent to this one peer

- **UPDATE generation with peer-groups/update-groups**

  A leader is elected for each peer-group/update-group; the BGP table is walked for the leader only, prefixes are filtered through outbound policies, UPDATEs are generated and sent to the leader and replicated for peer-group/update-group members

peer-groups/update-groups



TCP Path MTU Discovery

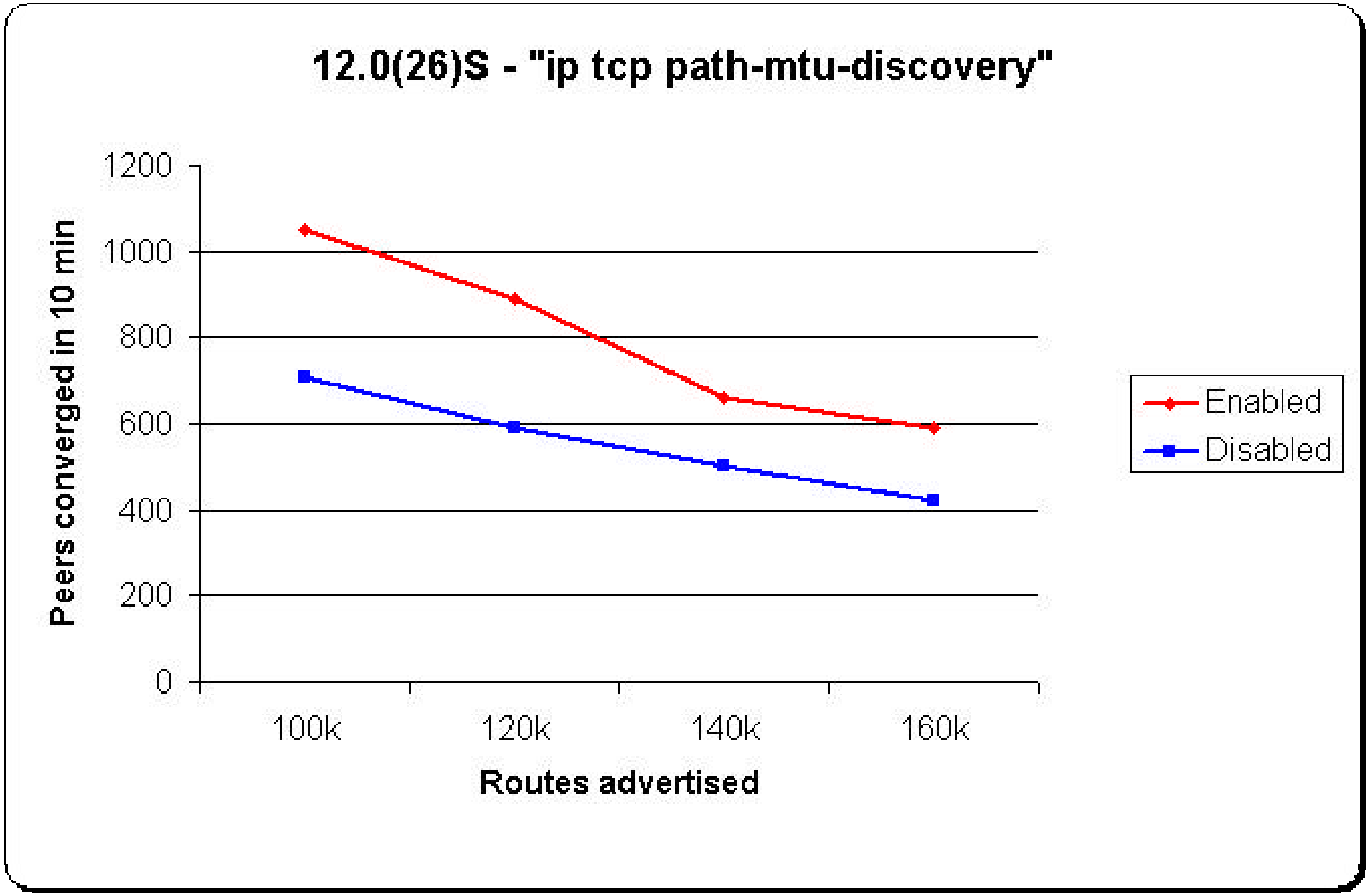
- Default MSS (Max Segment Size) for a TCP session is 536 bytes
- Inefficient for today's POS/Ethernet networks

  Ethernet MTU—1500

  POS MTU—4470
- "ip tcp path-mtu-discovery" determines the lowest MTU of all links between the end points of a TCP session
- MSS = lowest MTU—IP overhead
- Reduces TCP overhead

