# BGP for Internet Service Providers

Philip Smith <pfs@cisco.com>

NANOG 22, Scottsdale, Arizona

# BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Deploying BGP in an ISP network
- Trouble & Troubleshooting
- Multihoming Examples
- Using Communities

# BGP Basics

## What is this BGP thing?

# Border Gateway Protocol

- Routing Protocol used to exchange routing information between networks

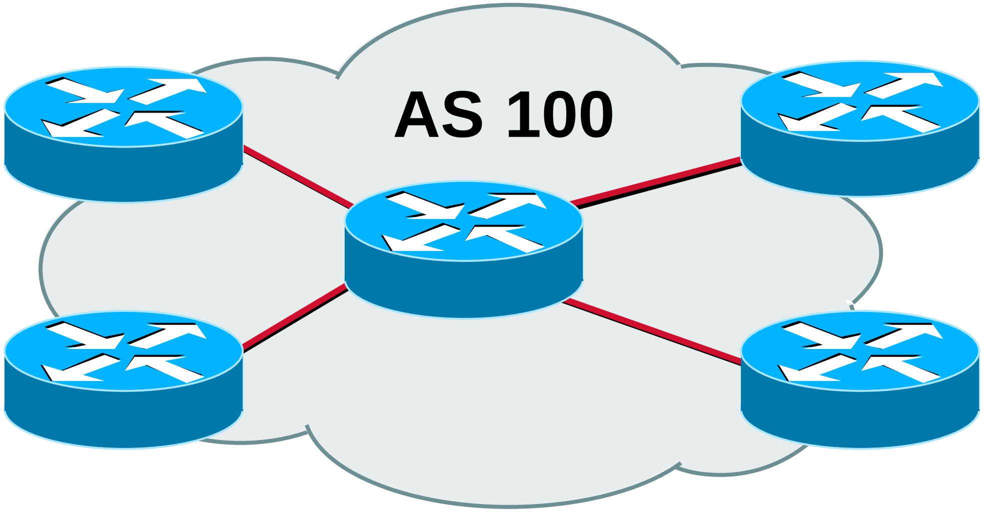
  exterior gateway protocol

- RFC1771

  work in progress to update

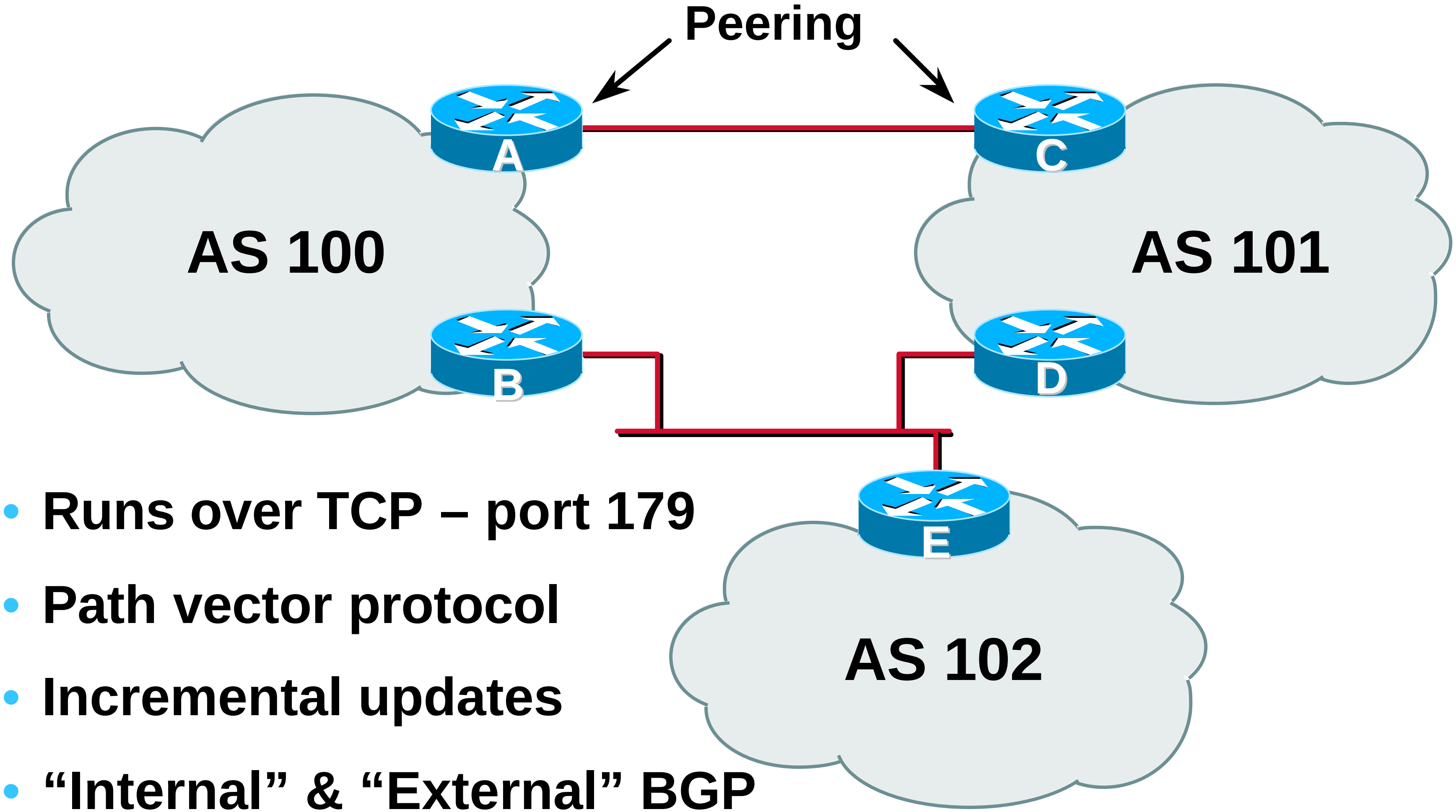
  `draft-ietf-idr-bgp4-12.txt`

# Autonomous System (AS)



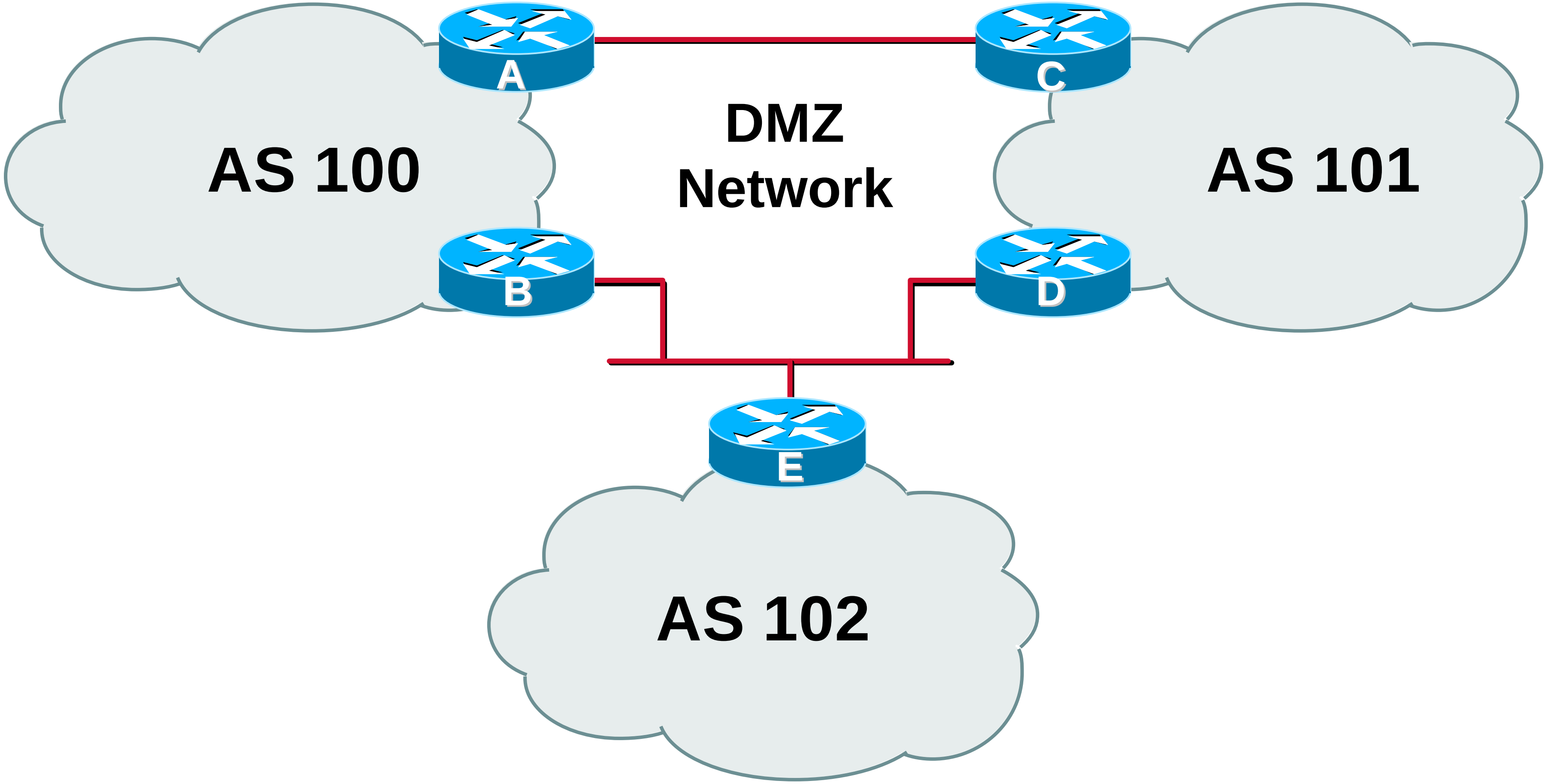
- Collection of networks with same routing policy
- Single routing protocol
- Usually under single ownership, trust and administrative control

# BGP Basics



- Runs over TCP – port 179
- Path vector protocol
- Incremental updates
- “Internal” & “External” BGP

# Demarcation Zone (DMZ)

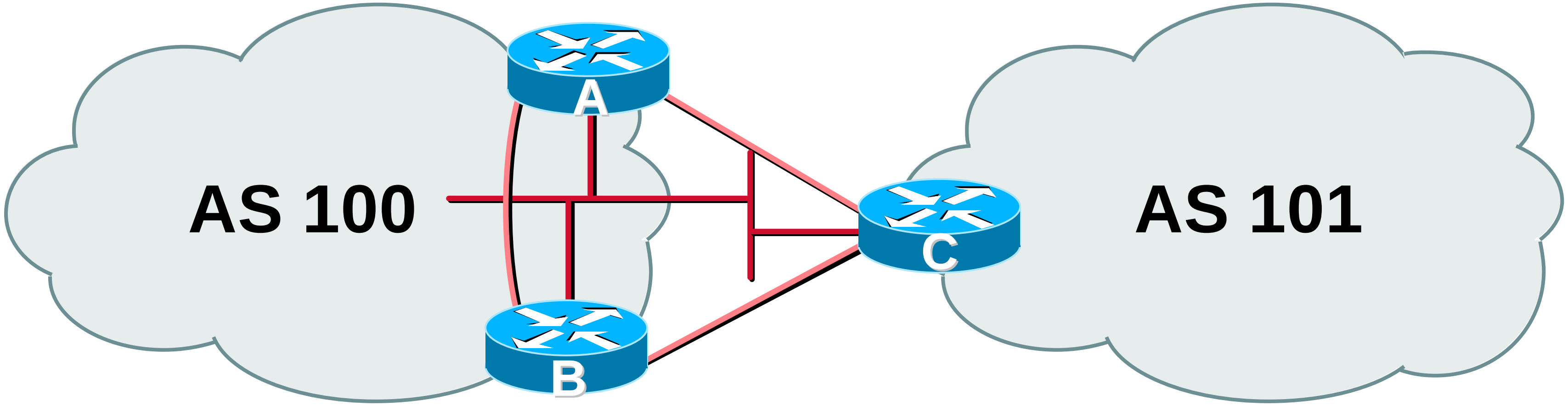


- **Shared network between ASes**

# BGP General Operation

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

# External BGP Peering (eBGP)



- Between BGP speakers in different AS
- Should be directly connected
- Never 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
 neighbor 222.222.10.1 prefix-list RouterC 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
 neighbor 222.222.10.2 prefix-list RouterA out
```

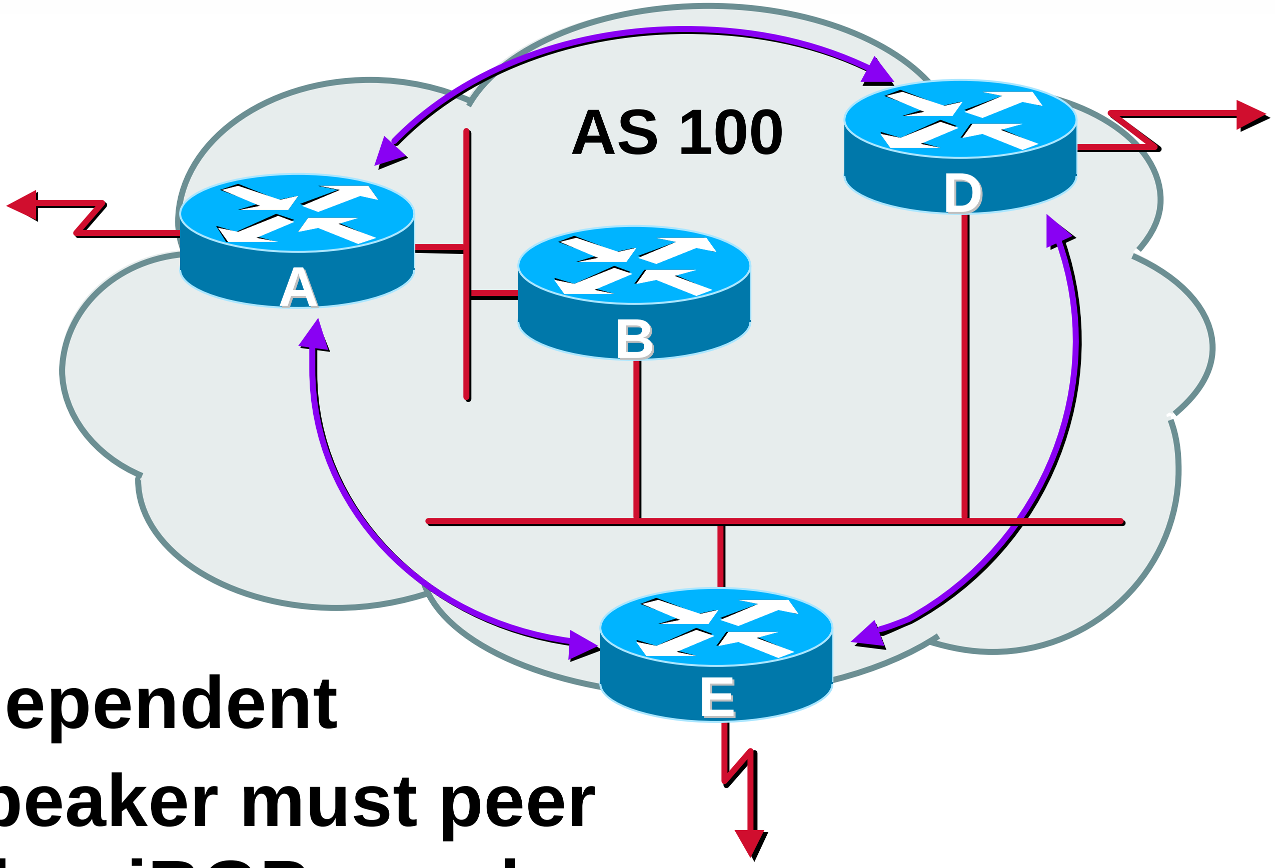
# Internal BGP (iBGP)

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

  they originate connected networks

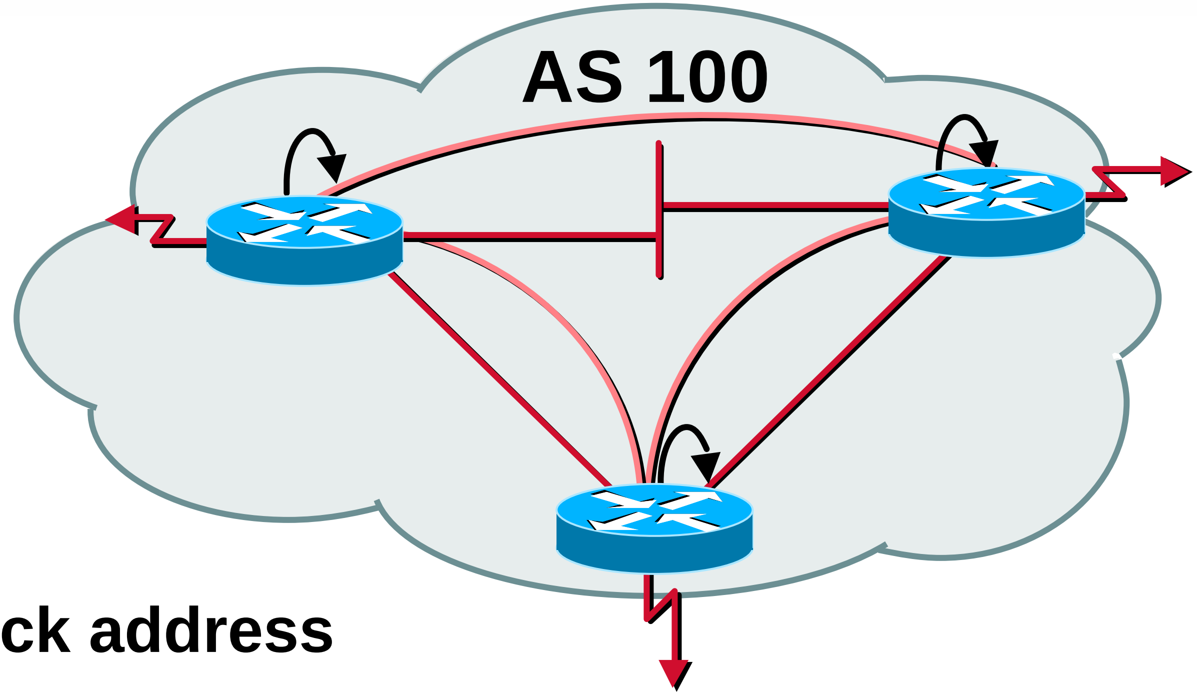
  they do not pass on prefixes learned from other iBGP speakers

# 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**

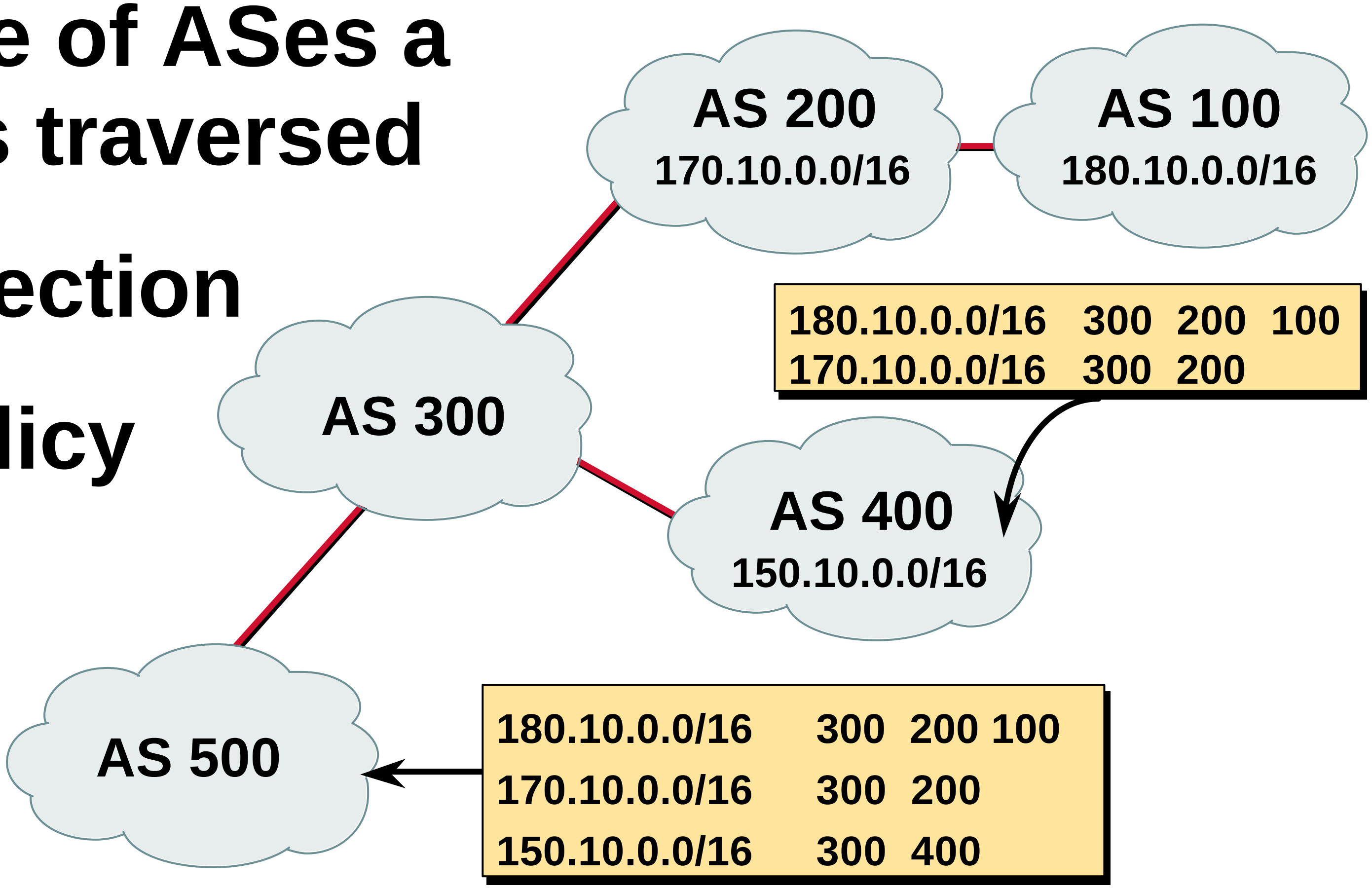
```
interface loopback 0
ip address 215.10.7.2 255.255.255.255
router bgp 100
  network 220.220.5.0
  neighbor 215.10.7.1 remote-as 100
  neighbor 215.10.7.1 update-source loopback0
  neighbor 215.10.7.3 remote-as 100
  neighbor 215.10.7.3 update-source loopback0
```

# BGP Attributes

## Recap

# AS-Path

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



AS 200
170.10.0.0/16

AS 100
180.10.0.0/16

AS 300

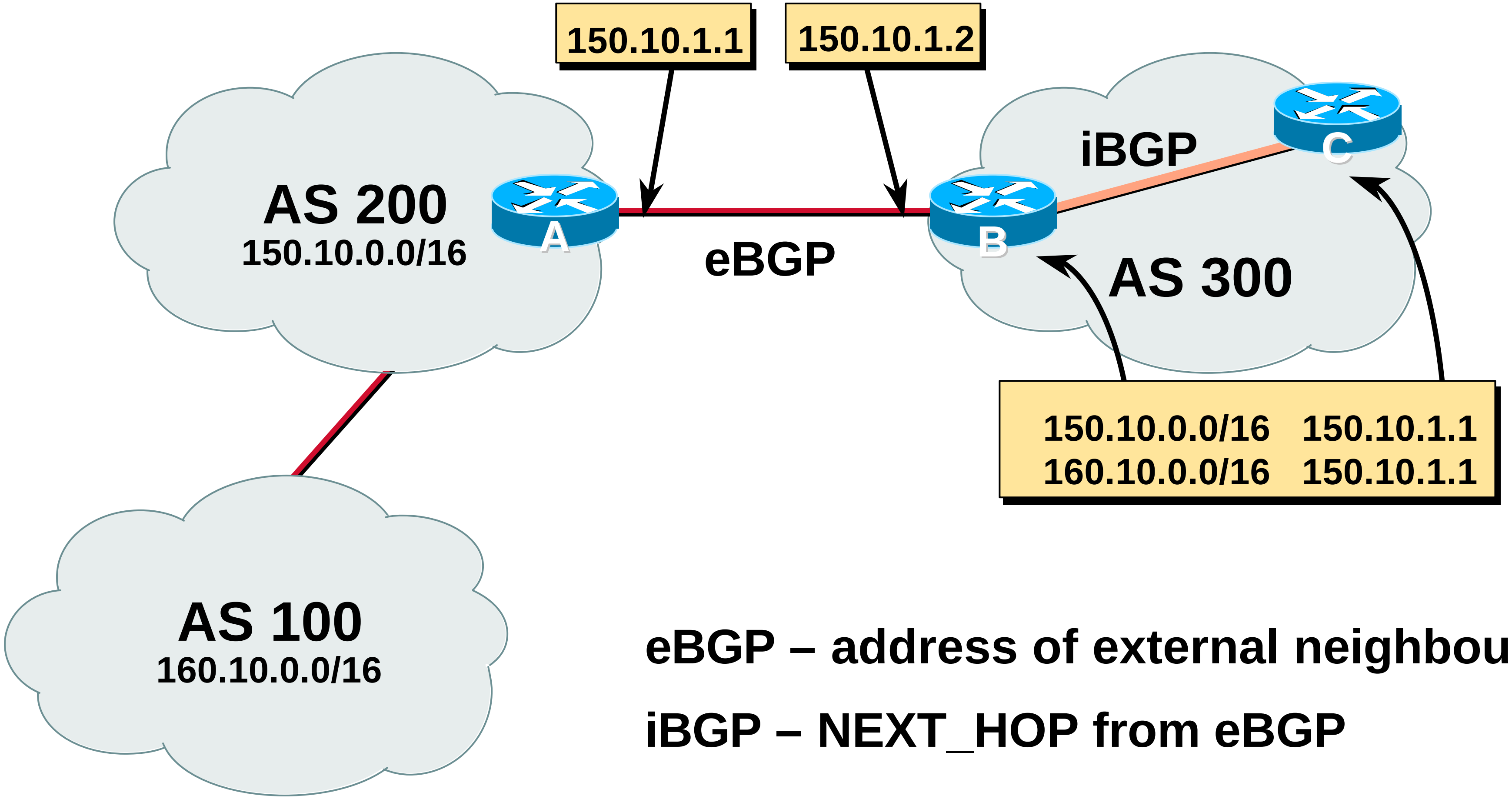
AS 400
150.10.0.0/16

AS 500

| | |
|---|---|
| 180.10.0.0/16 | 300 200 100 |
| 170.10.0.0/16 | 300 200 |

| | |
|---|---|
| 180.10.0.0/16 | 300 200 100 |
| 170.10.0.0/16 | 300 200 |
| 150.10.0.0/16 | 300 400 |

# Next Hop

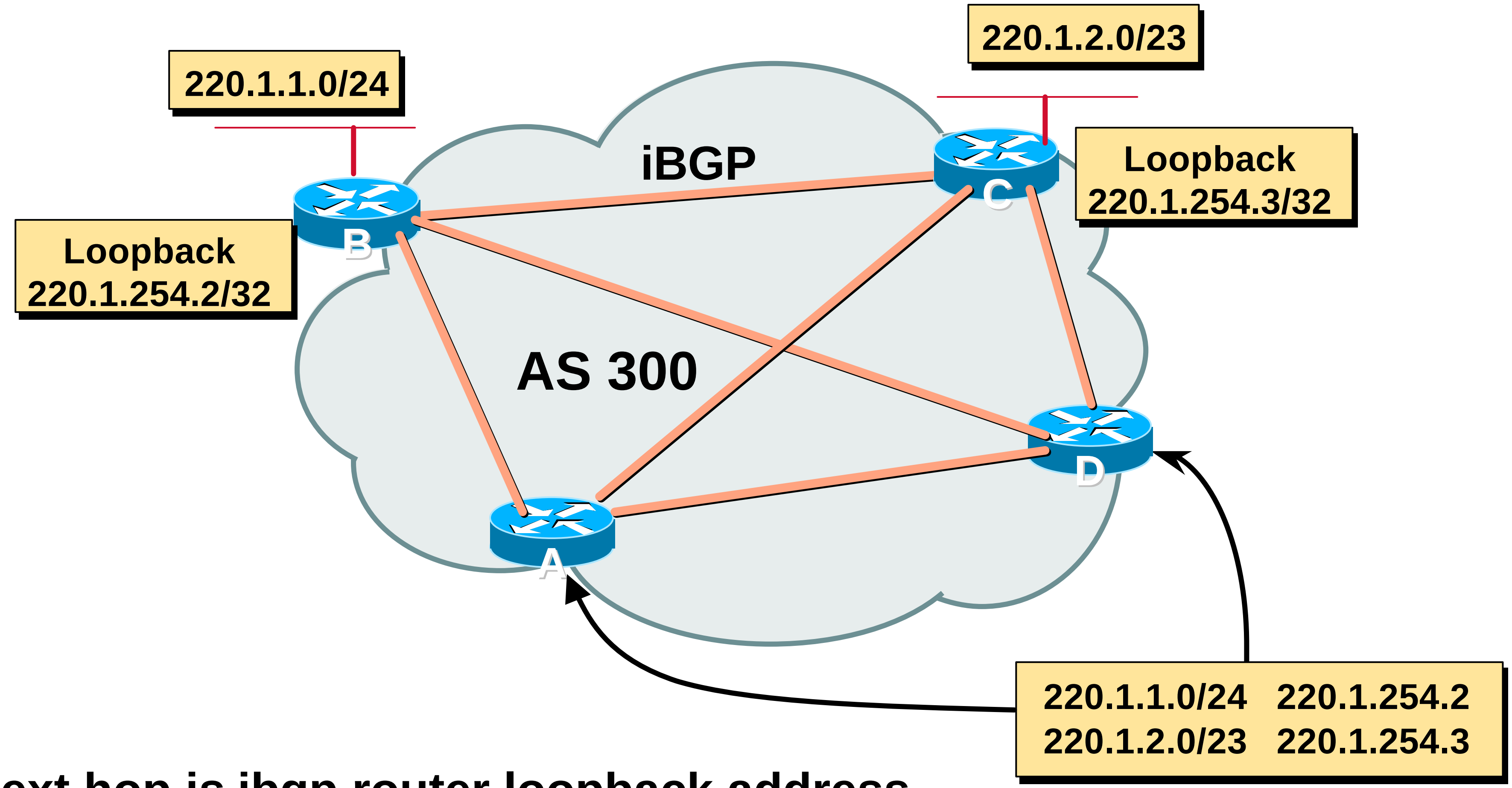


| | |
|---|---|
| 150.10.0.0/16 | 150.10.1.1 |
| 160.10.0.0/16 | 150.10.1.1 |

eBGP – address of external neighbour

iBGP – NEXT_HOP from eBGP

# iBGP Next Hop

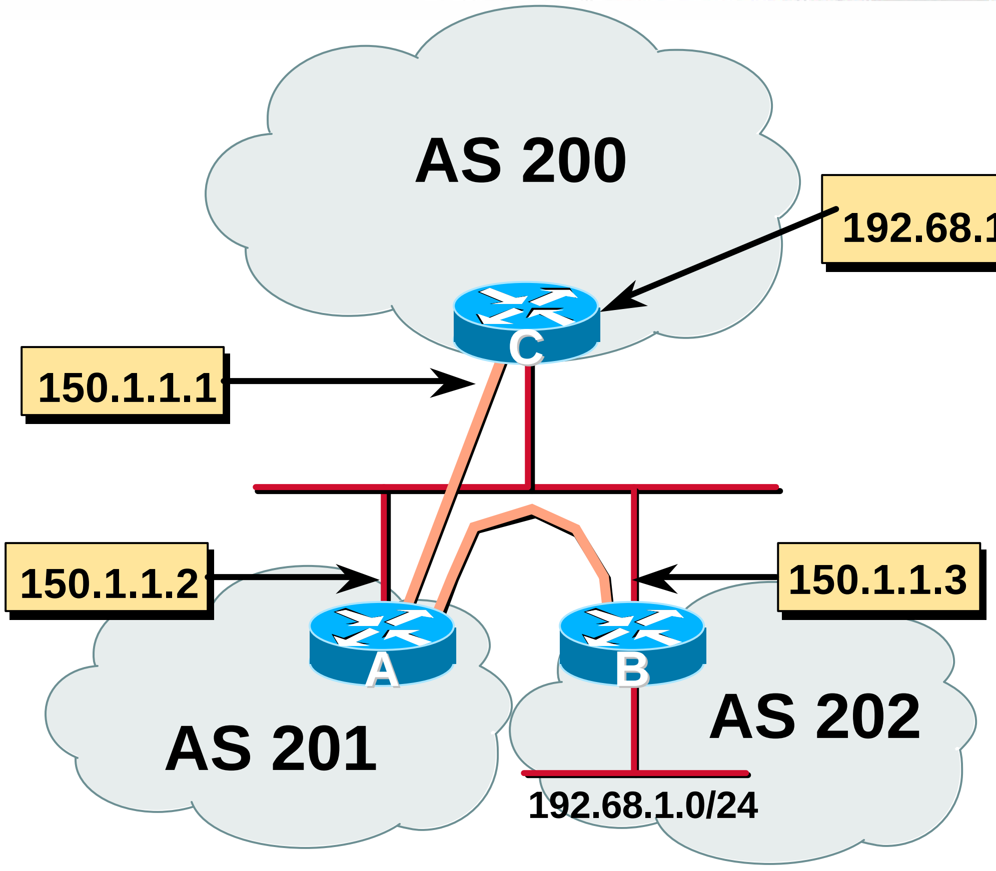


| | |
|---|---|
| 220.1.1.0/24 | 220.1.254.2 |
| 220.1.2.0/23 | 220.1.254.3 |

**Next hop is ibgp router loopback address**

**Recursive route look-up**

# Third Party Next Hop



- eBGP between Router A and Router C
- eBGP between Router A and Router B
- 192.68.1/24 prefix has next hop address of 150.1.1.3 – this is passed on to Router C instead of 150.1.1.2

# Next Hop (summary)

- 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

- Conveys the origin of the prefix
- “Historical” attribute
- Influences best path selection
- Three values: IGP, EGP, incomplete

  IGP – generated by BGP network statement

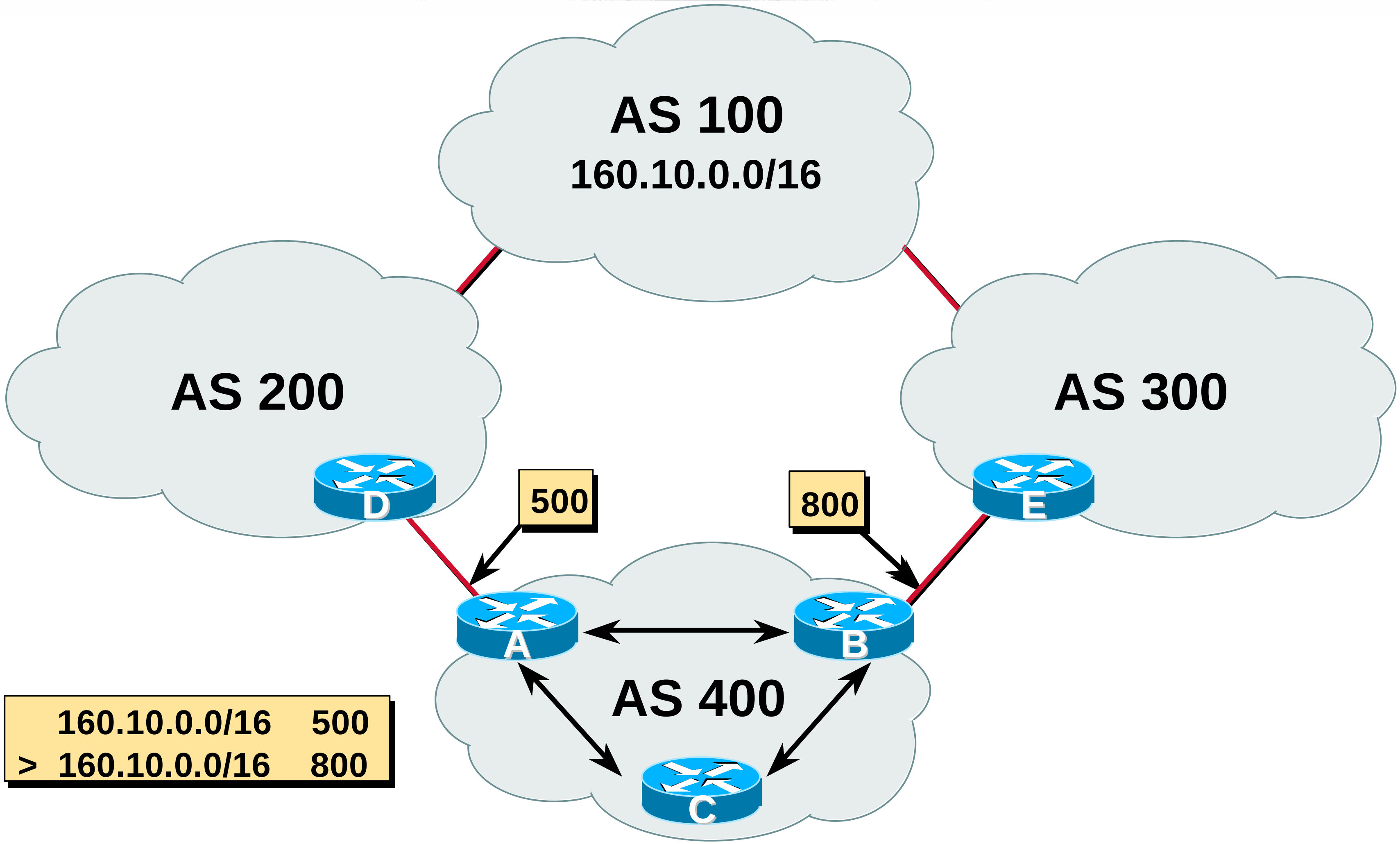
  EGP – generated by EGP

  incomplete – redistributed from another routing protocol

# Aggregator

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

# Local Preference



```
   160.10.0.0/16    500
>  160.10.0.0/16    800
```

# Local Preference

- Local to an AS – non-transitive

  Default local preference is 100

- Used to influence BGP path selection

  determines best path for *outbound* traffic

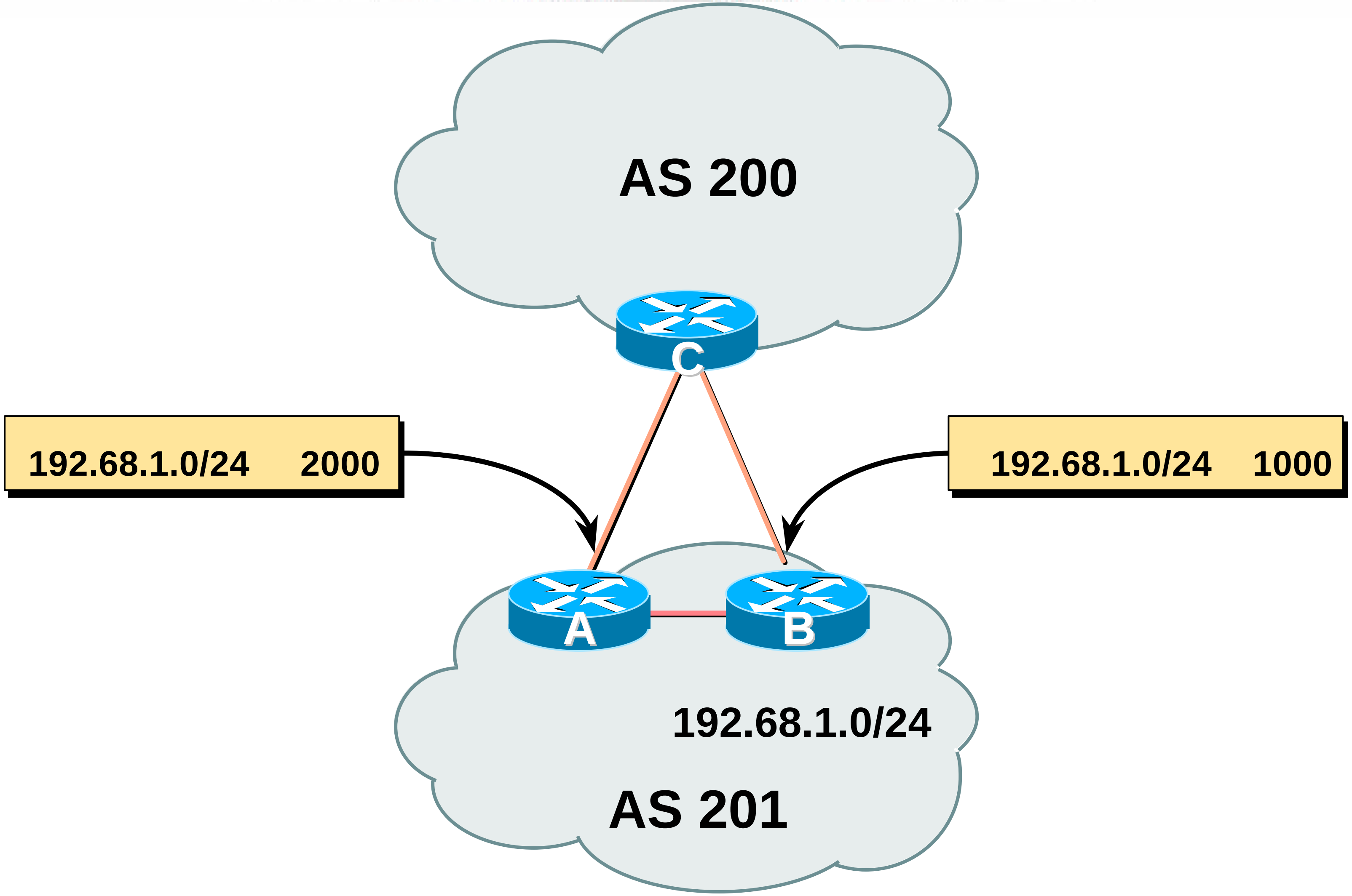
- Path with highest local preference wins

# 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
```

# Multi-Exit Discriminator (MED)



# Multi-Exit Discriminator

- Inter-AS – non-transitive
- 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

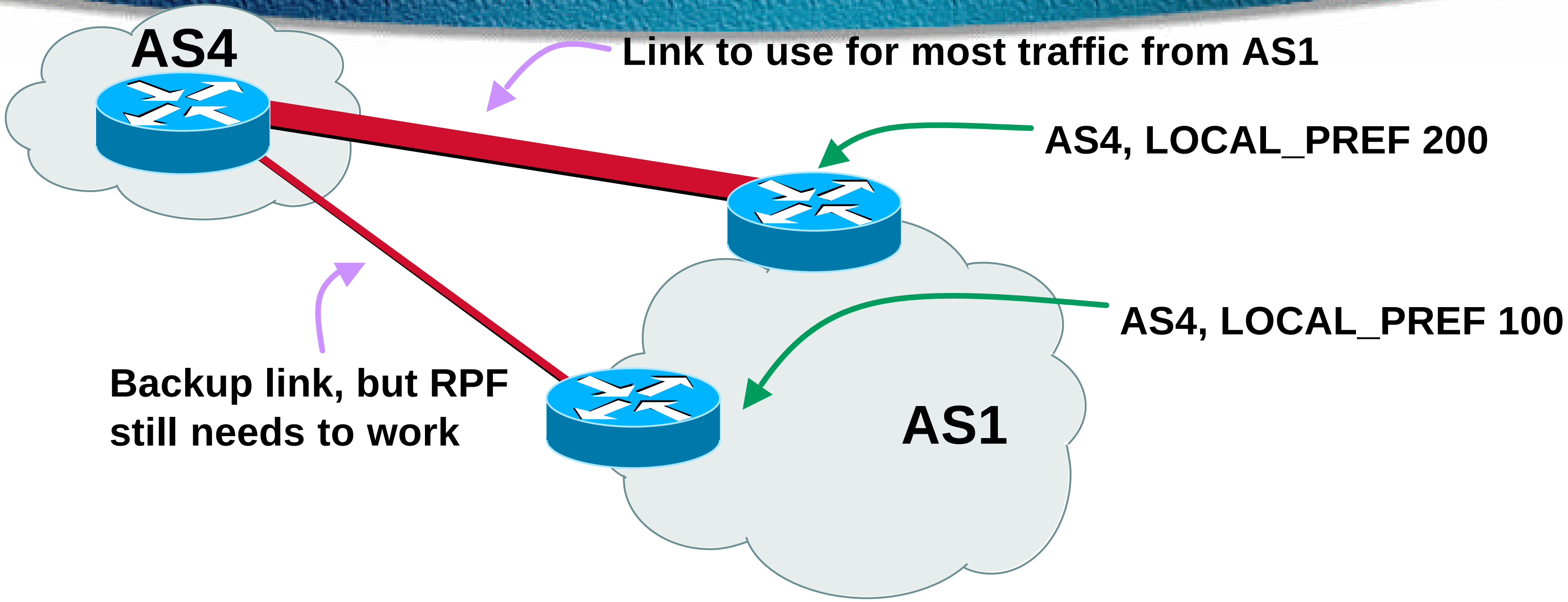
  set metric-type internal in route-map

# 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
```

# Weight – used to deploy RPF


AS4
Link to use for most traffic from AS1
AS4, LOCAL_PREF 200
AS4, LOCAL_PREF 100
Backup link, but RPF still needs to work
AS1

- 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

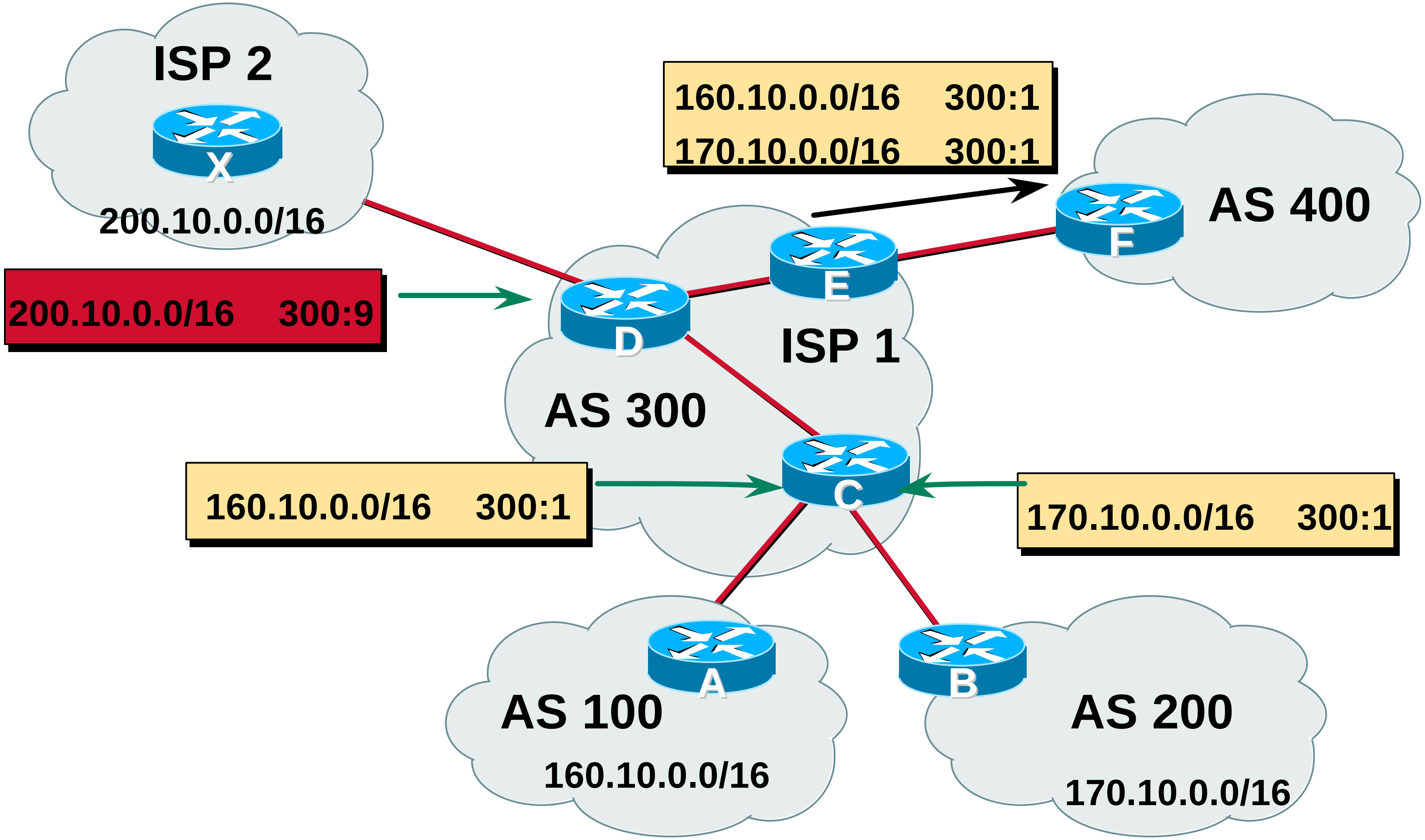
# Community

- BGP attribute
- Described in RFC1997
- 32 bit integer

  Represented as two 16bit integers
- Used to group destinations

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

# Community


ISP 2
X
200.10.0.0/16
200.10.0.0/16 300:9
160.10.0.0/16 300:1
170.10.0.0/16 300:1
AS 400
F
E
D
ISP 1
AS 300
160.10.0.0/16 300:1
C
170.10.0.0/16 300:1
A
B
AS 100
160.10.0.0/16
AS 200
170.10.0.0/16

# Well-Known Communities

- no-export

  do not advertise to eBGP peers

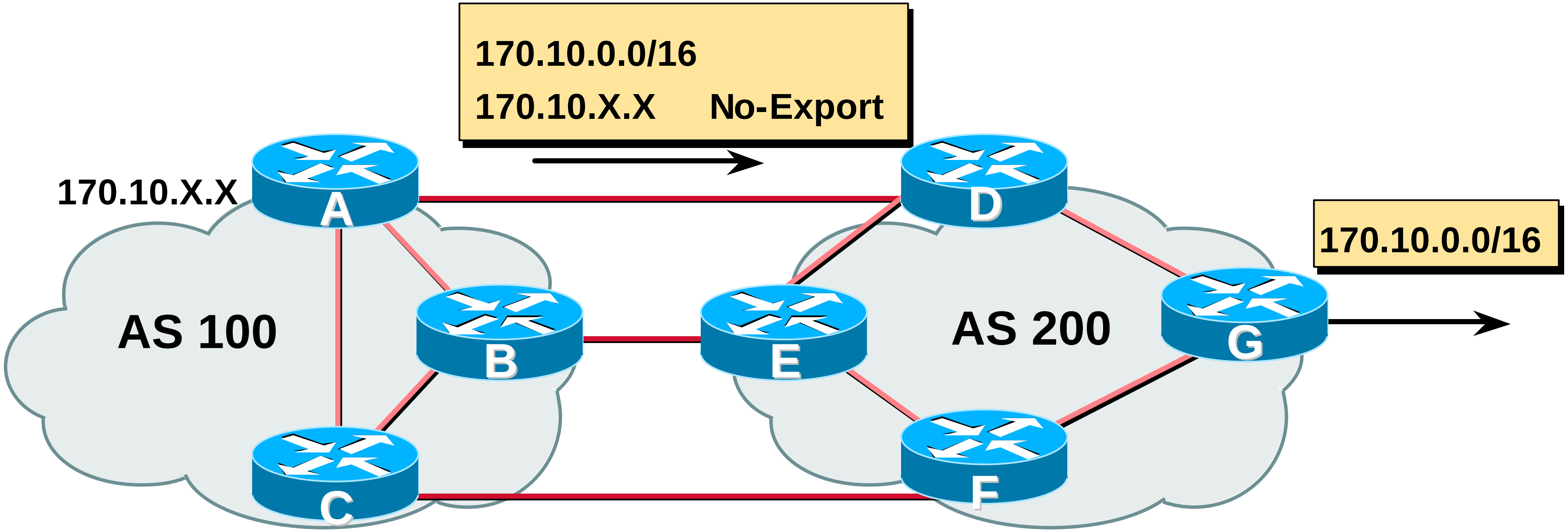
- no-advertise

  do not advertise to any peer

- local-AS

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

# No-Export Community



- **AS100 announces aggregate and subprefixes**

  **aim is to improve loadsharing 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

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

# BGP Path Selection Algorithm (continued)

- 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)

- Prefer eBGP path over iBGP path
- Path with lowest IGP metric to next-hop
- Lowest router-id (originator-id for reflected routes)
- Shortest Cluster-List

  Client must be aware of Route Reflector attributes!

- Lowest neighbour IP address

# Applying Policy with BGP

Control!

# Applying Policy with BGP

- Applying Policy

  Decisions based on AS path, community or the prefix

  Rejecting/accepting selected routes

  Set attributes to influence path selection

- Tools:

  Prefix-list (filter prefixes)

  Filter-list (filter ASes)

  Route-maps and communities

# Policy Control
# Prefix List

- Filter routes based on prefix
- Inbound and Outbound