TCP Path MTU Discovery

Cisco.com



Input Queues

- **The problem**

  If a BGP speaker is pushing a full Internet table to a large number of peers, convergence is degraded due to enormous numbers of drops (100k+) on the interface input queue; ISP Foo gets ~½ million drops in 15 minutes on their typical route reflector

- **Complicated solution**

  Make the input queues big enough to hold all of the TCP Acks that would be generated if all of your peers were to Ack their entire window size of data at the exact same time

  The result is that BGP will converge much faster because we are no longer dropping tons of packet on the interface input queues; we also have the benefit of keeping our input queues at reasonable depths

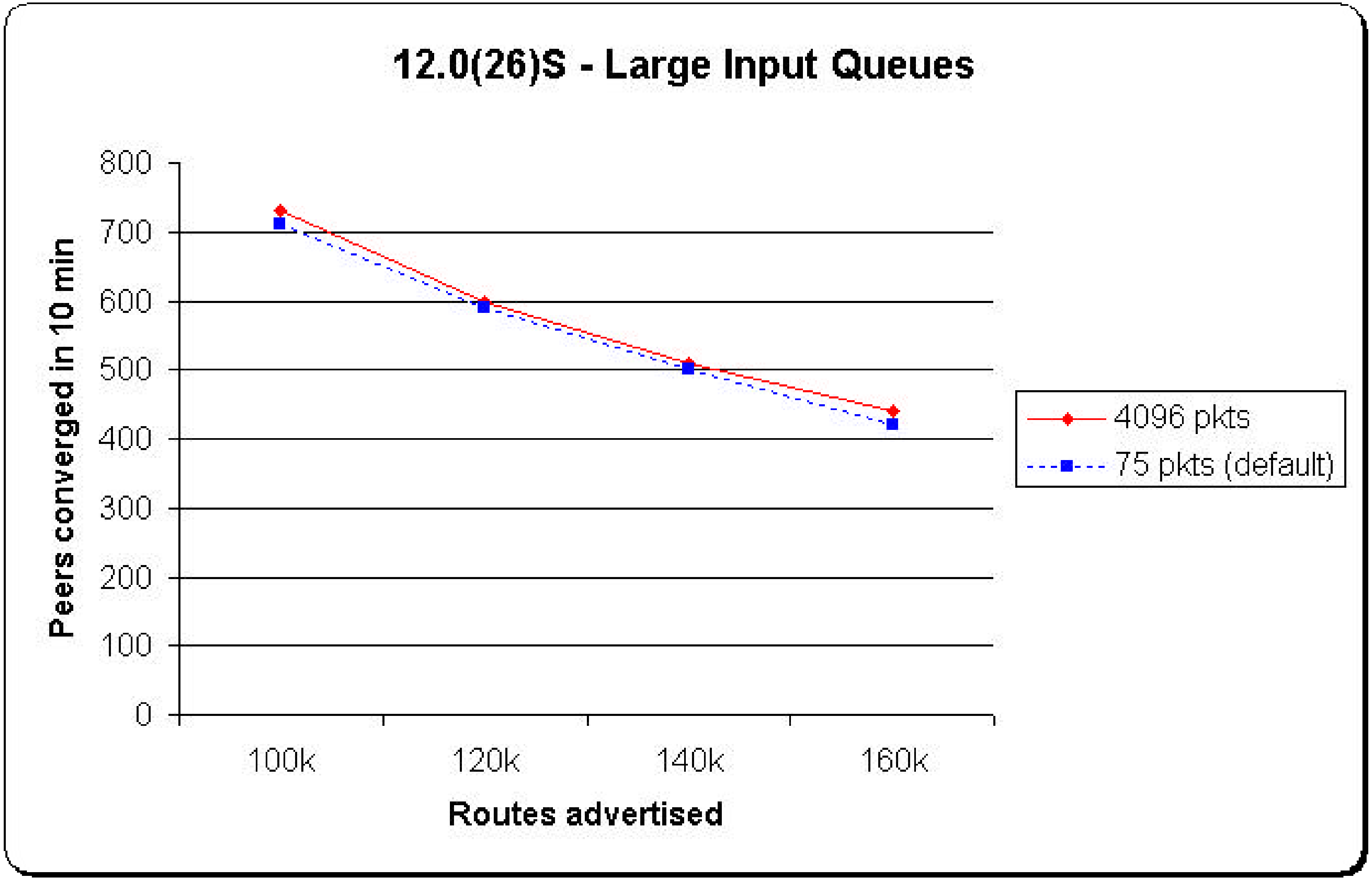
- **Easy solution**

  Just set your input queues or SPD headroom to 1000

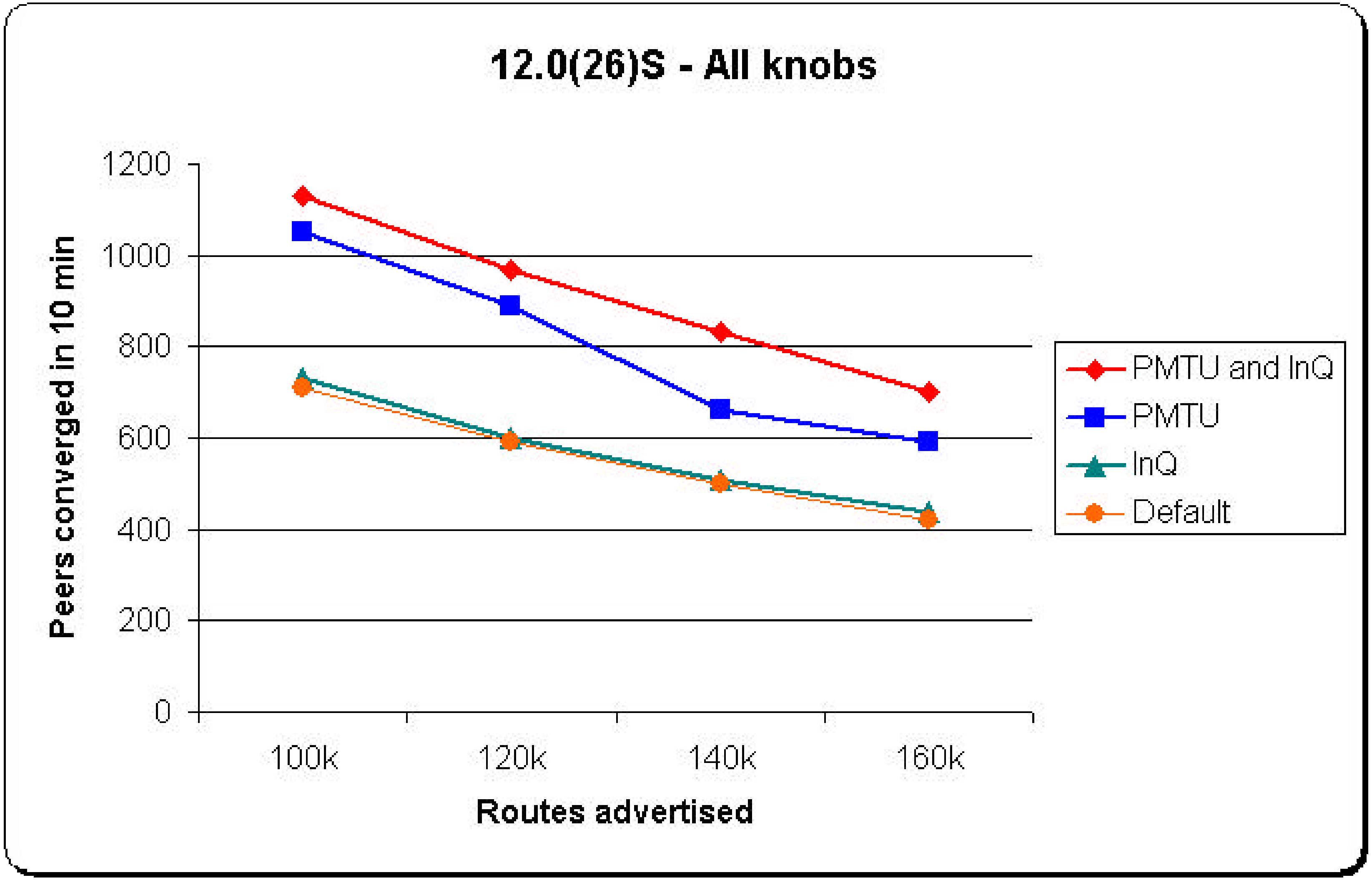
  1000 is deep enough for the # of routes/peers that we see on a heavily loaded box today

Input Queues

Cisco.com



Input Queues and PMTU



References

Cisco.com

- **TAC BGP pages—very nice**

  http://www.cisco.com/cgi-bin/Support/PSP/psp_view.pl?p=Internetworking:BGP

- **BGP Case Studies**

  http://www.cisco.com/warp/public/459/bgp-toc.html

- **Internet Routing Architectures**

  http://www.ciscopress.com/book.cfm?series=1&book=155

- **Standards**

  RFC 1771, 1997, etc…

  http://www.rfc-editor.org/rfcsearch.html

  http://search.ietf.org/search/brokers/internet-drafts/query.html

CISCO SYSTEMS ®