```
router bgp 200
 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
```

# Policy Control
# Filter List

- Filter routes based on AS path
- Inbound and Outbound

```
router bgp 100
 neighbor 220.200.1.1 remote-as 210
 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 1800 |
| ^1800_ | Received from 1800 |
| _1800_ | Via 1800 |
| _790_1800_ | Passing through 1800 then 790 |
| _(1800_)+ | Match at least one of 1800 in sequence |
| _\(65350\)_ | Via 65350 (confederation AS) |

# Policy Control
# Route Maps

- 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*

# Policy Control
# Route Maps

- **Example using 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
!
route-map infilter permit 30
!
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 using filter lists

```
router bgp 100
 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
!
route-map filter-on-as-path permit 30
!
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
# Setting Communities

- Example Configuration

```
router bgp 100
 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
!
ip prefix-list NO-ANNOUNCE permit 172.168.0.0/16 ge 17
```

# 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
```

# BGP Capabilities

## Extending BGP

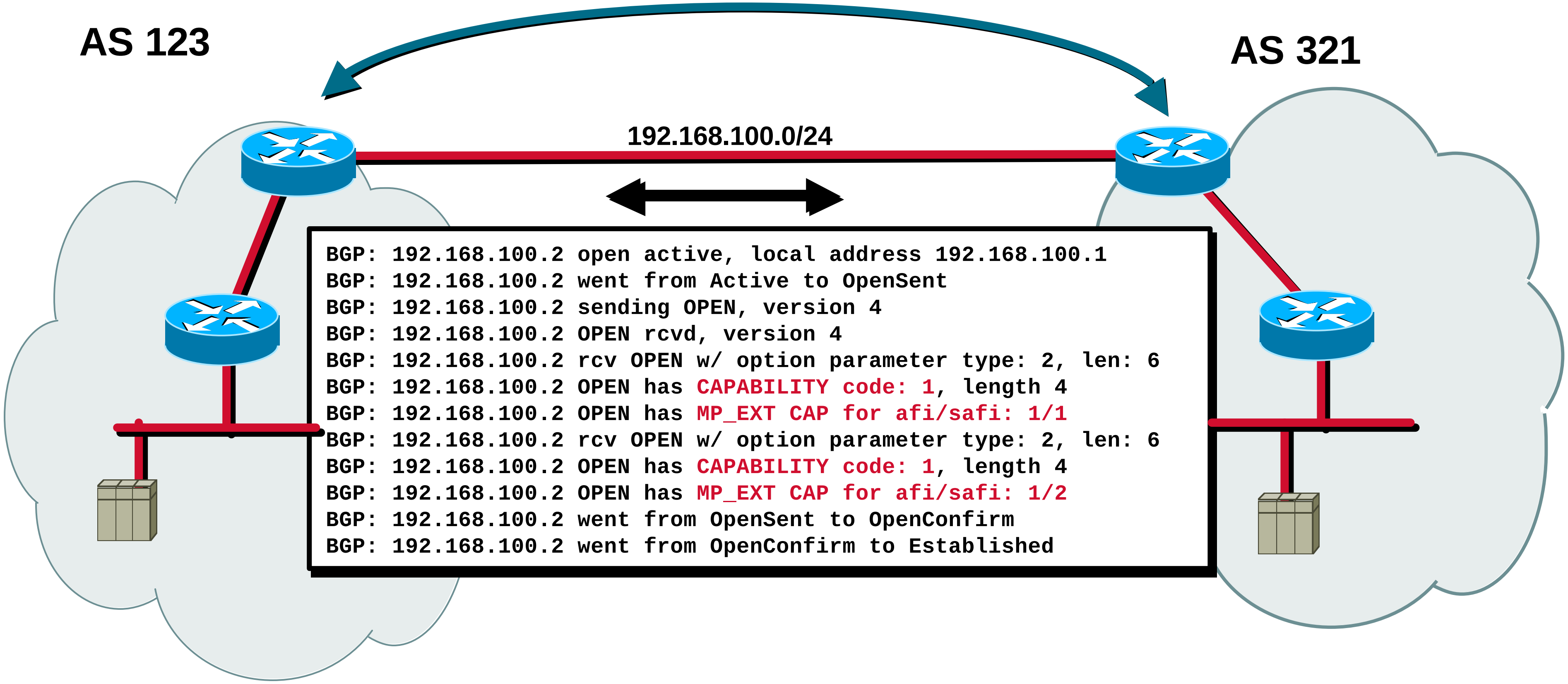
# BGP Capabilities

- Documented in RFC2842
- Capabilities parameters passed in BGP open message
- Unknown or unsupported capabilities will result in NOTIFICATION message
- Current capabilities are:

| | | |
|---|---|---|
| 0 | Reserved | [RFC2842] |
| 1 | Multiprotocol Extensions for BGP-4 | [RFC2858] |
| 2 | Route Refresh Capability for BGP-4 | [RFC2918] |
| 3 | Cooperative Route Filtering Capability | [] |
| 4 | Multiple routes to a destination capability | [RFC3107] |
| 64 | Graceful Restart Capability | [] |

# BGP Capabilities Negotiation

BGP session for unicast and multicast NLRI



AS 123

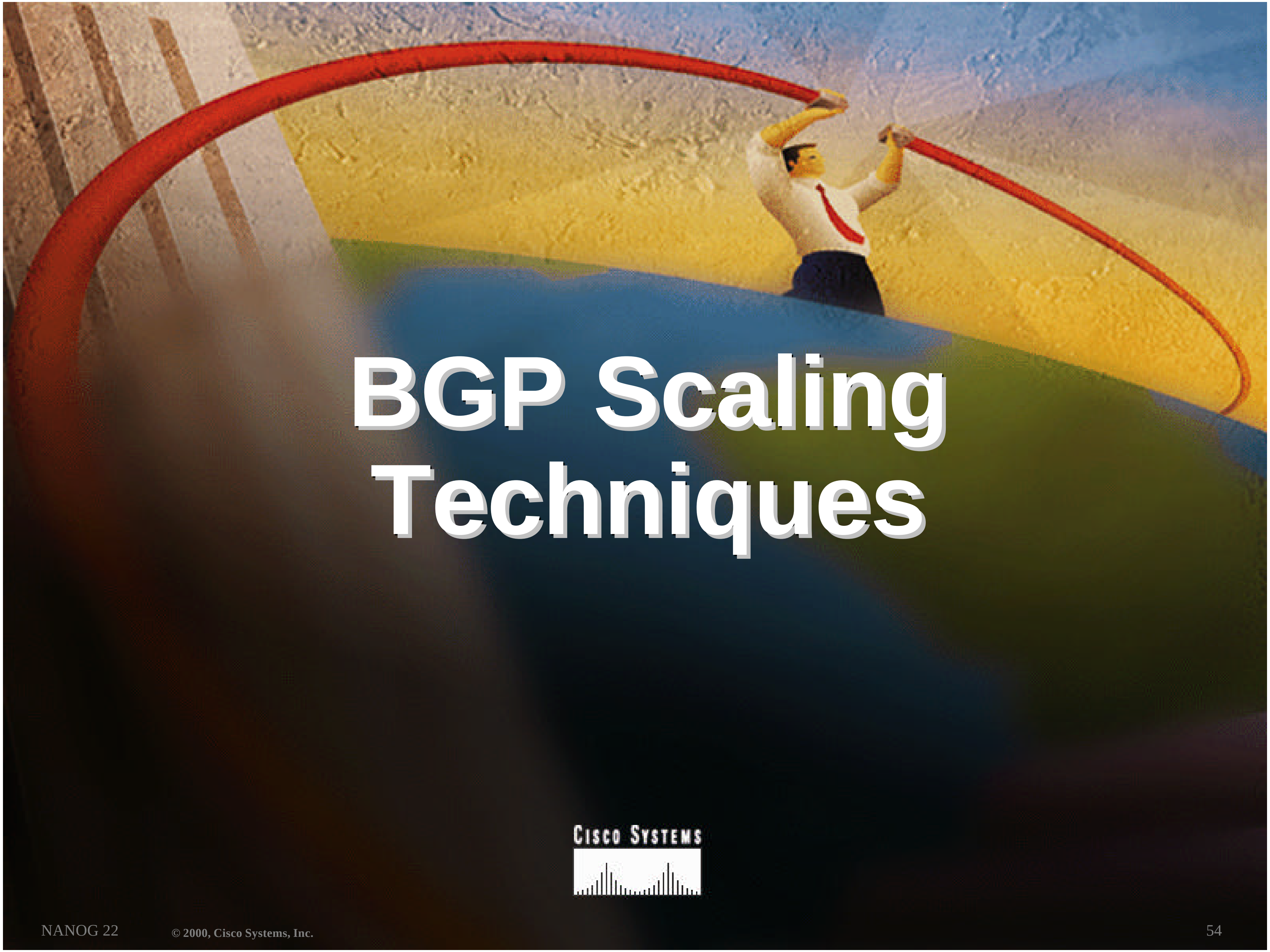
AS 321

192.168.100.0/24

```
BGP: 192.168.100.2 open active, local address 192.168.100.1
BGP: 192.168.100.2 went from Active to OpenSent
BGP: 192.168.100.2 sending OPEN, version 4
BGP: 192.168.100.2 OPEN rcvd, version 4
BGP: 192.168.100.2 rcv OPEN w/ option parameter type: 2, len: 6
BGP: 192.168.100.2 OPEN has CAPABILITY code: 1, length 4
BGP: 192.168.100.2 OPEN has MP_EXT CAP for afi/safi: 1/1
BGP: 192.168.100.2 rcv OPEN w/ option parameter type: 2, len: 6
BGP: 192.168.100.2 OPEN has CAPABILITY code: 1, length 4
BGP: 192.168.100.2 OPEN has MP_EXT CAP for afi/safi: 1/2
BGP: 192.168.100.2 went from OpenSent to OpenConfirm
BGP: 192.168.100.2 went from OpenConfirm to Established
```

# BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Deploying BGP in an ISP network
- Trouble & Troubleshooting
- Multihoming Examples
- Using Communities



# 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?
- How to keep the network stable, scalable, as well as simple?

# BGP Scaling Techniques

- Dynamic Reconfiguration
- Peer groups
- Route flap damping
- Route Reflectors & Confederations

# Dynamic Reconfiguration

## Soft Reconfiguration and Route Refresh

# Soft Reconfiguration

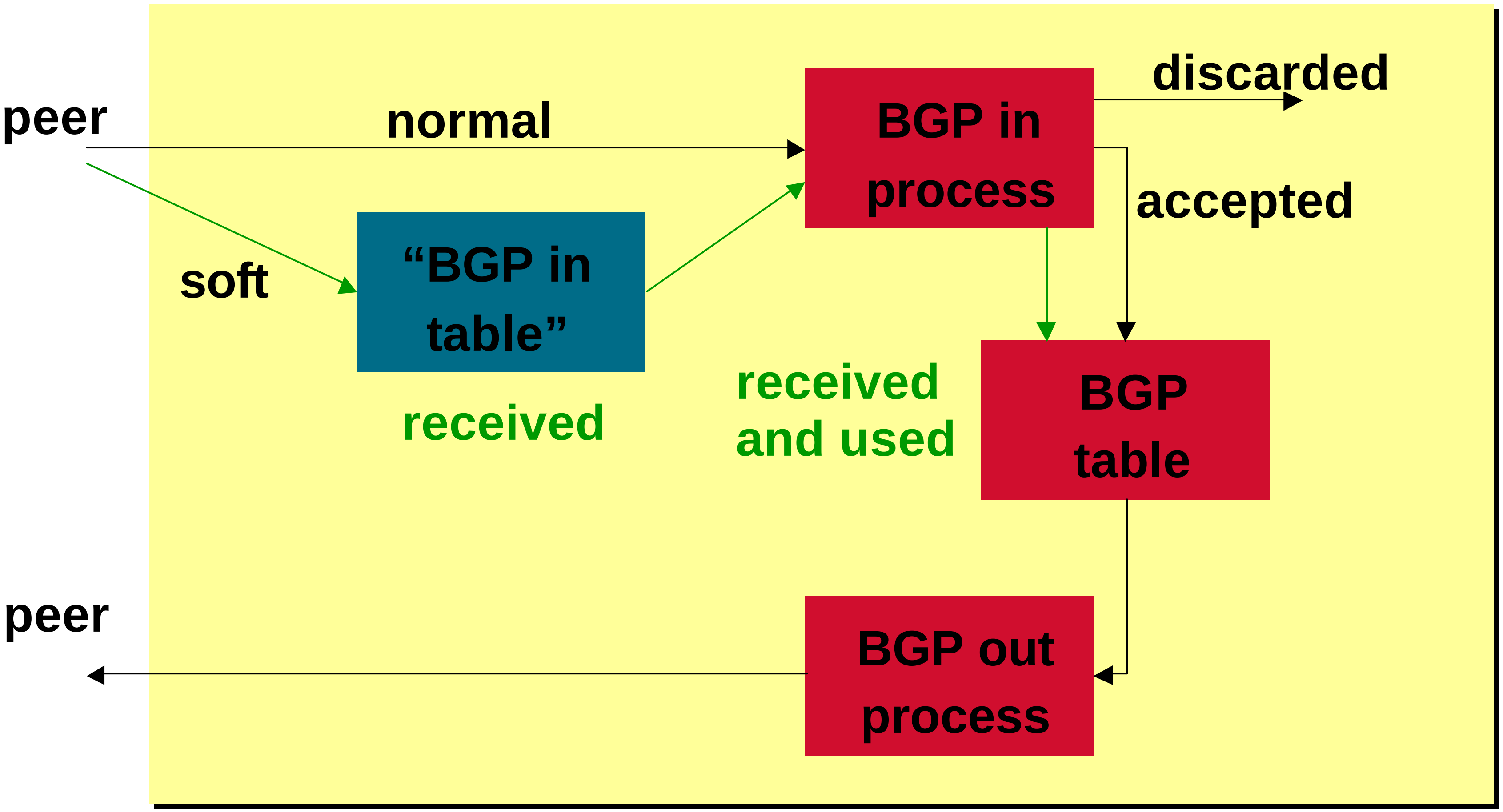
Problem:

- Hard BGP peer clear required after every policy change because the router does not store prefixes that are denied by a filter
- Hard BGP peer clearing consumes CPU and affects connectivity for all networks

Solution:

- Soft-reconfiguration

# Soft Reconfiguration



# Soft Reconfiguration

- New policy is activated without tearing down and restarting the peering session
- Per-neighbour basis
- Use more memory to keep prefixes whose attributes have been changed or have not been accepted

# Configuring Soft reconfiguration

```
router bgp 100
 neighbor 1.1.1.1 remote-as 101
 neighbor 1.1.1.1 route-map infilter in
 neighbor 1.1.1.1 soft-reconfiguration inbound
```

**! *Outbound does not need to be configured* !**

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

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

# 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

# Soft Reconfiguration vs Route Refresh

- Use Route Refresh capability if supported

  find out from “show ip bgp neighbor”

  uses much less memory

- Otherwise use Soft Reconfiguration

- Only hard-reset a BGP peering as a last resort

# Peer Groups

# Peer Groups

Without peer groups

- iBGP neighbours receive same update
- Large iBGP mesh slow to build
- Router CPU wasted on repeat calculations

Solution – peer groups!

- Group peers with same outbound policy
- Updates are generated once per group

# Peer Groups – Advantages

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

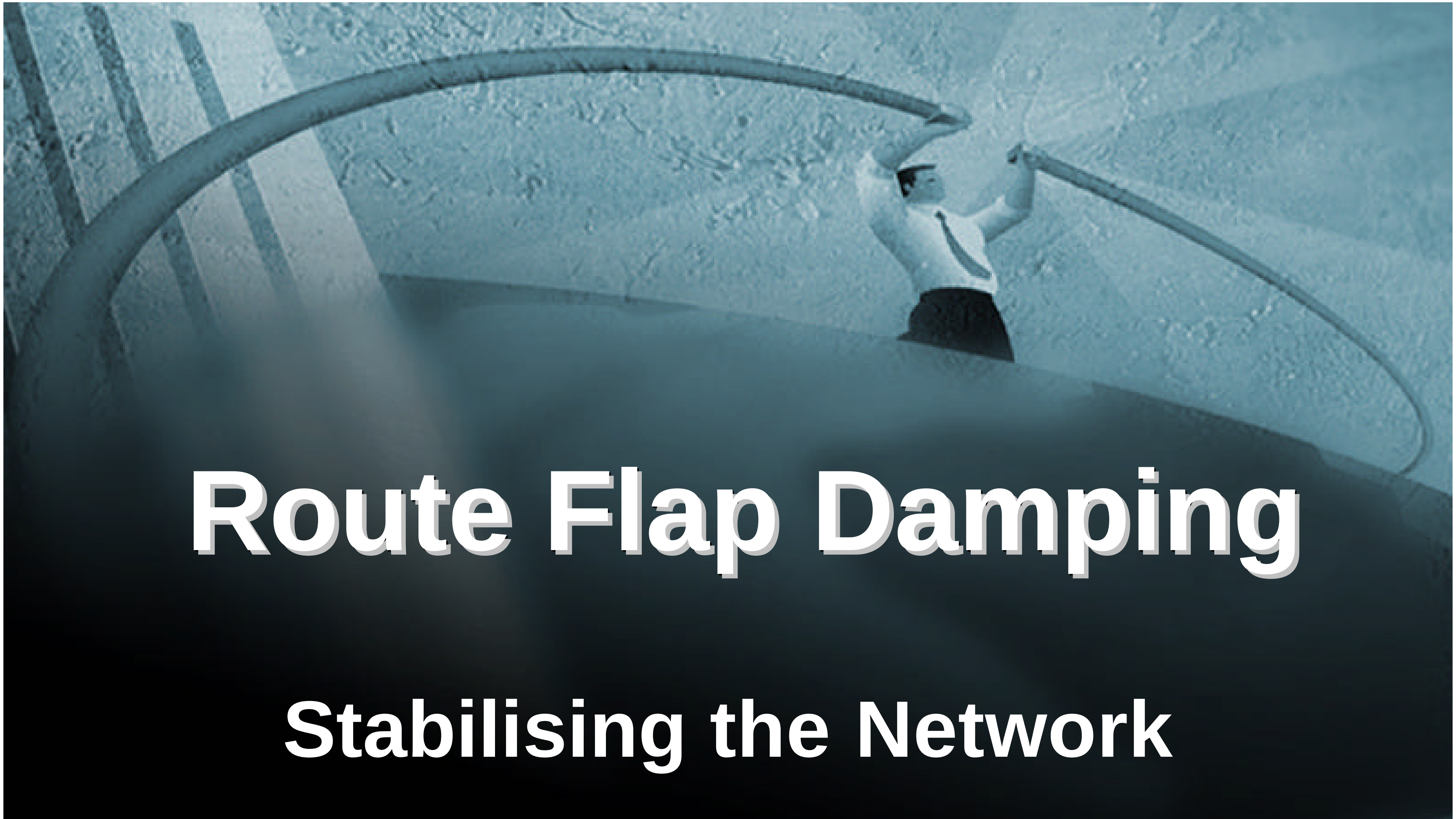
# Configuring Peer Group

```
router bgp 100
 neighbor ibgp-peer peer-group
 neighbor ibgp-peer remote-as 100
 neighbor ibgp-peer update-source loopback 0
 neighbor ibgp-peer send-community
 neighbor ibgp-peer route-map outfilter out
 neighbor 1.1.1.1 peer-group ibgp-peer
 neighbor 2.2.2.2 peer-group ibgp-peer
 neighbor 2.2.2.2 route-map  infilter in
 neighbor 3.3.3.3 peer-group ibgp-peer
```

! *note how 2.2.2.2 has different inbound filter from peer-group* !

# Configuring Peer Group

```
router bgp 109
 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
```



# 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)

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

# Operation

- Add penalty (1000) for each flap

  Change in attribute gets penalty of 500

- Exponentially decay penalty

  half life determines decay rate

- Penalty above suppress-limit

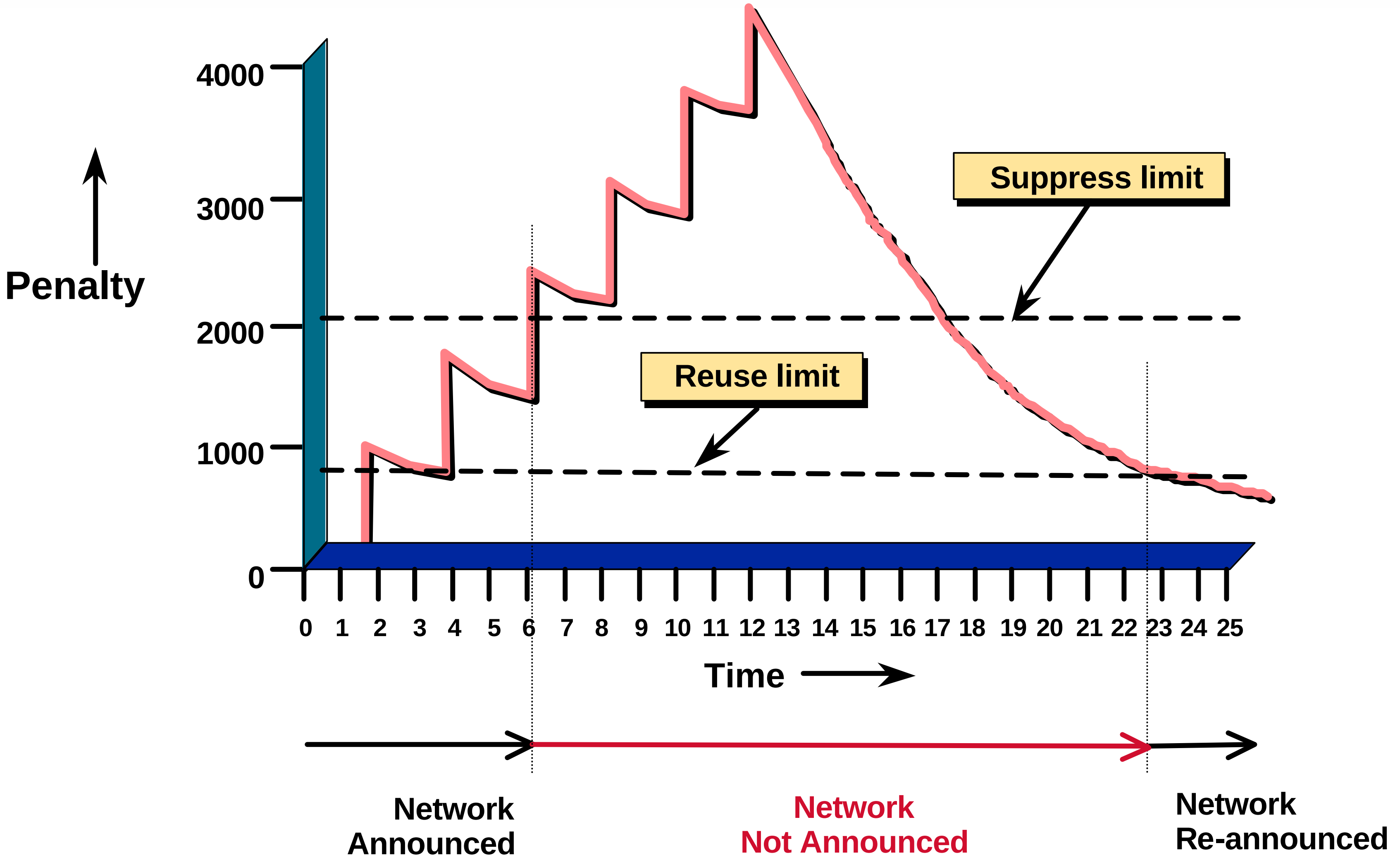
  do not advertise route to BGP peers

- Penalty decayed below reuse-limit

  re-advertise route to BGP peers

  penalty reset to zero when it is half of reuse-limit

# Operation



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

# Operation

- Only applied to inbound announcements from eBGP peers
- Alternate paths still usable
- Controlled by:

  Half-life (default 15 minutes)

  reuse-limit (default 750)

  suppress-limit (default 2000)

  maximum suppress time (default 60 minutes)

# Configuration

## Fixed damping

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

## Selective and variable damping

```
 bgp dampening [route-map <name>]
```

## Variable damping

recommendations for ISPs

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

# 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

- 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

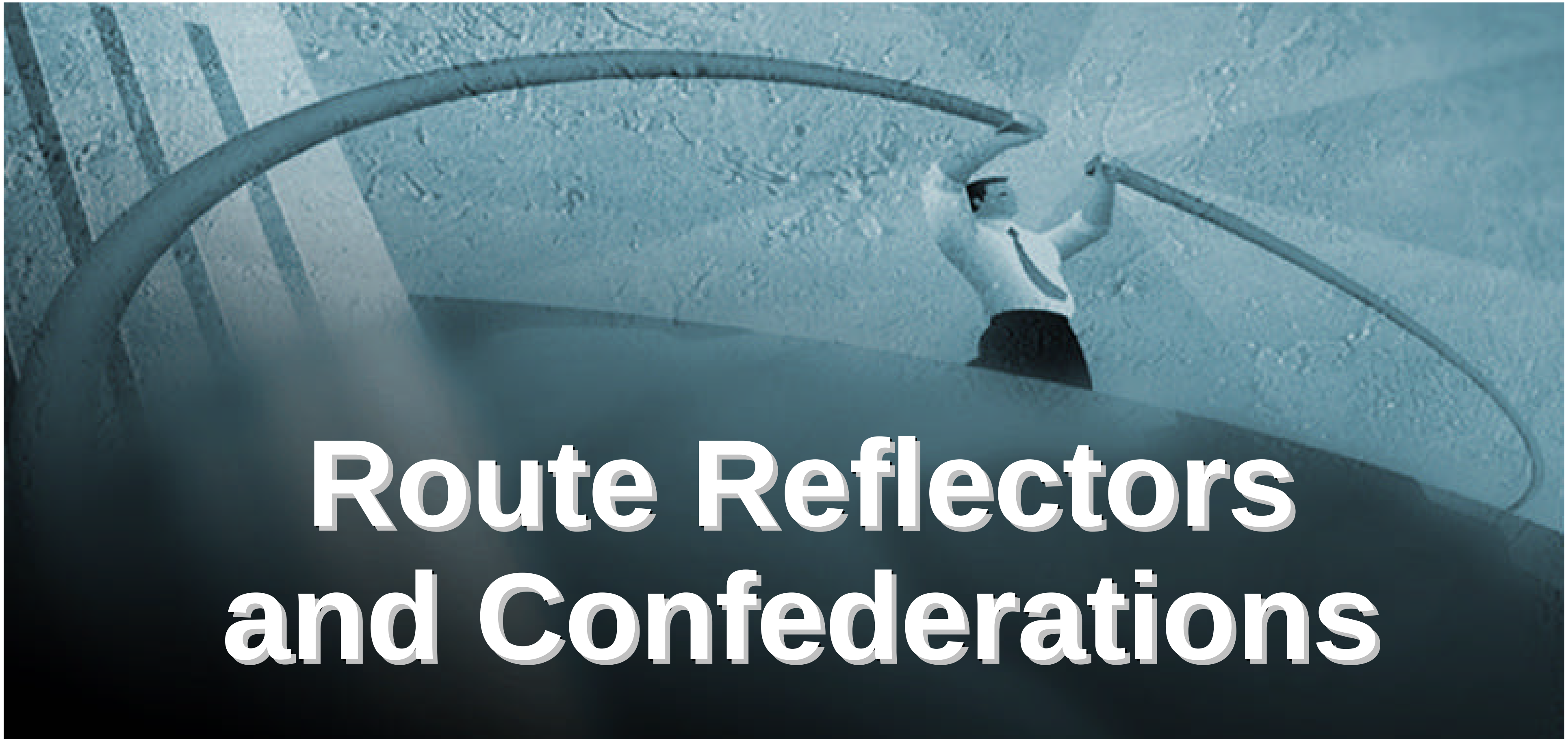
  reuse-limit of 2000 means maximum possible penalty is 8000 – suppress limit is easily reached

# Maths!

- 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 flap damping

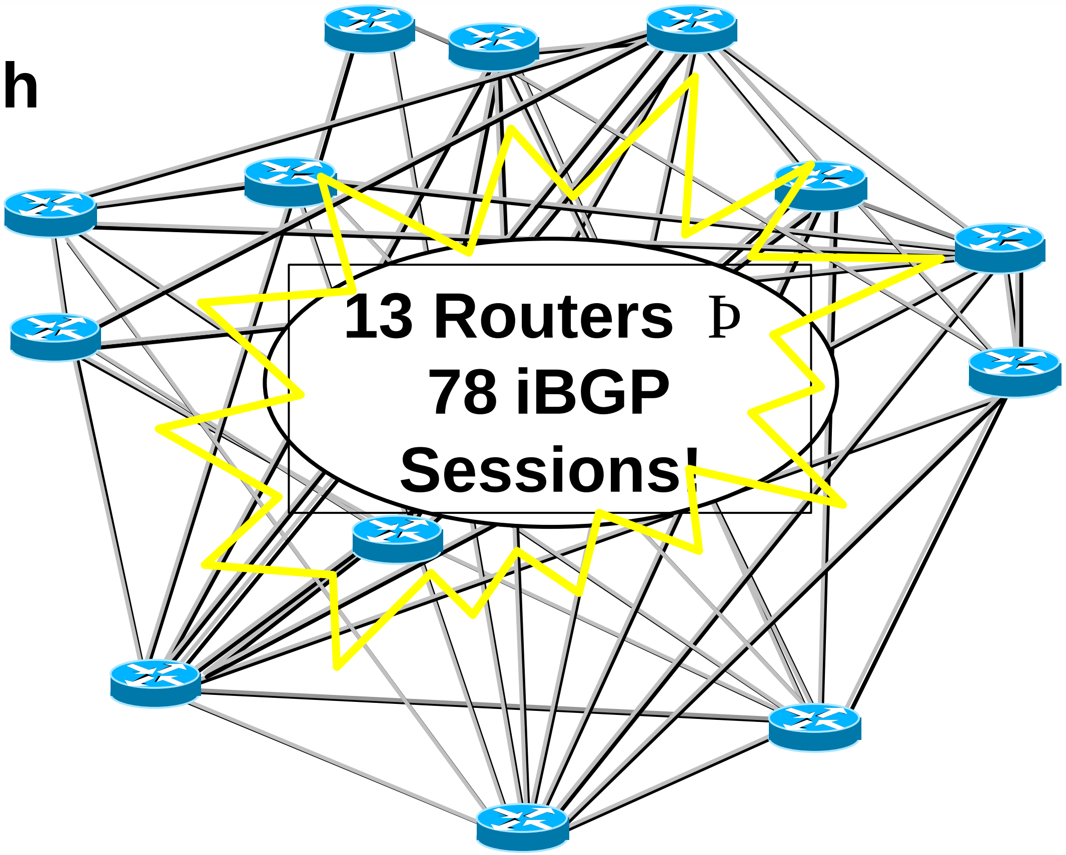


# Route Reflectors and Confederations

# Scaling iBGP mesh

Avoid n(n-1)/2 iBGP mesh

**n=1000 Þ nearly half a million ibgp sessions!**



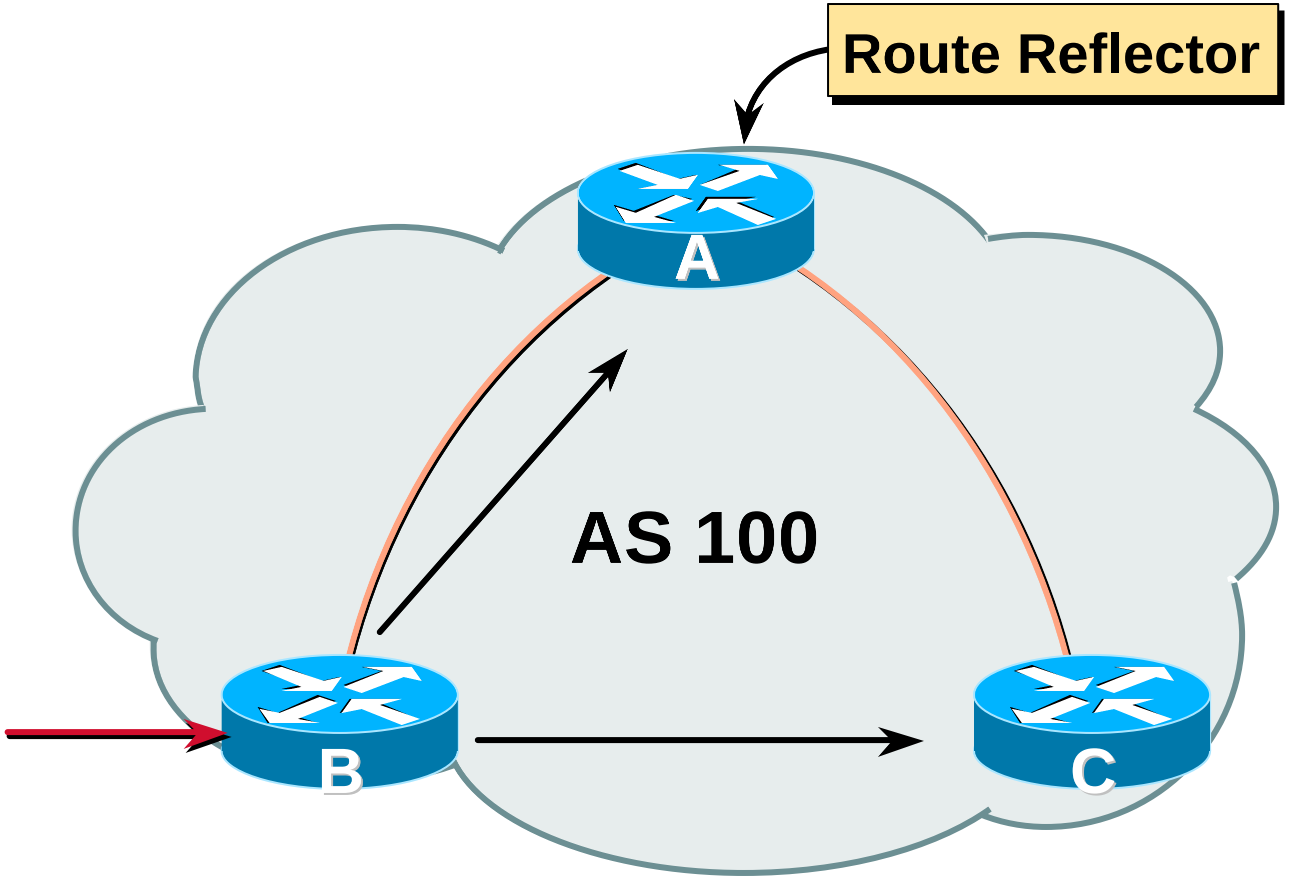
13 Routers Þ 78 iBGP Sessions!

Two solutions

Route reflector – simpler to deploy and run

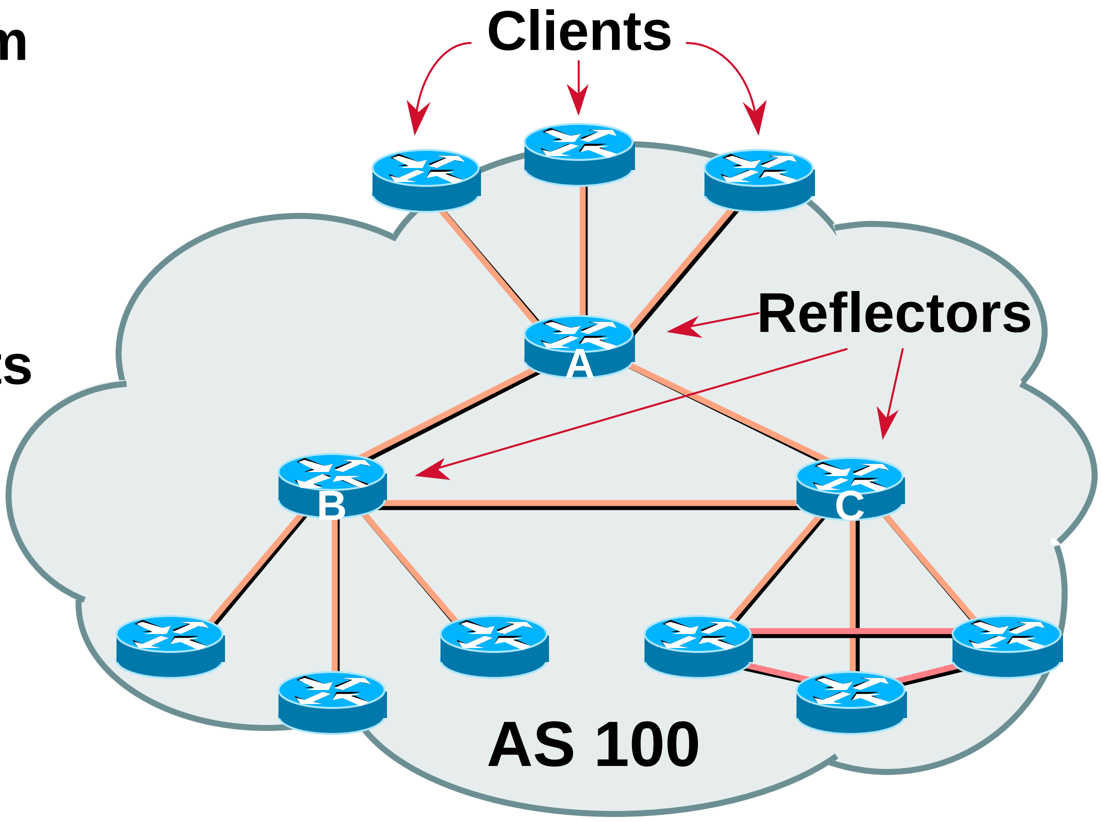
Confederation – more complex, corner case benefits

# Route Reflector: Principle



# Route Reflector

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

- **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 automatically set from 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!

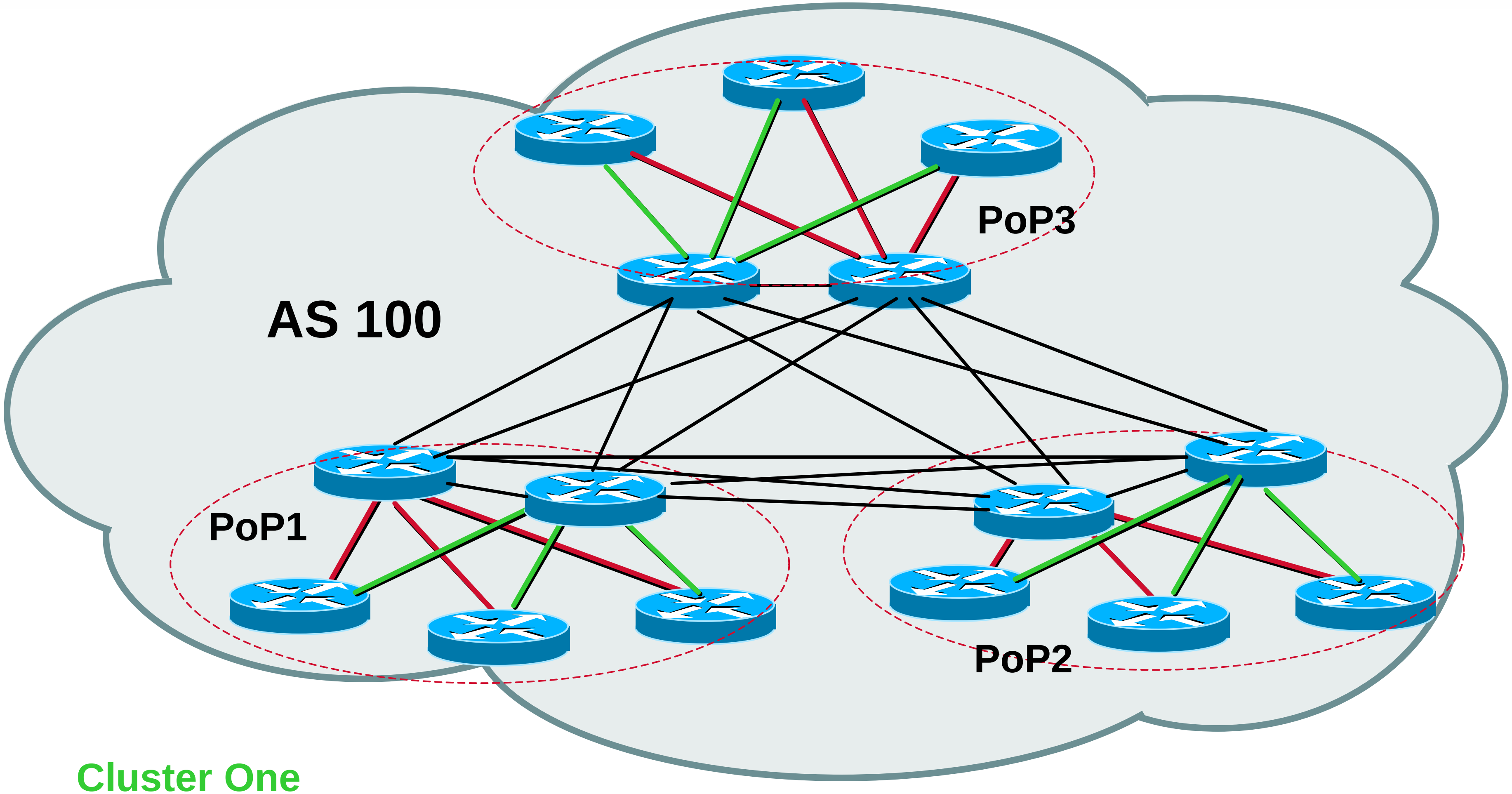
  All RRs in the cluster must have the same cluster-id (otherwise it is a different cluster)

- 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 Reflectors: Redundancy



Cluster One

Cluster Two

# Route Reflectors: Migration

- Where to place the route reflectors?

  Always follow the physical topology!

  This will guarantee that the packet forwarding won’t be affected

- Typical ISP network:

  PoP has two core routers

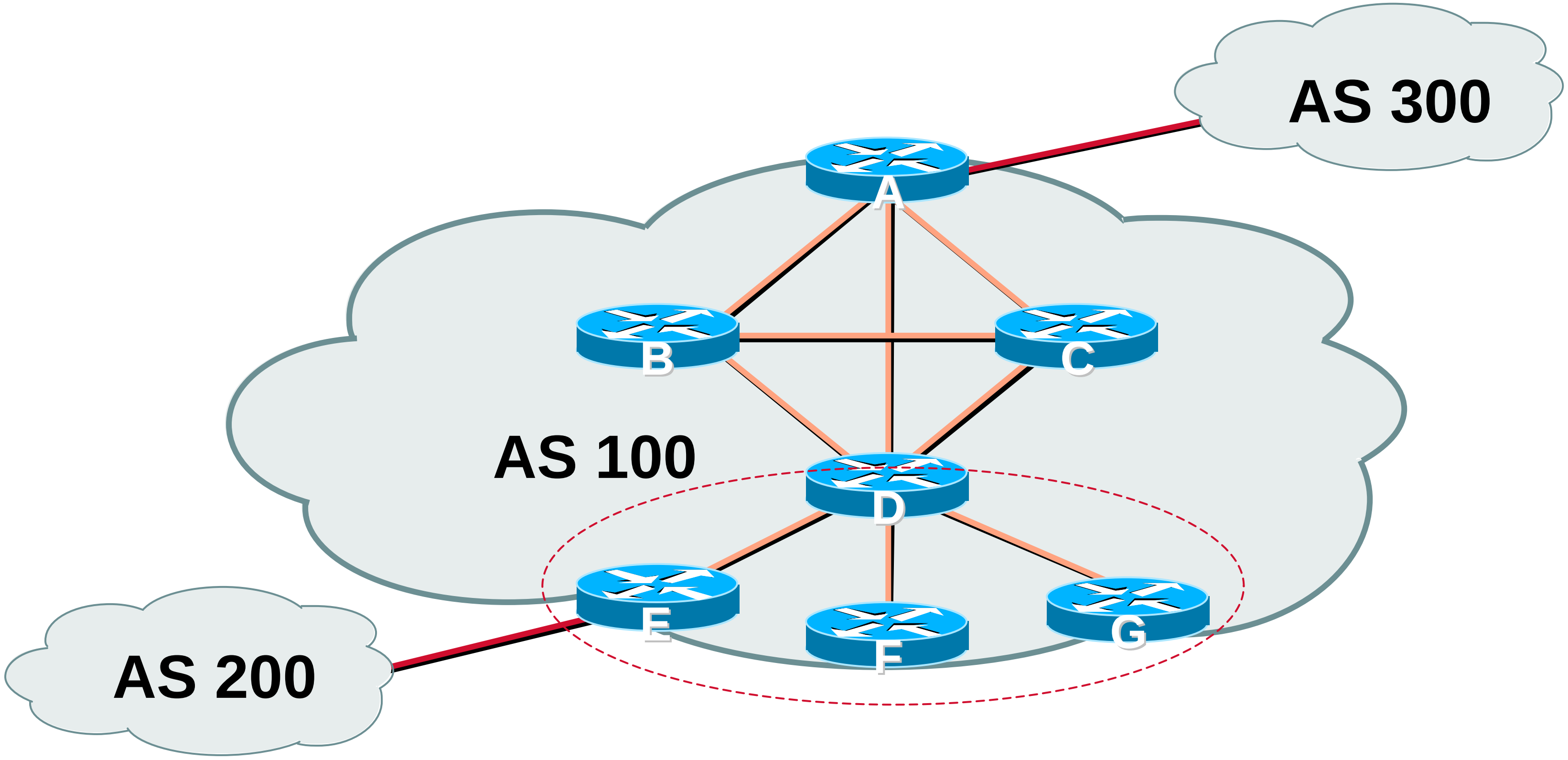
  Core routers are RR for the PoP

  Two overlaid clusters

# Route Reflectors: Migration

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

# Route Reflector: Migration



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

# 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
```

# 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 (Cont.)

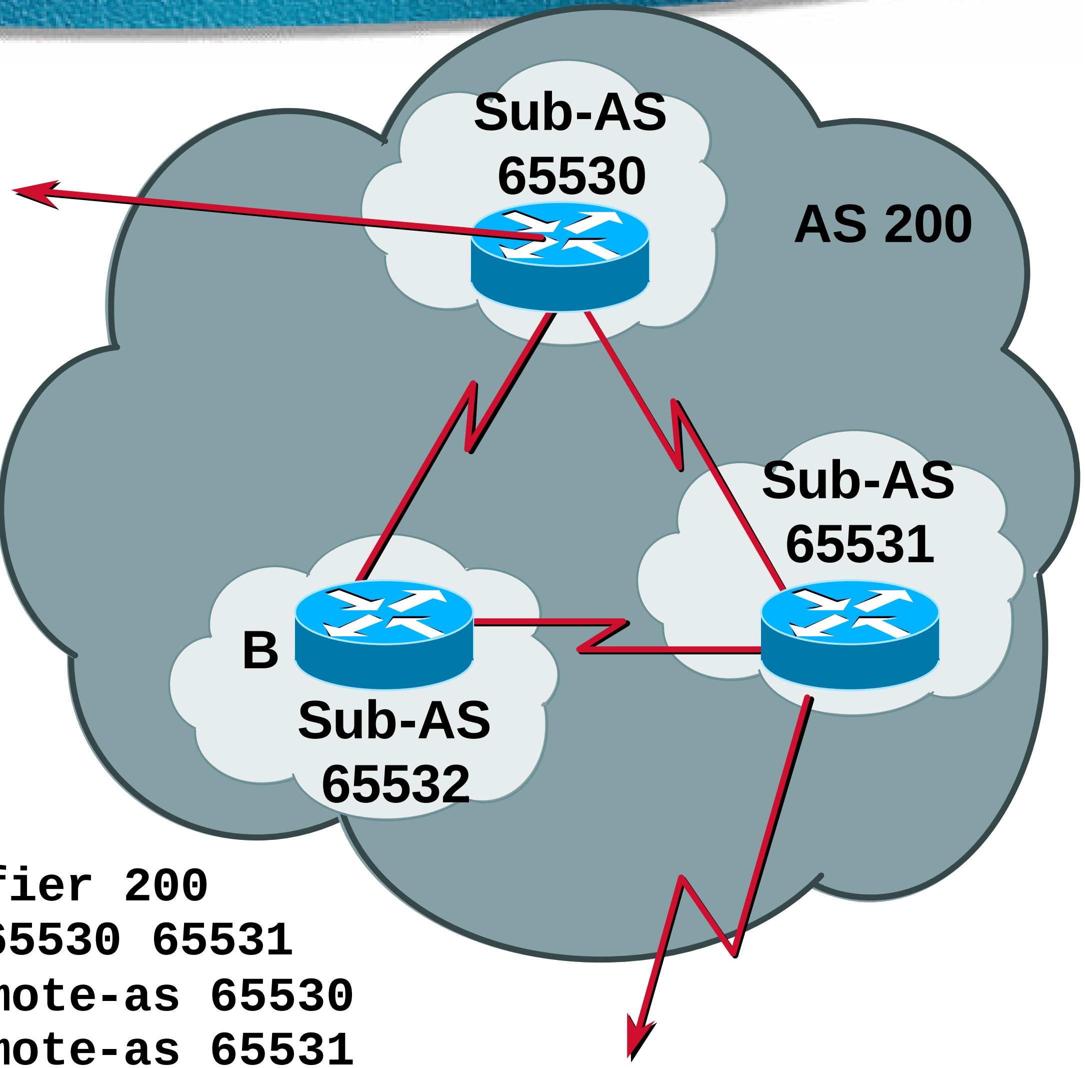
- Visible to outside world as single AS – “Confederation Identifier”

  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 (cont.)



- **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
```

# 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.)

- 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
```

# 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**

# 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

# BGP Scaling Techniques

- These 4 techniques should be core requirements in all ISP networks

  Soft reconfiguration/Route Refresh

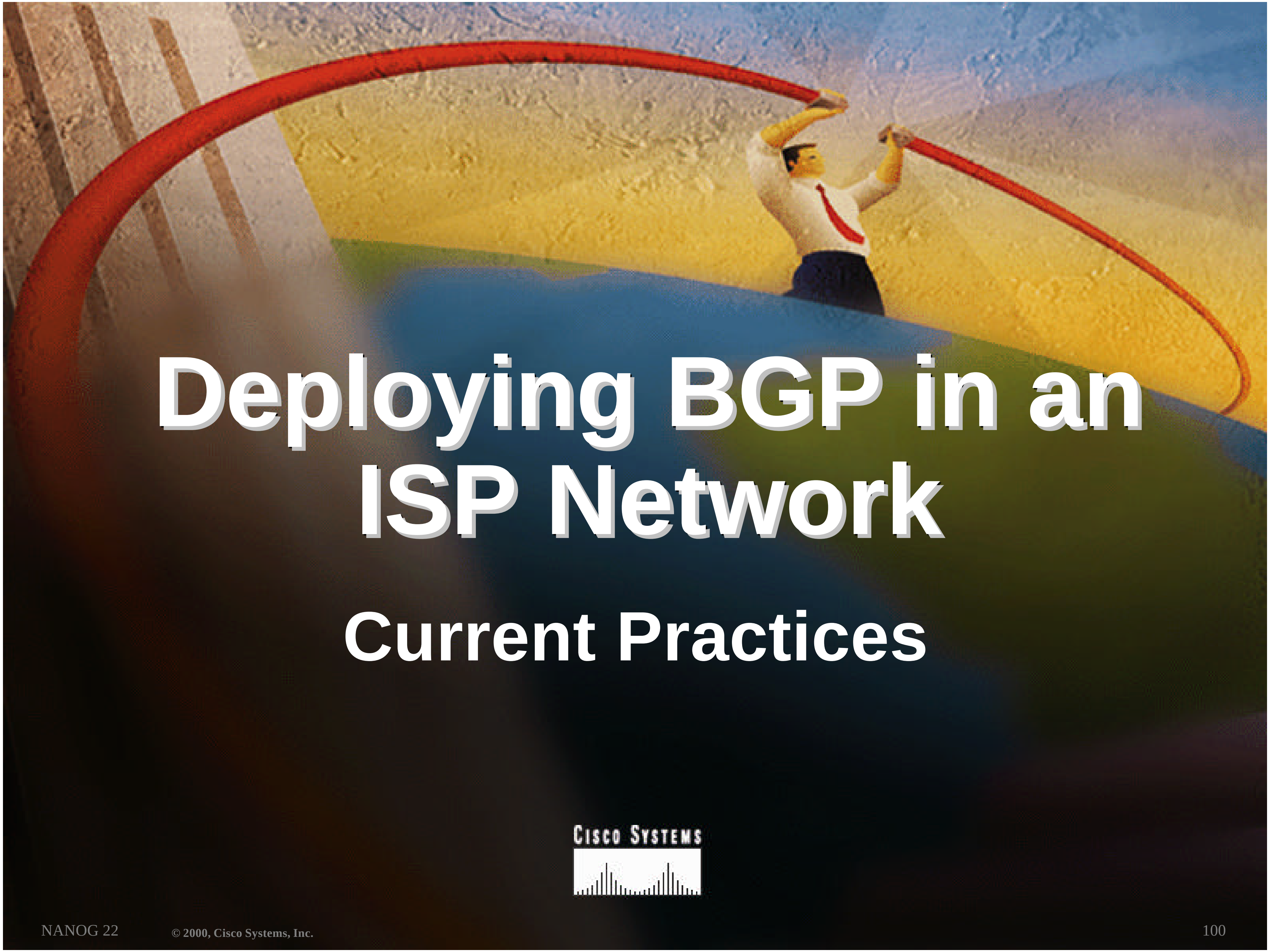
  Peer groups

  Route flap damping

  Route reflectors

# BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Deploying BGP in an ISP network
- Trouble & Troubleshooting
- Multihoming Examples
- Using Communities



# Deploying BGP in an ISP Network

## Current Practices

# BGP versus OSPF/ISIS

- Internal Routing Protocols (IGPs)

  examples are ISIS and OSPF

  used for carrying infrastructure addresses

  NOT used for carrying Internet prefixes or customer prefixes

  design goal is to minimise number of prefixes in IGP to aid scalability and rapid convergence

# BGP versus OSPF/ISIS

- BGP used internally (iBGP) and externally (eBGP)
- iBGP used to carry
    - some/all Internet prefixes across backbone
    - customer prefixes
- eBGP used to
    - exchange prefixes with other ASes
    - implement routing policy

# BGP versus OSPF/ISIS

- DO NOT:

  distribute BGP prefixes into an IGP

  distribute IGP routes into BGP

  use an IGP to carry customer prefixes

- YOUR NETWORK WILL NOT SCALE

# Aggregation

## Quality or Quantity?

# Aggregation

- ISPs receive address block from Regional Registry or upstream provider
- Aggregation means announcing the address block only, not subprefixes

  Subprefixes should only be announced in special cases – see later.

- Aggregate should be generated internally

  Not on the network borders!

# Configuring Aggregation – Method One

- ISP has 221.10.0.0/19 address block
- To put into BGP as an aggregate:

```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
ip route 221.10.0.0 255.255.224.0 null0
```

- The static route is a "pull up" route

  more specific prefixes within this address block ensure connectivity to ISP's customers

  "longest match lookup"

# Configuring Aggregation – Method Two

- Configuration Example

```
router bgp 109
 network 221.10.0.0 mask 255.255.252.0
 aggregate-address 221.10.0.0 255.255.224.0 [summary-only]
```

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

  ensures that only the summary is announced if a more specific prefix exists in the routing table
- Sets “aggregator” attribute

  Useful for debugging

# Announcing Aggregate – Cisco IOS

- Configuration Example

```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 101
 neighbor 222.222.10.1 prefix-list out-filter out
!
ip route 221.10.0.0 255.255.224.0 null0
!
ip prefix-list out-filter permit 221.10.0.0/19
```

# Announcing an Aggregate

- ISPs who don’t and won’t aggregate are held in poor regard by community
- Registries’ minimum allocation size is now a /20

  no real reason to see subprefixes of allocated blocks in the Internet

  BUT there are currently >60000 /24s!

# Receiving Prefixes

# Receiving Prefixes from downstream peers

- ISPs should only accept prefixes which have been assigned or allocated to their downstream peer
- For example

  downstream has 220.50.0.0/20 block

  should only announce this to peers

  peers should only accept this from them

# Receiving Prefixes – Cisco IOS

- Configuration Example on upstream

```
router bgp 100
 neighbor 222.222.10.1 remote-as 101
 neighbor 222.222.10.1 prefix-list customer in
!
ip prefix-list customer permit 220.50.0.0/20
```

# Receiving Prefixes from upstream peers

- Not desirable unless really necessary

  special circumstances – see later

- Ask upstream to either:

  originate a default-route

  announce one prefix you can use as default

# Receiving Prefixes from upstream peers

- Downstream Router Configuration

```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
 neighbor 221.5.7.1 remote-as 101
 neighbor 221.5.7.1 prefix-list infilter in
 neighbor 221.5.7.1 prefix-list outfilter out
!
ip prefix-list infilter permit 0.0.0.0/0
!
ip prefix-list outfilter permit 221.10.0.0/19
```

# Receiving Prefixes from upstream peers

- Upstream Router Configuration

```
router bgp 101
 neighbor 221.5.7.2 remote-as 100
 neighbor 221.5.7.2 default-originate
 neighbor 221.5.7.2 prefix-list cust-in in
 neighbor 221.5.7.2 prefix-list cust-out out
!
ip prefix-list cust-in permit 221.10.0.0/19
!
ip prefix-list cust-out permit 0.0.0.0/0
```

# Receiving Prefixes from upstream peers

- If necessary to receive prefixes from upstream provider, care is required

  don’t accept RFC1918 etc prefixes

  http://www.ietf.org/internet-drafts/draft-manning-dsua-06.txt

  don’t accept your own prefix

  don’t accept default (unless you need it)

  don’t accept prefixes longer than /24

# Receiving Prefixes

```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
 neighbor 221.5.7.1 remote-as 101
 neighbor 221.5.7.1 prefix-list in-filter in
!
ip prefix-list in-filter deny 0.0.0.0/0           ! Block default
ip prefix-list in-filter deny 0.0.0.0/8 le 32
ip prefix-list in-filter deny 10.0.0.0/8 le 32
ip prefix-list in-filter deny 127.0.0.0/8 le 32
ip prefix-list in-filter deny 169.254.0.0/16 le 32
ip prefix-list in-filter deny 172.16.0.0/12 le 32
ip prefix-list in-filter deny 192.0.2.0/24 le 32
ip prefix-list in-filter deny 192.168.0.0/16 le 32
ip prefix-list in-filter deny 221.10.0.0/19 le 32 ! Block local prefix
ip prefix-list in-filter deny 224.0.0.0/3 le 32   ! Block multicast
ip prefix-list in-filter deny 0.0.0.0/0 ge 25     ! Block prefixes >/24
ip prefix-list in-filter permit 0.0.0.0/0 le 32
```

# Prefixes into iBGP

# Injecting prefixes into iBGP

- Use iBGP to carry customer prefixes

  don’t ever use IGP

- Point static route to customer interface
- Use BGP network statement
- As long as static route exists (interface active), prefix will be in BGP

# Router Configuration network statement

- **Example:**

```
interface loopback 0
 ip address 215.17.3.1 255.255.255.255
!
interface Serial 5/0
 ip unnumbered loopback 0
 ip verify unicast reverse-path
!
ip route 215.34.10.0 255.255.252.0 Serial 5/0
!
router bgp 100
 network 215.34.10.0 mask 255.255.252.0
```

# Injecting prefixes into iBGP

- interface flap will result in prefix withdraw and re-announce

  use “ip route…permanent”

  Static route always exists, even if interface is down ® prefix announced in iBGP

- many ISPs use redistribute static rather than network statement

  only use this if you understand why

# Inserting prefixes into BGP – redistribute static

- Care required with redistribute!

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

  Does not scale if uncontrolled

  Best avoided if at all possible

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

# Router Configuration redistribute static

- Example:

```
ip route 215.34.10.0 255.255.252.0 Serial 5/0
!
router bgp 100
 redistribute static route-map static-to-bgp
<snip>
!
route-map static-to-bgp permit 10
 match ip address prefix-list ISP-block
 set origin igp
<snip>
!
ip prefix-list ISP-block permit 215.34.10.0/22 le 30
!
```

# Injecting prefixes into iBGP

- Route-map ISP-block can be used for many things:
  - setting communities and other attributes
  - setting origin code to IGP, etc
- Be careful with prefix-lists and route-maps
  - absence of either/both could mean all statically routed prefixes go into iBGP

# Configuration Tips

# iBGP and IGPs

- Make sure loopback is configured on router

  iBGP between loopbacks, NOT real interfaces

- Make sure IGP carries loopback /32 address

- Make sure IGP carries DMZ nets

  Or use next-hop-self on iBGP neighbours

  neighbor x.x.x.x next-hop-self

# Next-hop-self

- Used by many ISPs on edge routers

  Preferable to carrying DMZ /30 addresses in the IGP

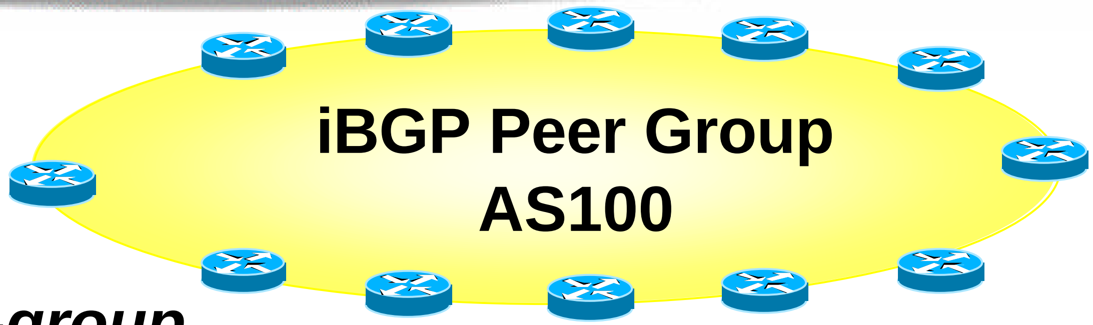
  Reduces size of IGP to just core infrastructure

  Alternative to using ip unnumbered

  Helps scale network

  BGP speaker announces external network using local address (loopback) as next-hop

# BGP Template – iBGP peers



```
router bgp  100
neighbor internal peer-group
neighbor internal description ibgp peers
neighbor internal remote-as 100
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
```

# BGP Template – iBGP peers

- Use peer-groups
- iBGP between loopbacks!
- Next-hop-self

  Keep DMZ and point-to-point out of IGP
- Always send communities in iBGP

  Otherwise accidents will happen
- Hardwire BGP to version 4

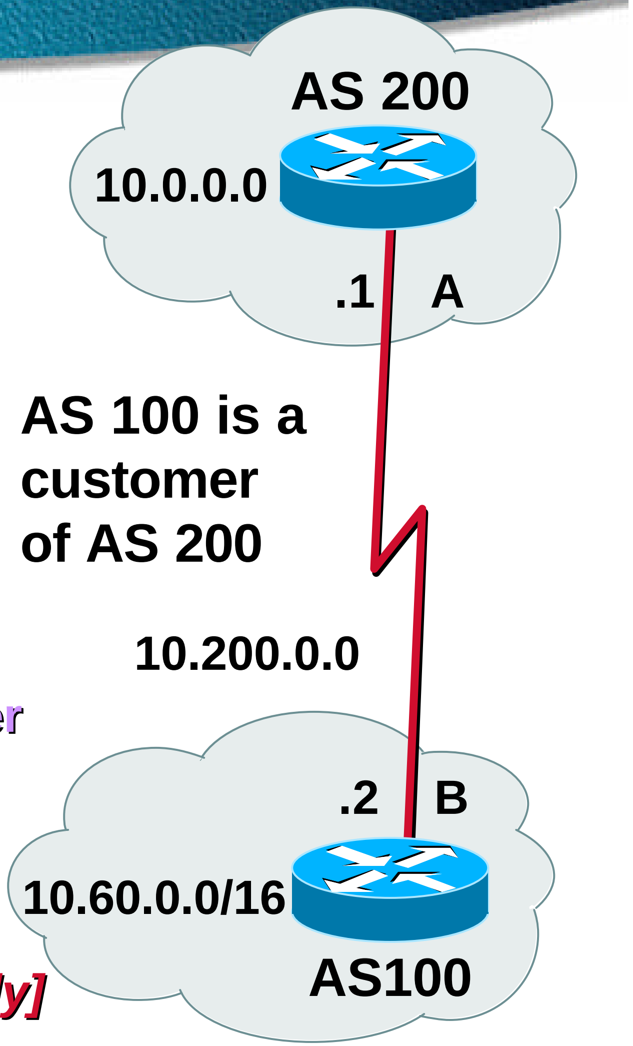
  Yes, this is being paranoid!
- Use passwords on iBGP session

  Not being paranoid, VERY necessary

# BGP Template – eBGP peers

**Router B:**

```
router bgp 100
bgp dampening route-map RIPE-210-flap
network 10.60.0.0 mask 255.255.0.0
neighbor external peer-group
neighbor external remote-as 200
neighbor external description ISP connection
neighbor external remove-private-AS
neighbor external version 4
neighbor external prefix-list ispout out ; "accident" filter
neighbor external route-map ispout out ; "real" filter
neighbor external route-map ispin in
neighbor external password 7 020A0559
neighbor external maximum-prefix 120000 [warning-only]
neighbor 10.200.0.1 peer-group external
ip route 10.60.0.0 255.255.0.0 null0 254
```



AS 100 is a customer of AS 200

# BGP Template – eBGP peers

- BGP damping – use RIPE-210 parameters
- Remove private ASes from announcements

  Common omission today
- Use extensive filters, with “backup”
- Use password agreed between you and peer on eBGP session
- Use maximum-prefix tracking

  Router will warn you if there are sudden changes in BGP table size, bringing down eBGP if necessary

# More BGP “defaults”

- Log neighbour changes

  ```
  bgp log-neighbor-changes
  ```

- Enable deterministic MED

  ```
  bgp deterministic-med
  ```

  Otherwise bestpath could be different every time BGP session is reset

- Make BGP admin distance higher than any IGP

  ```
  distance bgp 200 200 200
  ```

# Customer Aggregation

- BGP customers

  Offer max 3 types of feeds (easier than custom configuration per peer)

  Use communities

- Static customers

  Use communities

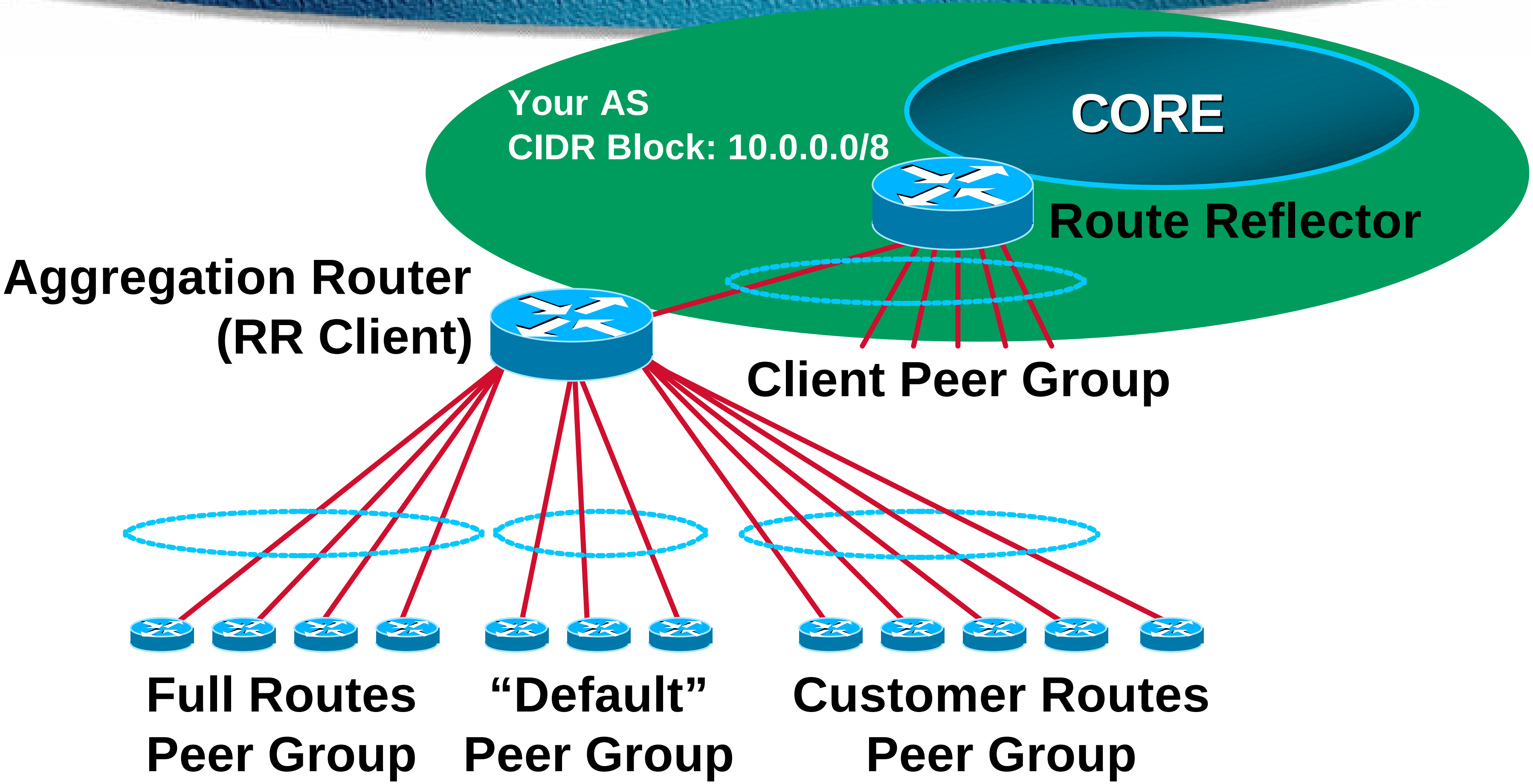
- Differentiate between different types of prefixes

  Makes eBGP filtering easy

# BGP Customer Aggregation Guidelines

- Define at least three peer groups:
  - cust-default—send default route only
  - cust-cust—send customer routes only
  - cust-full —send full Internet routes
- Identify routes via communities e.g.
  - 100:4100=customers; 100:4500=peers
- Apply passwords per neighbour
- Apply inbound & outbound prefix-list per neighbour

# BGP Customer Aggregation



**Apply passwords and in/outbound prefix-list directly to each neighbour**

# Static Customer Aggregation Guidelines

- Identify routes via communities, e.g.

  100:4000=my address blocks

  100:4200=customers from my block

  100:4300=customers outside my block

  Helps with aggregation, iBGP, filtering

- BGP network statements on aggregation routers set correct community

# Sample core configuration

- eBGP peers and upstreams

  Send communities 100:4000, 100:4100 and 100:4300, receive everything

- iBGP full routes

  Send everything (only network core)

- iBGP partial routes

  Send communities 100:4000, 100:4100, 100:4200, 100:4300 and 100:4500 (edge routers, peering routers, IXP routers)

- Simple configuration with peer-groups and route-maps

# Acquisitions!

- Your ISP has just bought another ISP

  How to merge networks?

- Options:

  use confederations – make their AS a sub-AS (only useful if you are using confederations already)

  use the BGP local-as feature to implement a gradual transition – overrides BGP process ID

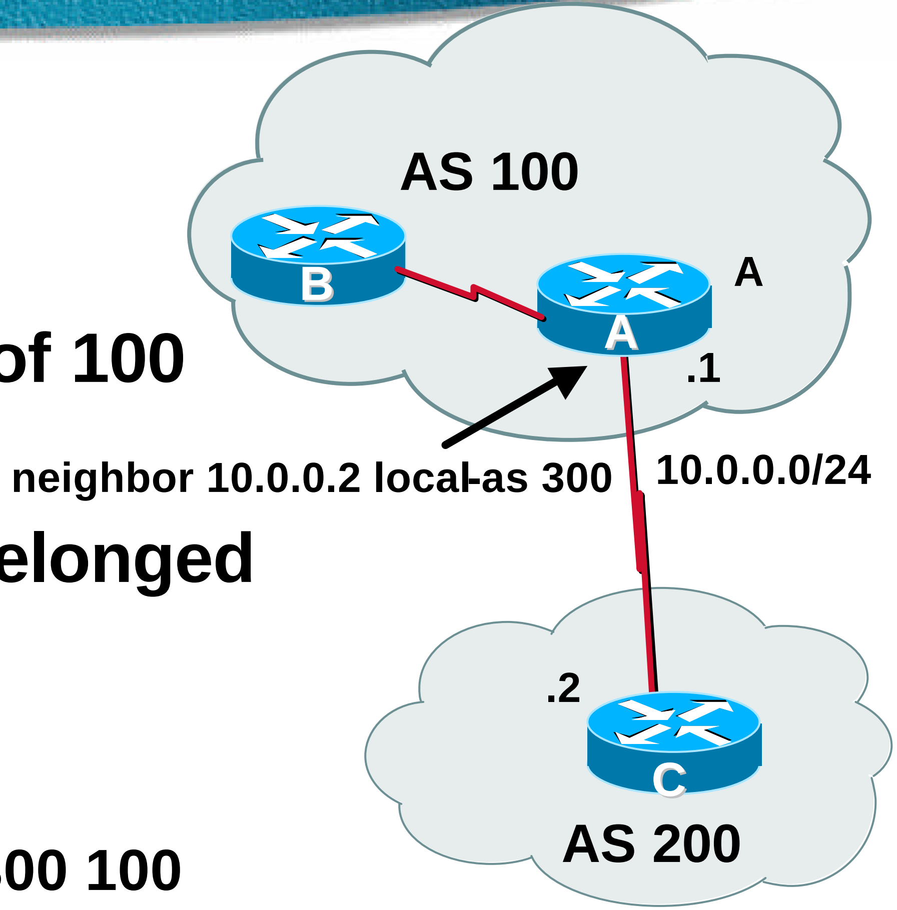
  neighbor *x.x.x.x* local-as *as-number*

# local-AS – Application

- **Router A has a process ID of 100**
- **The peering with AS200 is established as if router A belonged to AS300.**
- **AS_PATH**

  **routes originated in AS100 = 300 100**

  **routes received from AS200 = 300 200**



```
neighbor 10.0.0.2 local-as 300
```

# BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Deploying BGP in an ISP network
- Trouble & Troubleshooting
- Multihoming Examples
- Using Communities

# Troubleshooting

## Staying out of Trouble

Cisco Systems

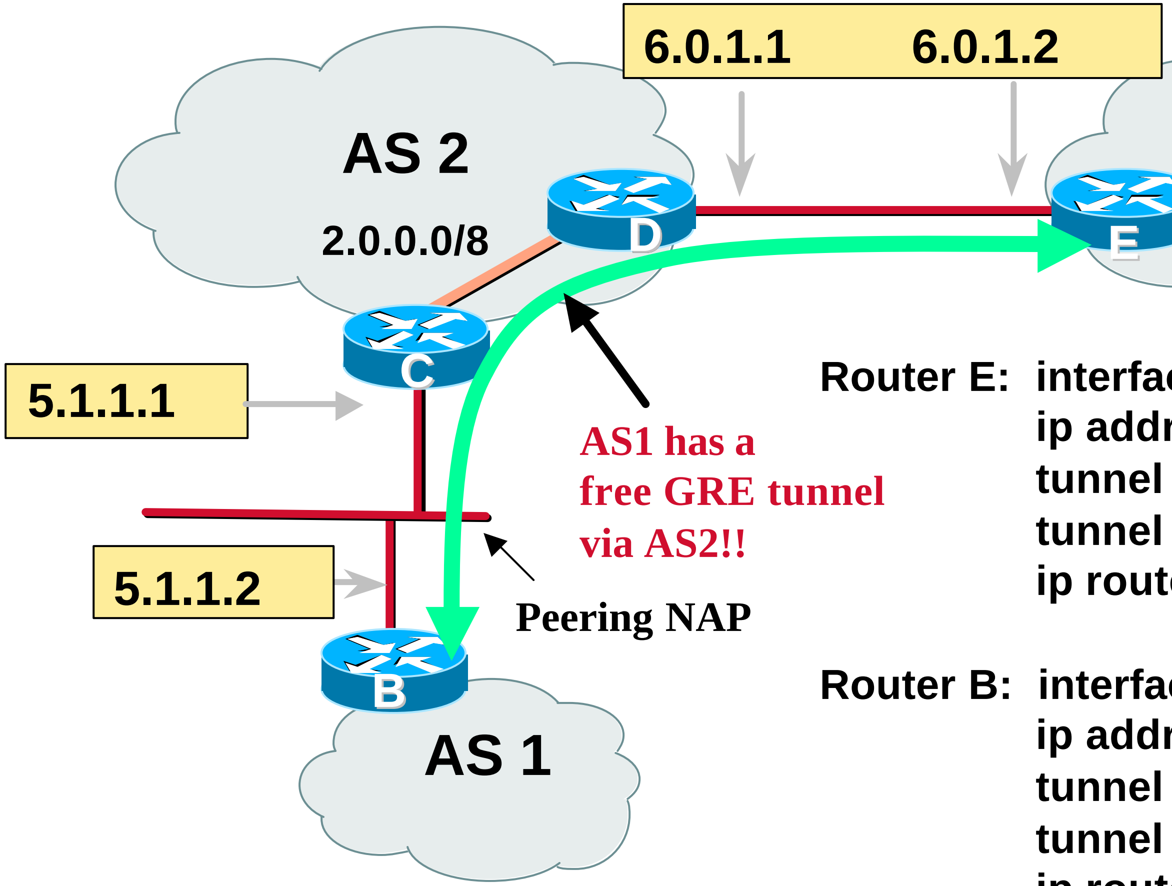
# Potential Caveats and Operational Problems

- GRE Tunnels & IXPs
- Auto-summarisation & synchronisation
- Route Reflectors

  Follow the topology
- Common Problems

  …and the solutions!

# Prevent GRE VPNs



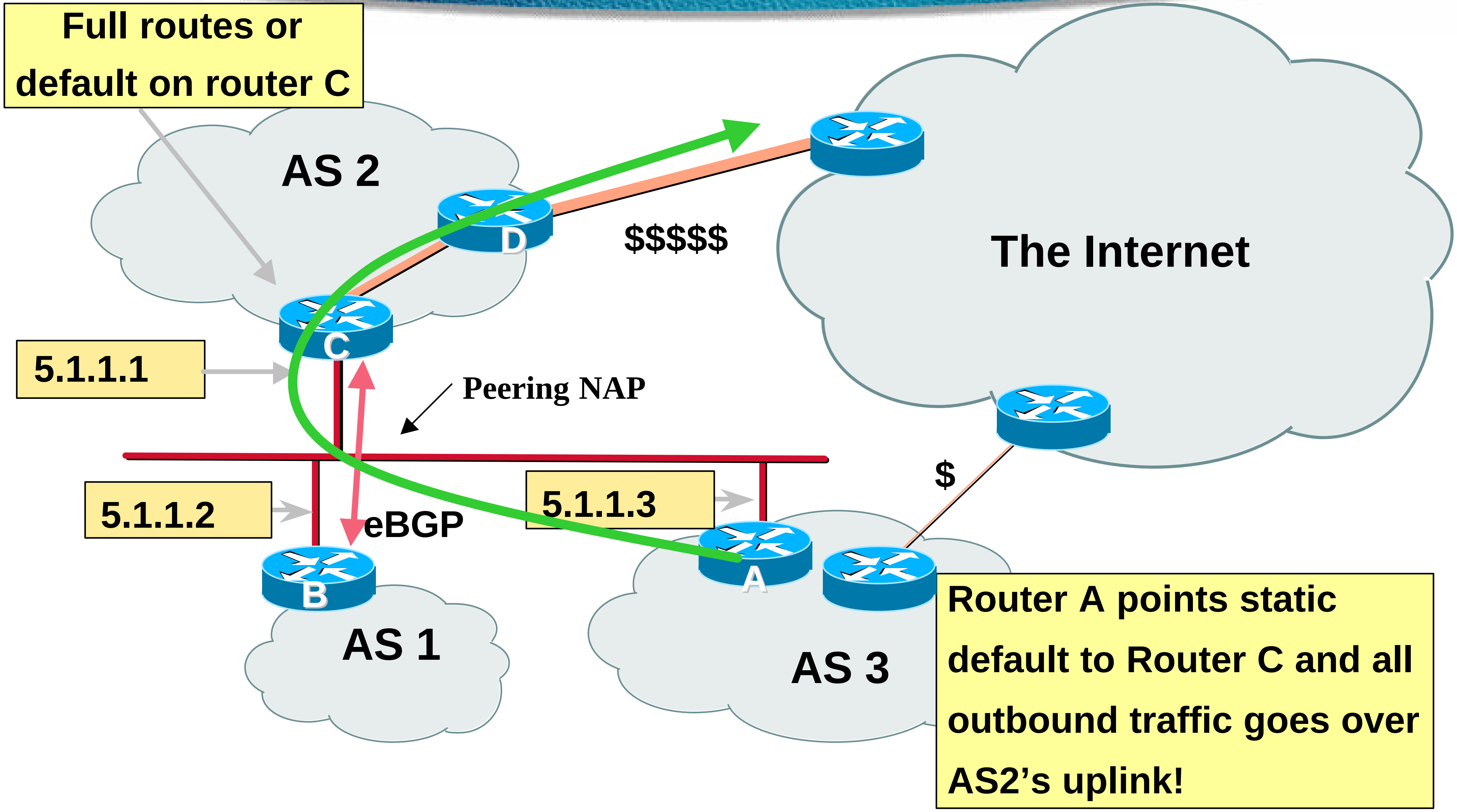
AS1 has a free GRE tunnel via AS2!!

```
Router E:  interface tunnel 0
           ip address 6.0.0.1 255.255.255.252
           tunnel source 6.0.1.2
           tunnel destination 5.1.1.2
           ip route 5.1.1.2 255.255.255.255 6.0.1.1

Router B:  interface tunnel 0
           ip address 6.0.0.2 255.55.255.252
           tunnel source 5.1.1.2
           tunnel destination 6.0.1.2
           ip route 6.0.1.2 255.255.255.255 5.1.1.1
```

**Don’t carry IXP net in your IGP – use next-hop-self!**

# Prevent "Defaulting"



Full routes or default on router C

5.1.1.1

5.1.1.2

5.1.1.3

Router A points static default to Router C and all outbound traffic goes over AS2's uplink!

# Watch out at IXPs/NAPs

- IXP router should not carry full routes or have a default
- ISP should not carry IXP/NAP network prefix internally

  Use BGP next-hop-self

- or -

- Use RPF check for non-peers
- Use good filters for peers

# Auto Summarisation – Cisco IOS

- Historical feature
- Automatically summarises subprefixes to the classful network for prefixes redistributed into BGP

  Example:

  ```
    61.10.8.0/22 --> 61.0.0.0/8
  ```
- Must be turned off for any Internet connected site using BGP.

  ```
  router bgp 109
   no auto-summary
  ```

# Synchronisation – Cisco IOS

- **Historical feature**
- **BGP will 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 109
 no synchronization
```

# Troubleshooting

## Common Problems and their Solutions

# Troubleshooting – Examples

- Missing routes
- Route Oscillation
- Routing Loops
- Troubleshooting hints

# Route Origination

- Network statement with mask

```
R1# show run | begin bgp
 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

- Nail down routes you want to originate

```
R1#ip route 200.200.0.0 255.255.252.0 Null 0  200
```

- 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
*> 200.200.0.0/22    0.0.0.0          0    32768
```

# Route Oscillation

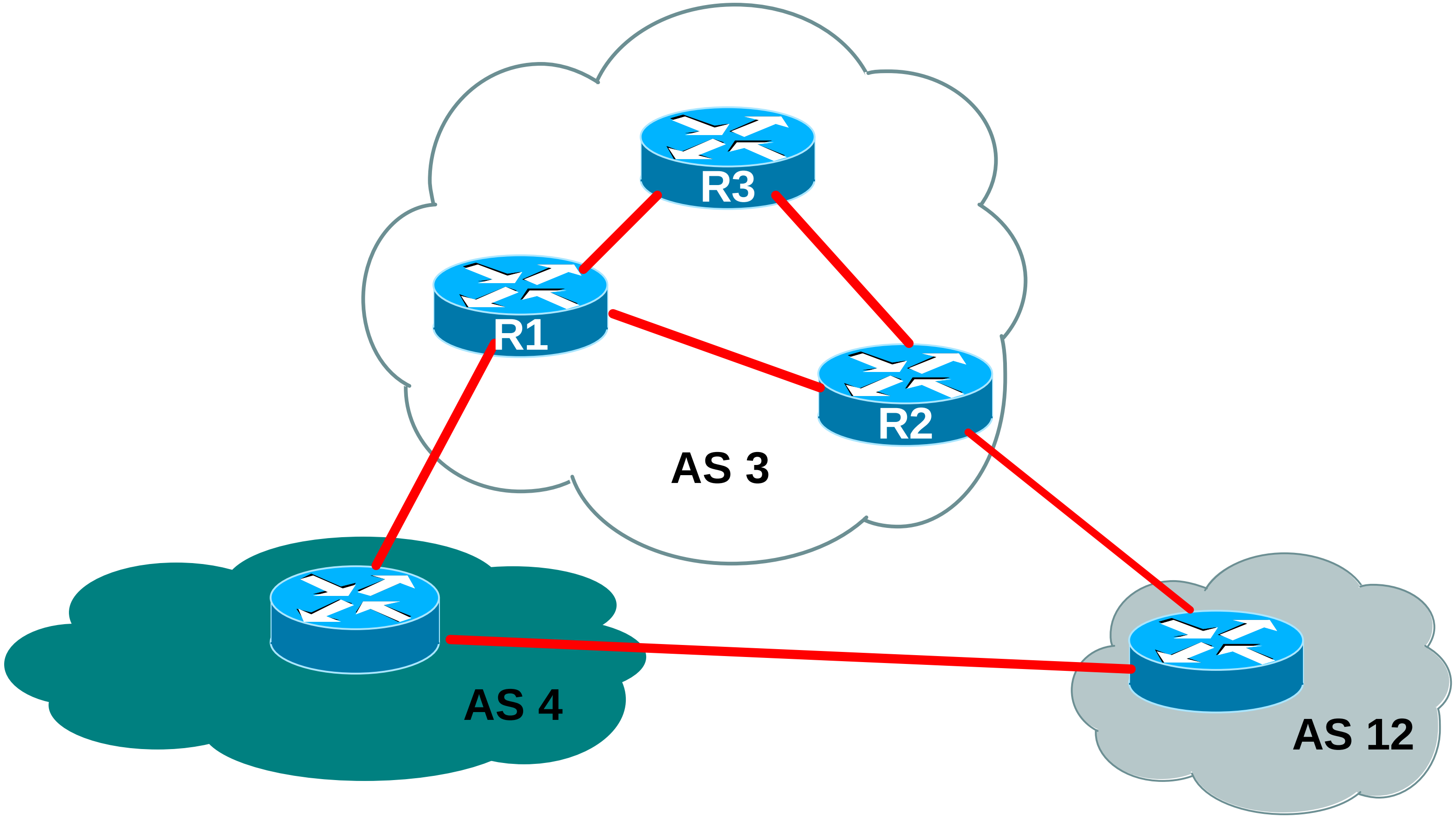
- One of the most common problems!

  Every minute routes flap in the routing table from one next hop to another

  With large routing table the most obvious symptom is high CPU in the “BGP-Router” process

  Can be frustrating to track down unless you have seen it before!

# Route Oscillation – Diagram



# 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 up a bgp route from  the RIB that is less than a minute old and watch what happens with the routing/bgp table …**

```
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

## …and after bgp_scanner runs (by default once a minute):

```
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

## Let’s take a look at the next hop at this point!

```
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

**Next-hop is recursive !!!**

**This will be detected next time the scanner runs and the other path will be installed in the RIB instead**

```
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

The route to the next-hop is now valid and at the next bgp scan we will change to the shorter as-path path, and so on …

```
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 – Summary

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

  use "next-hop-self" for iBGP peering

  -or-

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

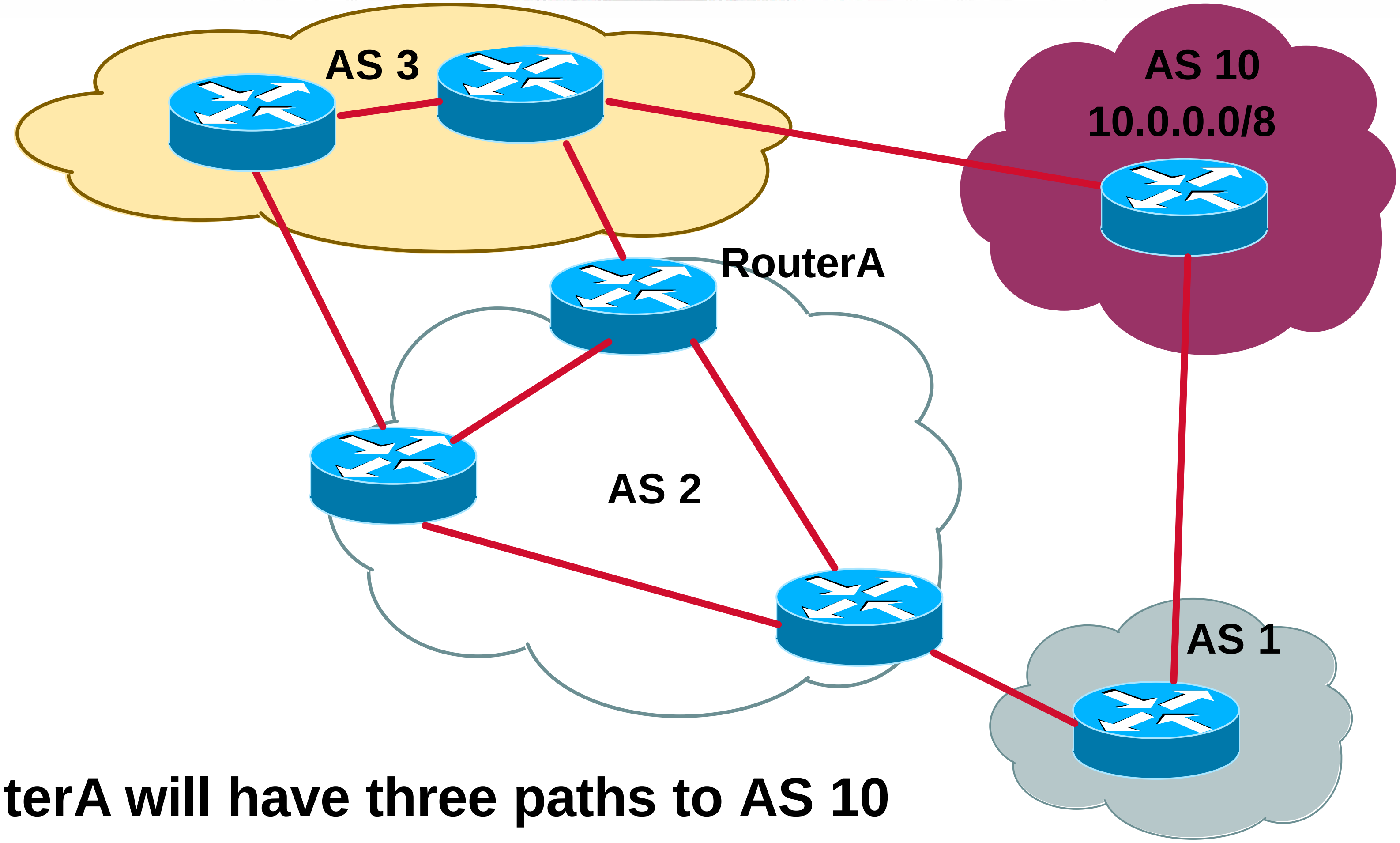
# Inconsistent Route Selection

- 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 in RFC1771

# Inconsistent Route Selection

- RFC says that MED is not always compared
- As a result, the ordering of the paths can affect 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

# Symptom – Diagram



- RouterA will have three paths to AS 10
- MEDs from AS 3 will not be compared with MEDs from AS 1

# Inconsistent Route Selection

```
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 Route Selection

```
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 Neighbour AS**
- **The bestpath for each Neighbour 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**

# Deterministic MED – Summary

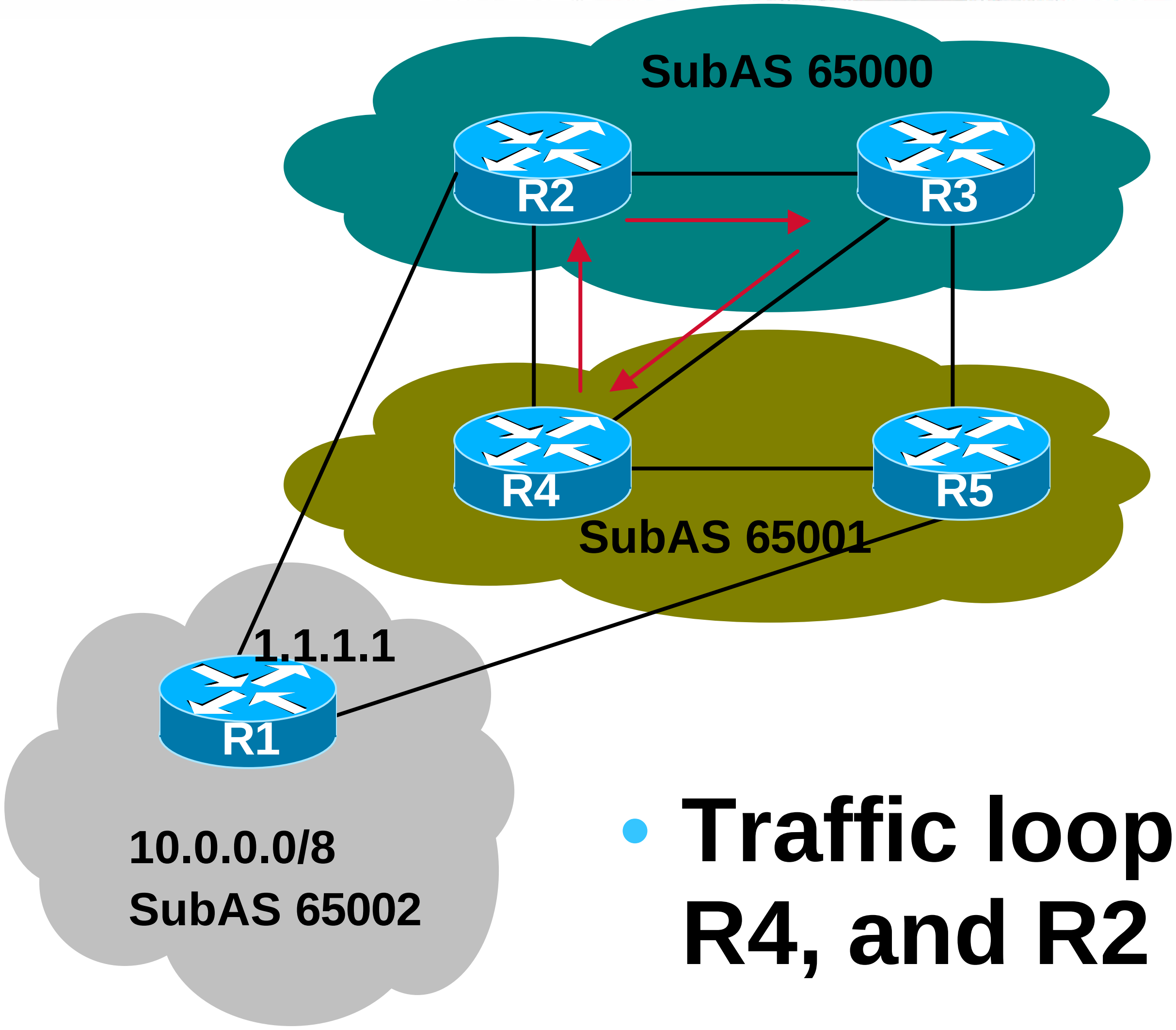
- If multihoming with multiple ISPs and peering with one ISP at multiple points:

  use “bgp deterministic-med”

  enable it on all routers in the AS

- Always use “bgp deterministic-med”

# Routing Loop – Problem



```
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 – Diagnosis

- First grab 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 – Diagnosis

- 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 – Diagnosis

- 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 – Diagnosis

- 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 5010) 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 NEXT_HOP
- R3 is advertising the path to us with a NEXT_HOP of 3.3.3.3 ???

# Routing Loop – Diagnosis

- **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 | include maximum
 maximum-paths 6
```

# Routing Loop – Solution

- “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 BGP NEXT_HOPs
- BGP loops are normally caused by
  - Not following physical topology in RR environment
  - Multipath within 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

# 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 in”

  Ask x.x.x.x to resend his UPDATEs to us

- “clear ip bgp x.x.x.x 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

# Summary/Tips

- Isolate the problem!!
- Use ACLs when enabling debug commands
- Enable bgp log-neighbor-changes
- IP reachability must exist for sessions to be established

  learned from IGP

  make sure the source and destination addresses match the configuration

# BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Deploying BGP in an ISP network
- Trouble & Troubleshooting
- Multihoming Examples
- Using Communities

# Multihoming

# Multihoming Definition

- More than one link external to the local network
  - two or more links to the same ISP
  - two or more links to different ISPs
- Usually two external facing routers
  - one router gives link and provider redundancy only

# AS Numbers

- An Autonomous System Number is required by BGP
- Obtained from upstream ISP or Regional Registry
- Necessary when you have links to more than one ISP or exchange point

# Configuring Policy

- Three BASIC Principles

  prefix-lists to filter prefixes

  filter-lists to filter ASNs

  route-maps to apply policy

- Avoids confusion!

# Originating Prefixes

- Basic Assumptions

  MUST announce assigned address block to Internet

  MAY also announce subprefixes – reachability is not guaranteed

  RIR minimum allocation is /20

    several ISPs filter RIR blocks on this boundary

    called “Net Police” by some

# Part of the “Net Police” prefix list

```
!! APNIC
ip prefix-list FILTER permit 61.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 202.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 210.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 218.0.0.0/8 ge 9 le 20
!! ARIN
ip prefix-list FILTER permit 63.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 64.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 66.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 199.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 200.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 204.0.0.0/6 ge 9 le 20
ip prefix-list FILTER permit 208.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 216.0.0.0/8 ge 9 le 20
!! RIPE NCC
ip prefix-list FILTER permit 62.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 80.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 193.0.0.0/8 ge 9 le 20
ip prefix-list FILTER permit 194.0.0.0/7 ge 9 le 20
ip prefix-list FILTER permit 212.0.0.0/7 ge 9 le 20
```

# “Net Police” prefix list issues

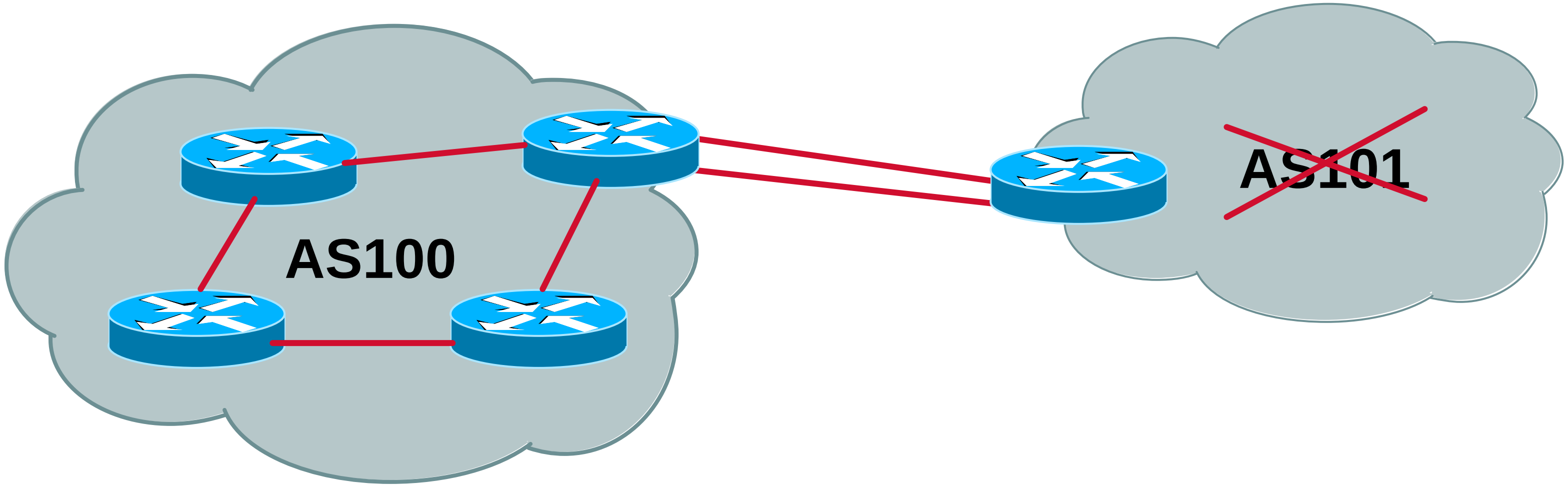
- meant to “punish” ISPs who won’t and don’t aggregate
- impacts legitimate multihoming
- impacts regions where domestic backbone is unavailable or costs $$$ compared with international bandwidth
- hard to maintain – requires updating when RIRs start allocating from new address blocks
- don’t do it unless consequences understood and you are prepared to keep it current

# Multihoming Options

# Multihoming Scenarios

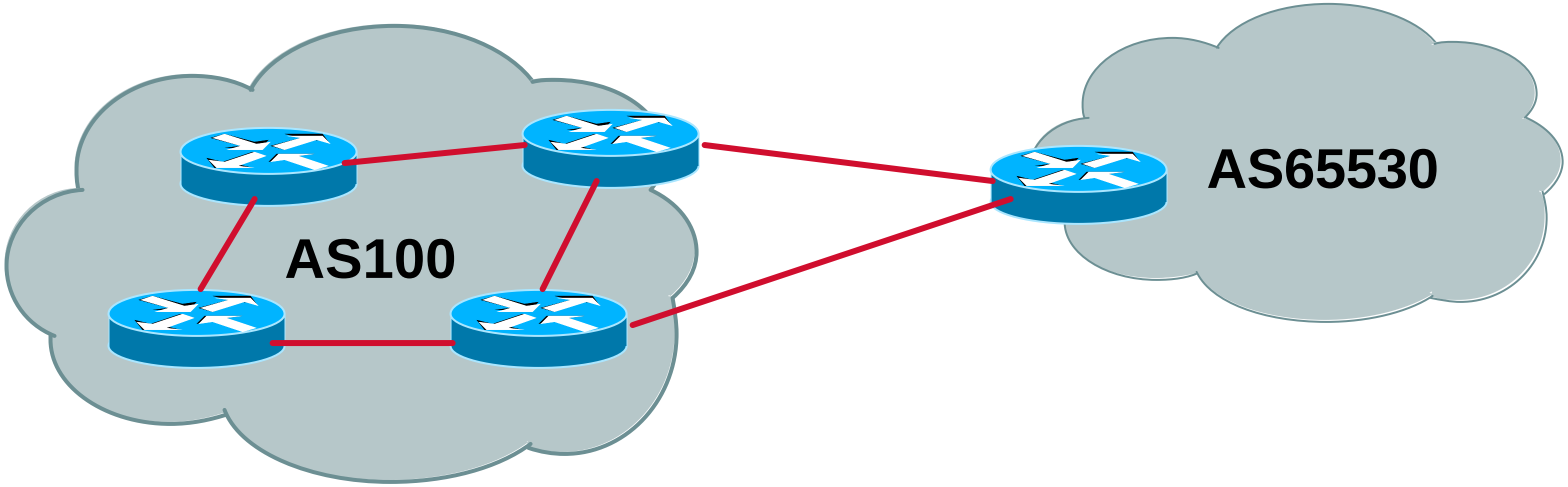
- Stub network
- Multi-homed stub network
- Multi-homed network
- Configuration Options

# Stub Network



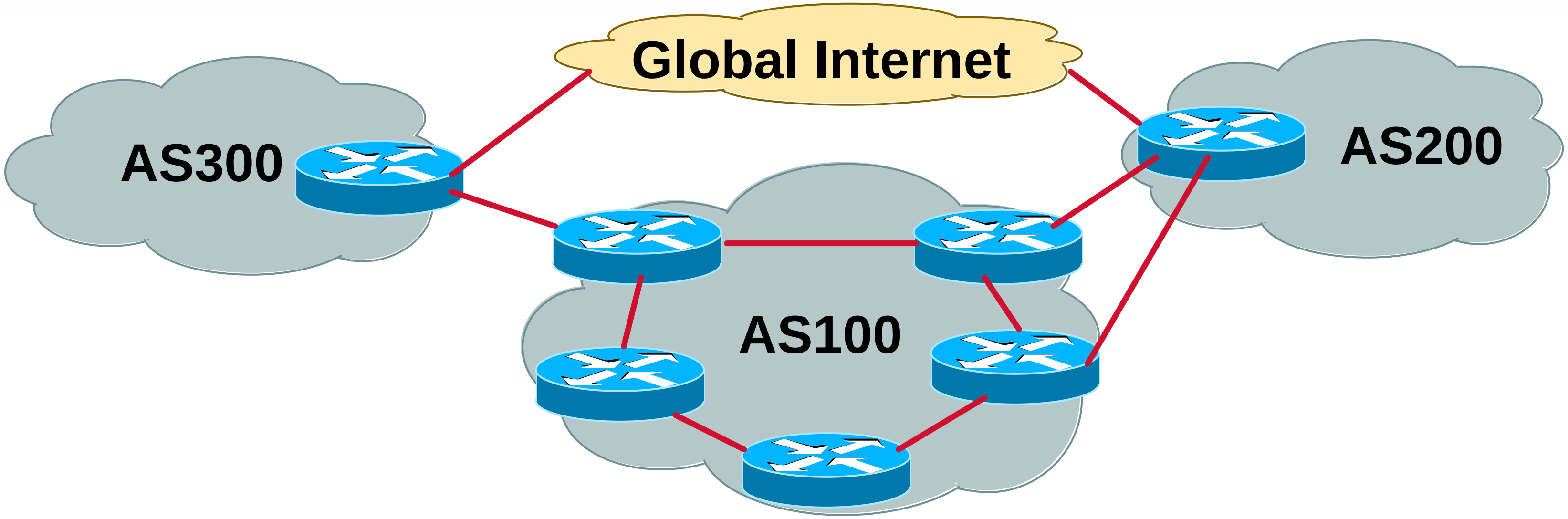
- No need for BGP
- Point static default to upstream ISP
- Upstream ISP advertises stub network
- Policy confined within upstream ISP’s policy

# Multi-homed Stub Network



- Use BGP (not IGP or static) to loadshare
- Use private AS (ASN > 64511)
- Upstream ISP advertises stub network
- Policy confined within upstream ISP's policy

# Multi-Homed Network



- **Many situations possible**

  **multiple sessions to same ISP**

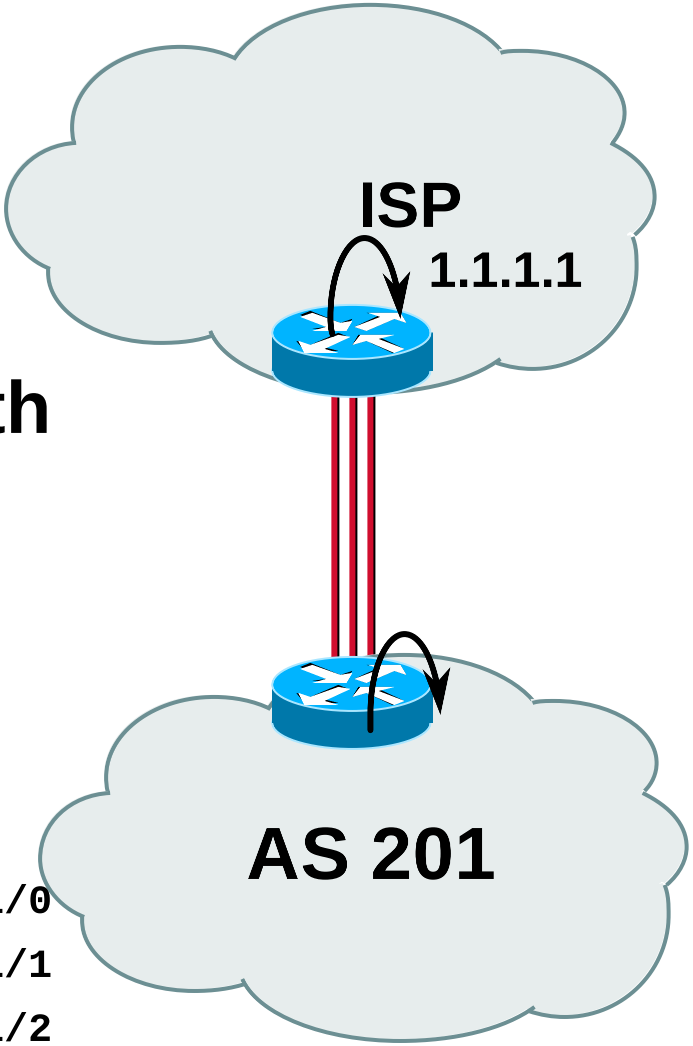
  **secondary for backup only**

  **load-share between primary and secondary**

  **selectively use different ISPs**

# Multiple Sessions to an ISP – Example One

- eBGP multihop
- eBGP to loopback addresses
- eBGP prefixes learned with loopback address as next hop

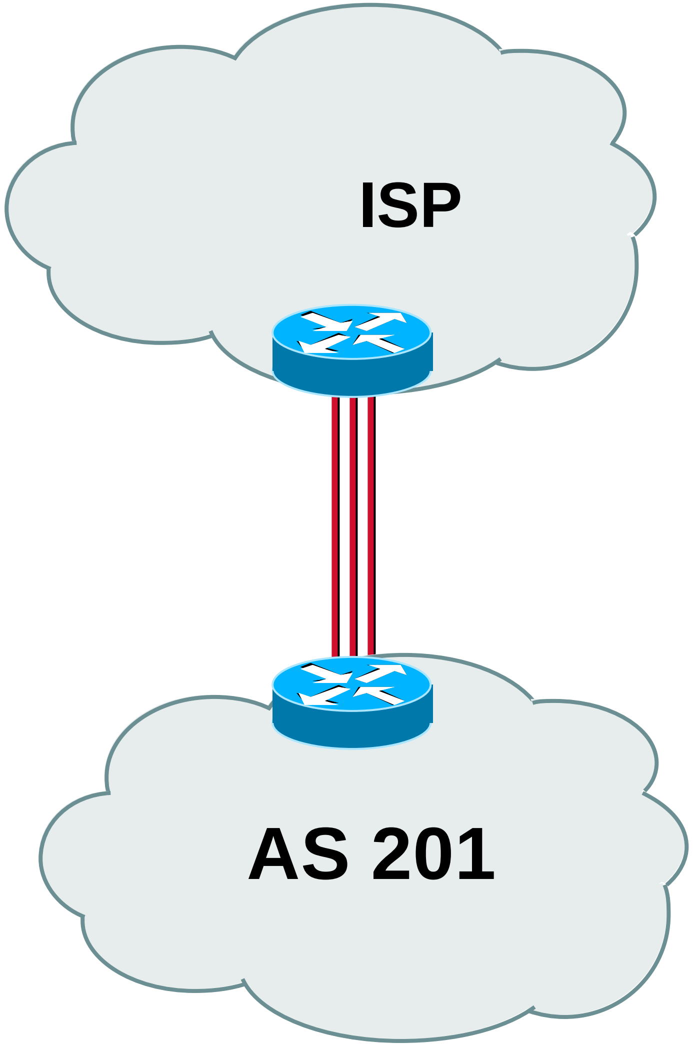


```
router bgp 201
 neighbor 1.1.1.1 remote-as 200
 neighbor 1.1.1.1 ebgp-multihop 5
ip route 1.1.1.1 255.255.255.255 serial 1/0
ip route 1.1.1.1 255.255.255.255 serial 1/1
ip route 1.1.1.1 255.255.255.255 serial 1/2
```

# Multiple Sessions to an ISP – Example Two

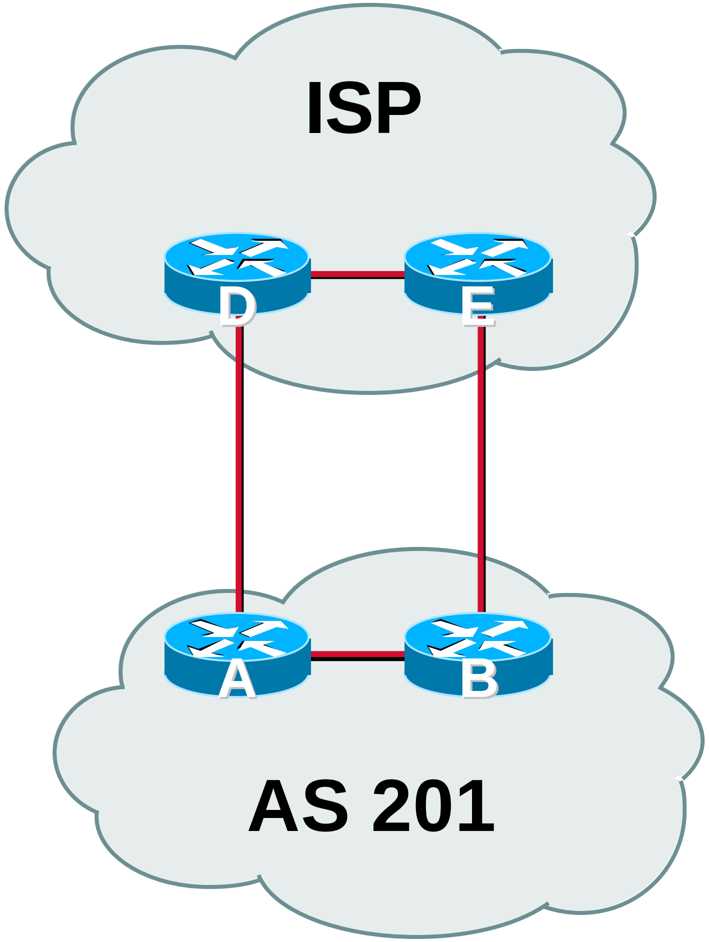
- BGP multi-path
- Three BGP sessions required
- limit of 6 parallel paths

```
router bgp 201
 neighbor 1.1.2.1 remote-as 200
 neighbor 1.1.2.5 remote-as 200
 neighbor 1.1.2.9 remote-as 200
 maximum-paths 3
```



# Multiple Sessions to an ISP

- Simplest scheme is to use defaults
- Learn/advertise prefixes for better control
- Planning and some work required to achieve loadsharing
- No magic solution

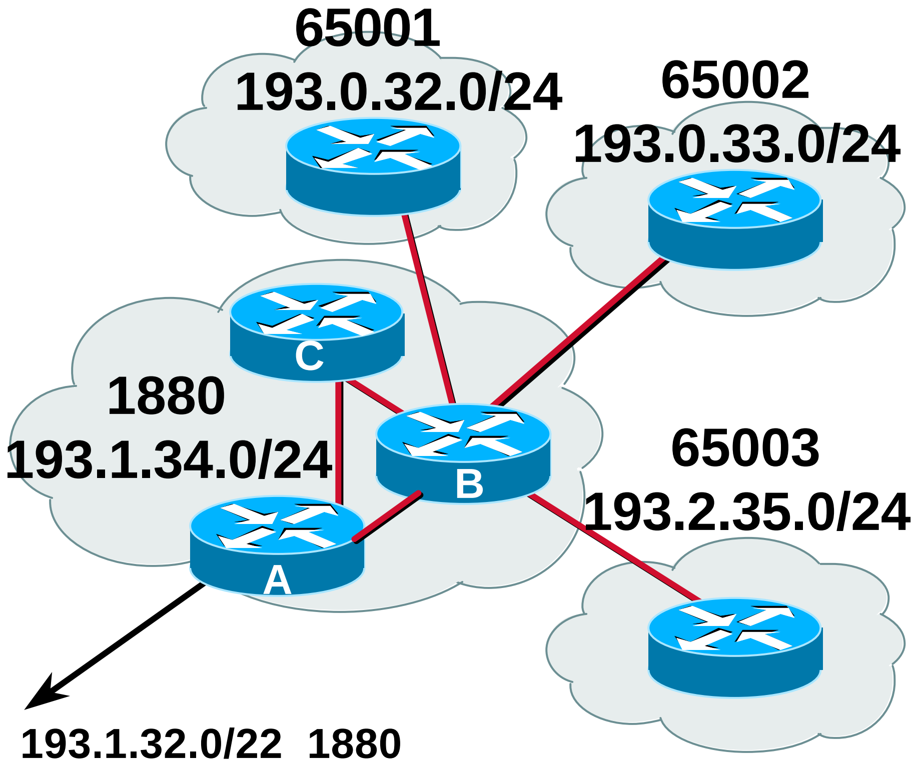


# Private-AS – Application

- Applications

  ISP with single-homed customers (RFC2270)

  corporate network with several regions and connections to the Internet only in the core



# Private-AS Removal

- neighbor x.x.x.x remove-private-AS
- Rules:

  available for eBGP neighbors only

  if the update has AS_PATH made up of private-AS numbers, the private-AS will be dropped

  if the AS_PATH includes private and public AS numbers, private AS number will not be removed…it is a configuration error!

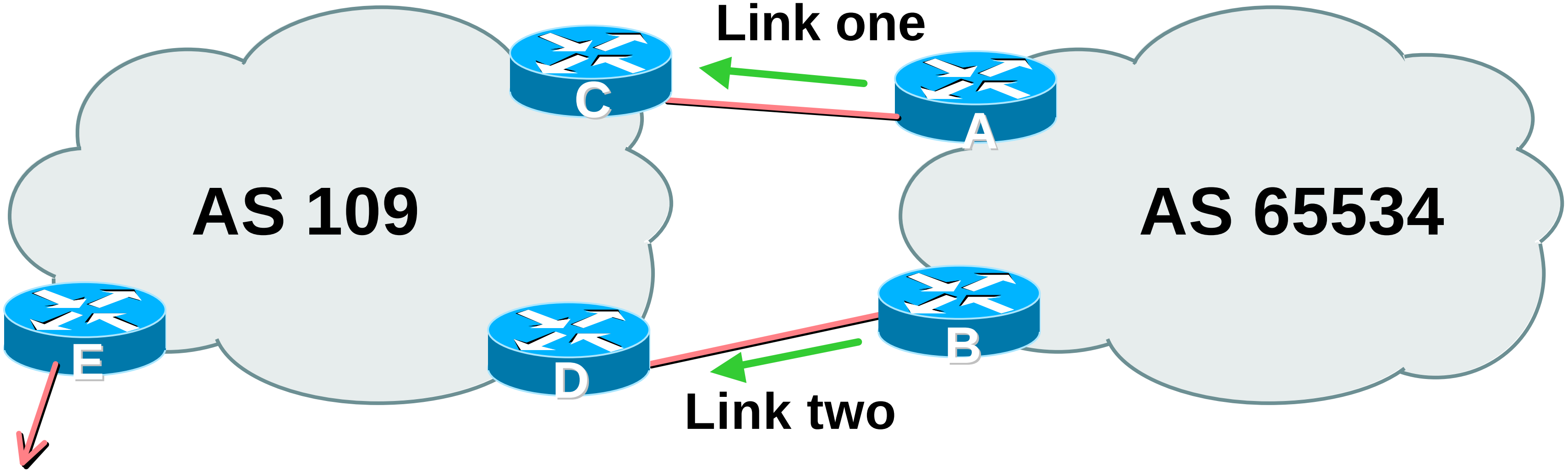
  if AS_PATH contains the AS number of the eBGP neighbor, the private-AS numbers will not be removed

  if used with confederations, it will work as long as the private AS numbers are after the confederation portion of the AS_PATH

# Two links to the same ISP

## With Redundancy and Loadsharing

# Two links to the same ISP (with redundancy)



- **AS109 removes private AS and any customer subprefixes from Internet announcement**

# Loadsharing to the same ISP

- Announce /19 aggregate on each link
- Split /19 and announce as two /20s, one on each link

  basic inbound loadsharing

  assumes equal circuit capacity and even spread of traffic across address block
- Vary the split until “perfect” loadsharing achieved
- Accept the default from upstream

  basic outbound loadsharing by nearest exit

  okay in first approx as most ISP and end-site traffic is inbound

# Two links to the same ISP

- **Router A Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.0.0 mask 255.255.240.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 prefix-list routerC out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerC permit 221.10.0.0/20
ip prefix-list routerC permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.240.0 null0
ip route 221.10.0.0 255.255.224.0 null0
```

**Router B configuration is similar but with the other /20**

# Two links to the same ISP

- Router C Configuration

```
router bgp 109
 neighbor 222.222.10.1 remote-as 65534
 neighbor 222.222.10.1 default-originate
 neighbor 222.222.10.1 prefix-list Customer in
 neighbor 222.222.10.1 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- Router C only allows in /19 and /20 prefixes from customer block
- Router D configuration is identical

# Loadsharing to the same ISP

- Loadsharing configuration is only on customer router
- Upstream ISP has to

  remove customer subprefixes from external announcements

  remove private AS from external announcements

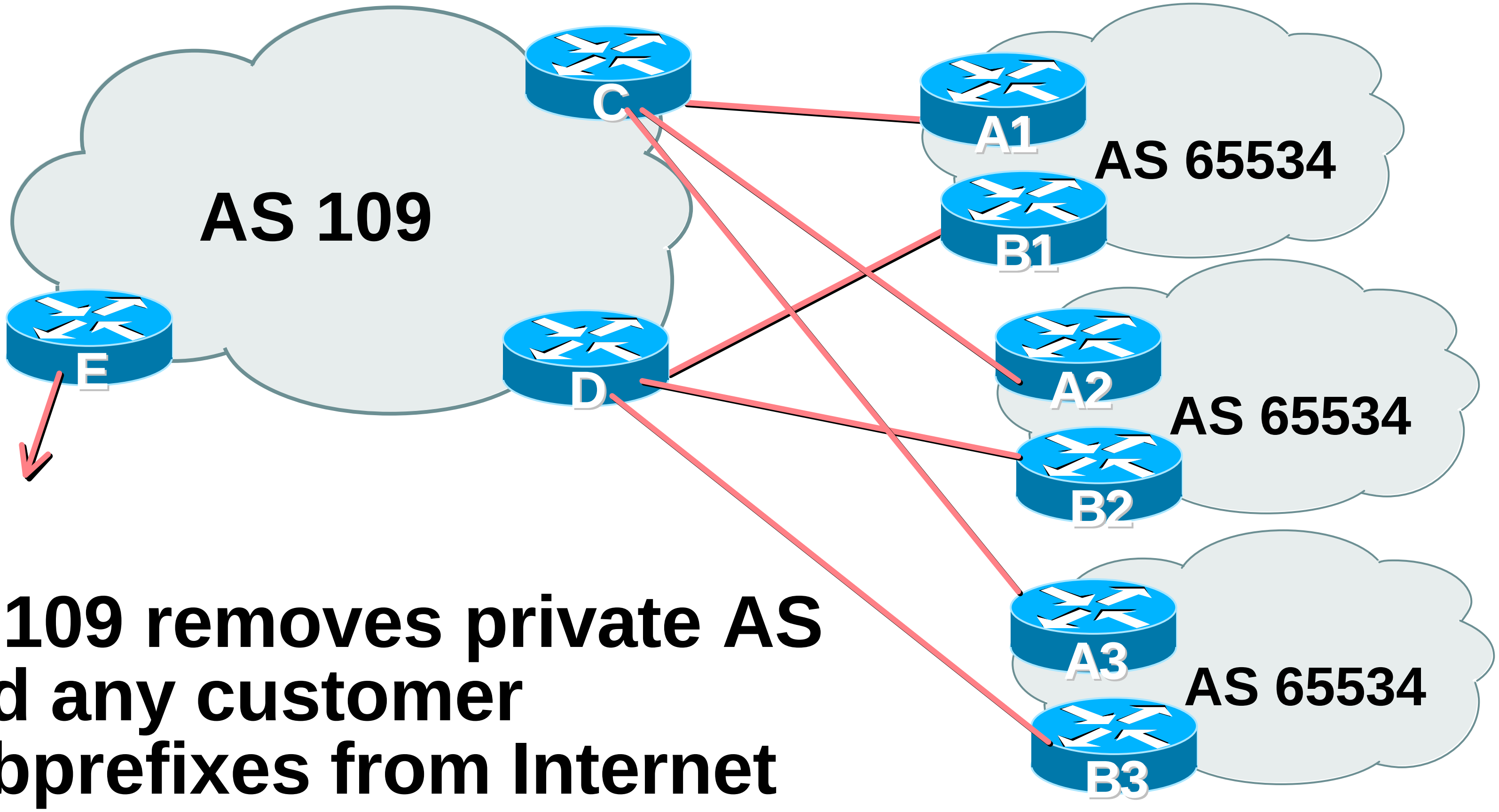
- Could also use BGP communities

# Two links to the same ISP

## Multiple Dualhomed Customers

## (RFC2270)

# Multiple Dualhomed Customers (RFC2270)



- **AS109 removes private AS and any customer subprefixes from Internet announcement**

# Multiple Dualhomed Customers

- Customer announcements as per previous example
- Use the *same* private AS for each customer

  documented in RFC2270

  address space is not overlapping

  each customer hears default only
- Router A*n* and B*n* configuration same as Router A and B previously

# Two links to the same ISP

- **Router A1 Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.0.0 mask 255.255.240.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 prefix-list routerC out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerC permit 221.10.0.0/20
ip prefix-list routerC permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.240.0 null0
ip route 221.10.0.0 255.255.224.0 null0
```

**Router B1 configuration is similar but for the other /20**

# Multiple Dualhomed Customers

- Router C Configuration

```
router bgp 109
 neighbor bgp-customers peer-group
 neighbor bgp-customers remote-as 65534
 neighbor bgp-customers default-originate
 neighbor bgp-customers prefix-list default out
 neighbor 222.222.10.1 peer-group bgp-customers
 neighbor 222.222.10.1 description Customer One
 neighbor 222.222.10.1 prefix-list Customer1 in
 neighbor 222.222.10.9 peer-group bgp-customers
 neighbor 222.222.10.9 description Customer Two
 neighbor 222.222.10.9 prefix-list Customer2 in
```

# Multiple Dualhomed Customers

```
 neighbor 222.222.10.17 peer-group bgp-customers
 neighbor 222.222.10.17 description Customer Three
 neighbor 222.222.10.17 prefix-list Customer3 in
!
ip prefix-list Customer1 permit 221.10.0.0/19 le 20
ip prefix-list Customer2 permit 221.16.64.0/19 le 20
ip prefix-list Customer3 permit 221.14.192.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- Router C only allows in /19 and /20 prefixes from customer block
- Router D configuration is almost identical

# Multiple Dualhomed Customers

- Router E Configuration

  assumes customer address space is not part of upstream’s address block

```
router bgp 109
 neighbor 222.222.10.17 remote-as 110
 neighbor 222.222.10.17 remove-private-AS
 neighbor 222.222.10.17 prefix-list Customers out
!
ip prefix-list Customers permit 221.10.0.0/19
ip prefix-list Customers permit 221.16.64.0/19
ip prefix-list Customers permit 221.14.192.0/19
```

- Private AS still visible inside AS109

# Multiple Dualhomed Customers

- If customers' prefixes come from ISP's address block

  do NOT announce them to the Internet

  announce ISP aggregate only

- Router E configuration:

```
router bgp 109
 neighbor 222.222.10.17 remote-as 110
 neighbor 222.222.10.17 prefix-list my-aggregate out
!
ip prefix-list my-aggregate permit 221.8.0.0/13
```

# Two links to different ISPs

## With Redundancy

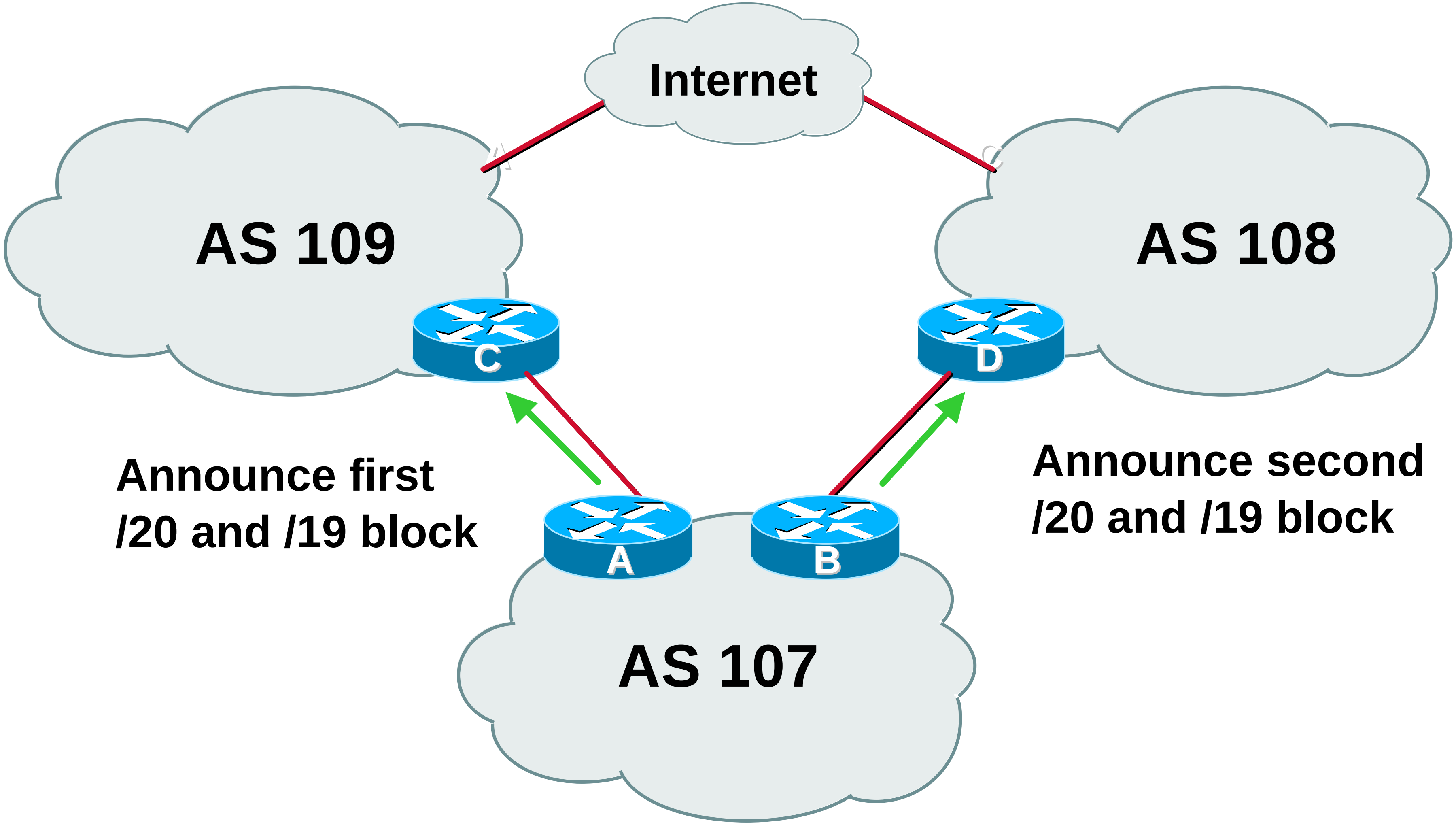
# Two links to different ISPs (with redundancy)

- Announce /19 aggregate on each link
- Split /19 and announce as two /20s, one on each link

  basic inbound loadsharing

- When one link fails, the announcement of the /19 aggregate via the other ISP ensures continued connectivity

# Two links to different ISPs (with redundancy)



Announce first /20 and /19 block

Announce second /20 and /19 block

# Two links to different ISPs (with redundancy)

- Router A Configuration

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.0.0 mask 255.255.240.0
 neighbor 222.222.10.1 remote-as 109
 neighbor 222.222.10.1 prefix-list firstblock out
 neighbor 222.222.10.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
!
ip prefix-list firstblock permit 221.10.0.0/20
ip prefix-list firstblock permit 221.10.0.0/19
```

# Two links to different ISPs (with redundancy)

- Router B Configuration

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 220.1.5.1 remote-as 108
 neighbor 220.1.5.1 prefix-list secondblock out
 neighbor 220.1.5.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
!
ip prefix-list secondblock permit 221.10.16.0/20
ip prefix-list secondblock permit 221.10.0.0/19
```

# Two links to different ISPs

## More Controlled Loadsharing

# Loadsharing with different ISPs

- Announce /19 aggregate on each link

  On first link, announce /19 as normal

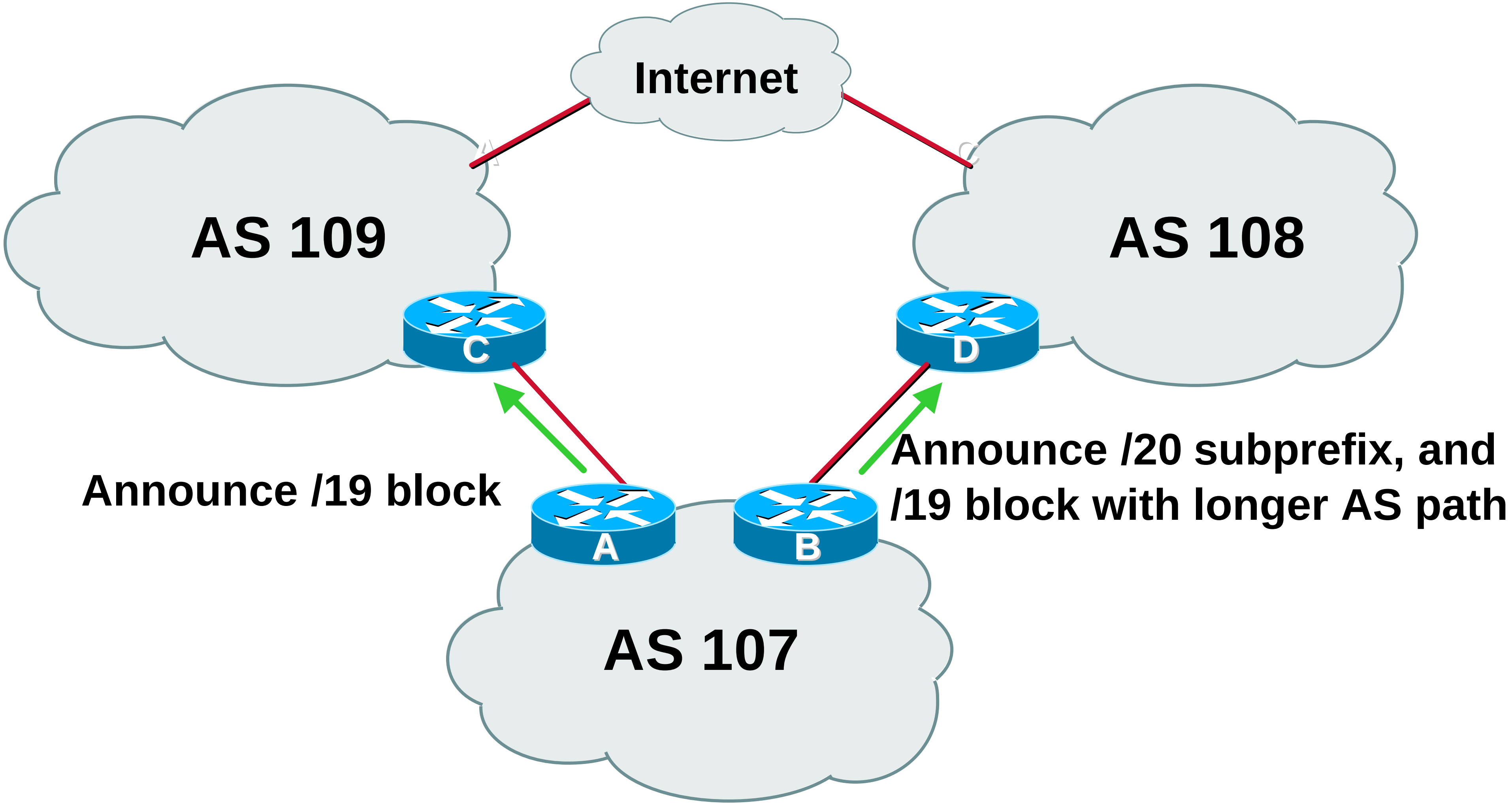
  On second link, announce /19 with longer AS PATH, and announce one /20 subprefix

    controls loadsharing between upstreams and the Internet

- Vary the subprefix size and AS PATH length until “perfect” loadsharing achieved

- Still require redundancy!

# Loadsharing with different ISPs



Announce /19 block

Announce /20 subprefix, and /19 block with longer AS path

# Loadsharing with different ISPs

- Router A Configuration

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 109
 neighbor 222.222.10.1 prefix-list default in
 neighbor 222.222.10.1 prefix-list aggregate out
!
ip prefix-list aggregate permit 221.10.0.0/19
```

# Loadsharing with different ISPs

- Router B Configuration

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 220.1.5.1 remote-as 108
 neighbor 220.1.5.1 prefix-list default in
 neighbor 220.1.5.1 prefix-list subblocks out
 neighbor 220.1.5.1 route-map routerD out
!
```

..next slide..

# Loadsharing with different ISPs

```
route-map routerD permit 10
 match ip address prefix-list aggregate
 set as-path prepend 107 107
route-map routerD permit 20
!
ip prefix-list subblocks permit 221.10.0.0/19 le 20
ip prefix-list aggregate permit 221.10.0.0/19
```

# Service Provider Multihoming

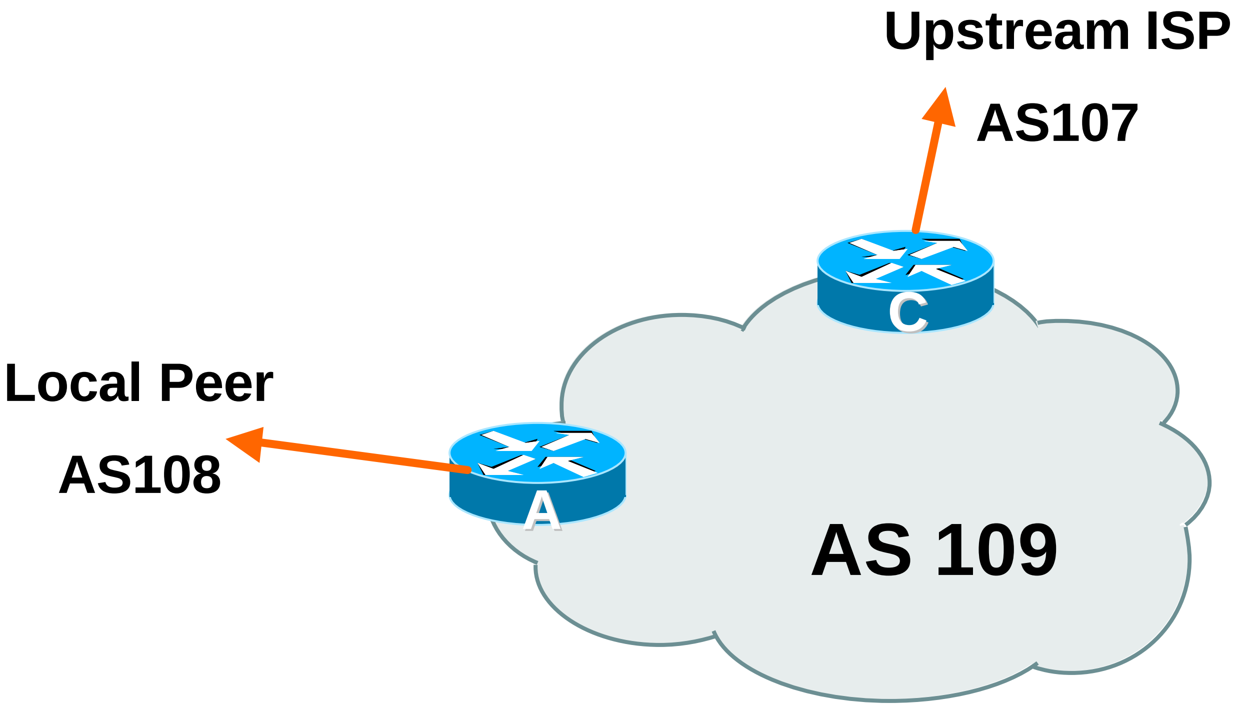
## One Upstream, One local peer

# One Upstream, One Local Peer

- Announce /19 aggregate on each link
- Accept default route only from upstream

  Either 0.0.0.0/0 or a network which can be used as default
- Accept all routes from local peer
- Border routers talk iBGP with each other

# One Upstream, One Local Peer



# One Upstream, One Local Peer

- **Router A Configuration**

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 108
 neighbor 222.222.10.2 prefix-list my-block out
 neighbor 222.222.10.2 prefix-list AS108-peer in
!
ip prefix-list AS108-peer permit 222.5.16.0/19
ip prefix-list AS108-peer permit 221.240.0.0/20
ip prefix-list my-block permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.224.0 null0
```

# One Upstream, One Local Peer

- Router C Configuration

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 neighbor 222.222.10.1 prefix-list default in
 neighbor 222.222.10.1 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 null0
```

# One Upstream, One Local Peer

- Two configurations possible for Router A
  - Filtering on ASes assumes peer knows what they are doing (never do this)
  - Prefix-list higher maintenance, but safer
- Local traffic goes to and from local peer, everything else goes to upstream
- Routers A and C have minimum memory and CPU requirements

# Service Provider Multihoming

## Two Upstreams, One local peer

# Two Upstreams, One Local Peer

- Two configuration options:

  Accept full routing from both upstreams

    Expensive!

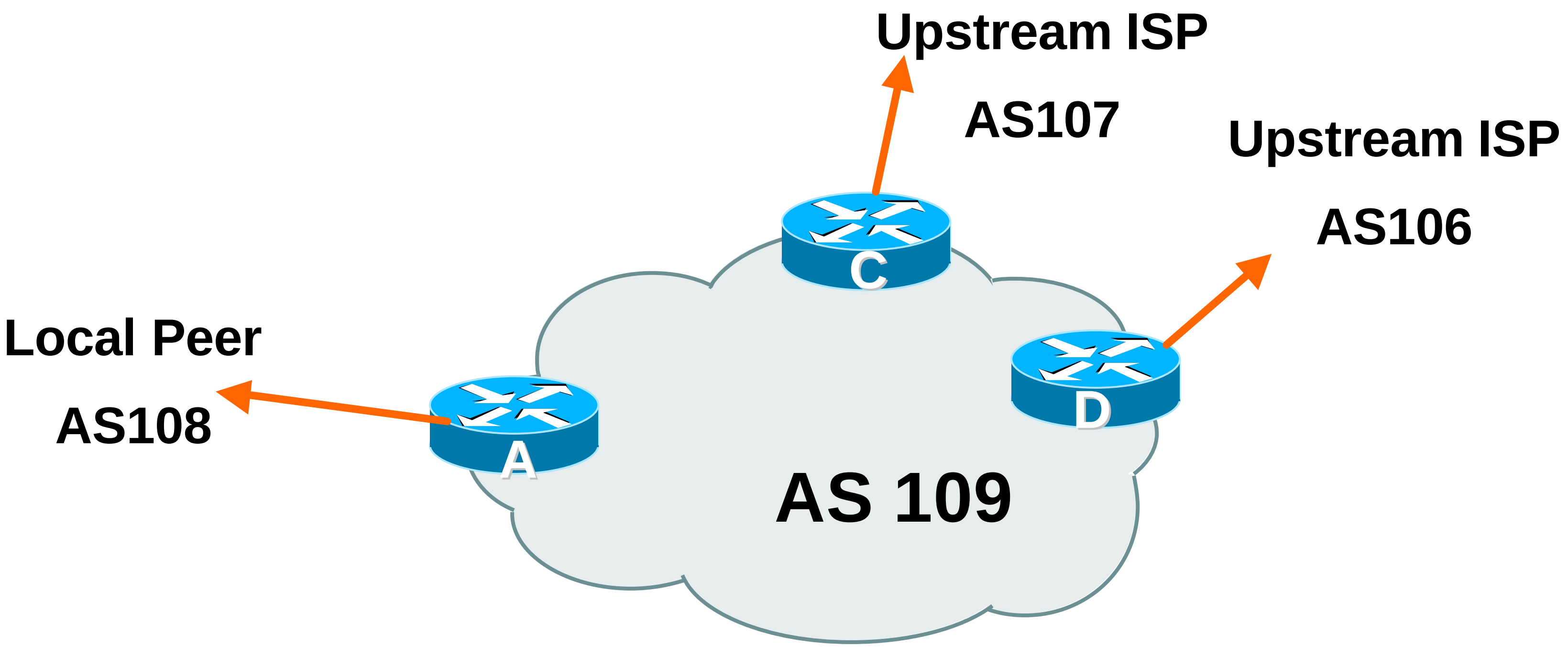
    But this is the popular choice today?!!

  Accept default from one upstream and some routes from the other upstream

    Best compromise, not expensive!

    Better convergence rate and stability

# Two Upstreams, One Local Peer



- Router A configuration is as previously

# Two Upstreams, One Local Peer – Full Routes

- Router C Configuration

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 neighbor 222.222.10.1 prefix-list rfc1918-deny in
 neighbor 222.222.10.1 prefix-list my-block out
 neighbor 222.222.10.1 route-map AS107-loadshare in
!
ip prefix-list my-block permit 221.10.0.0/19
! See earlier in tutorial for RFC1918 list
!
ip route 221.10.0.0 255.255.224.0 null0
..next slide
```

# Two Upstreams, One Local Peer – Full Routes

```
ip as-path access-list 10 permit ^(107_)+$
ip as-path access-list 10 permit ^(107_)+_[0-9]+$
!
route-map AS107-loadshare permit 10
 match ip as-path 10
 set local-preference 120
route-map AS107-loadshare permit 20
 set local-preference 80
!
```

# Two Upstreams, One Local Peer – Full Routes

- Router C configuration:
  - Accept full routes from AS107
  - Tag prefixes originated by AS107 and AS107’s neighbouring ASes with local preference 120
  - Remaining prefixes tagged with local preference of 80
  - Traffic to those ASes will go over AS107 link
  - Traffic to other all other ASes will go over the link to AS106
- Router D configuration same as Router C without the route-map
  - Hears full routing table!

# Two Upstreams, One Local Peer – Full Routes

- Full routes from upstreams

  Expensive – needs lots of memory today

  Expensive – contributes to network instability

  Need to play preference games

  Previous example is only an example – real life will need improved fine-tuning!

  Previous example doesn't consider inbound traffic – see earlier slides for examples

# Two Upstreams, One Local Peer – Partial Routes

- Router C Configuration

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 neighbor 222.222.10.1 prefix-list rfc1918-nodef-deny in
 neighbor 222.222.10.1 prefix-list my-block out
 neighbor 222.222.10.1 filter-list 10 in
 neighbor 222.222.10.1 route-map tag-default-low in
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
! See earlier in tutorial for RFC1918 list
!
ip route 221.10.0.0 255.255.224.0 null0
```

# Two Upstreams, One Local Peer – Partial Routes

```
ip as-path access-list 10 permit ^(107_)+$
ip as-path access-list 10 permit ^(107_)+_[0-9]+$
!
route-map tag-default-low permit 10
 match ip address prefix-list default
 set local-preference 80
route-map tag-default-low permit 20
!
```

# Two Upstreams, One Local Peer – Partial Routes

- Router D Configuration

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 106
 neighbor 222.222.10.5 prefix-list default in
 neighbor 222.222.10.5 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 null0
```

# Two Upstreams, One Local Peer – Partial Routes

- Router C configuration:

  Accept full routes from AS107

  (or get them to send less)

  Filter ASNs so only AS107 and AS107's neighbouring ASes are accepted

  Allow default, and set it to local preference 80

  Traffic to those ASes will go over AS107 link

  Traffic to other all other ASes will go over the link to AS106

  If AS106 link fails, backup via AS107 – and vice-versa

# Two Upstreams, One Local Peer – Partial Routes

- Partial routes from upstreams

  Not expensive – only carry the routes necessary for loadsharing

  Not expensive – network more stable!

  Need to filter on AS paths

  Previous example is only an example – real life will need improved fine-tuning!

  Previous example doesn’t consider inbound traffic – see earlier slides for examples

# BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Deploying BGP in an ISP network
- Trouble & Troubleshooting
- Multihoming Examples
- Using Communities

# Communities

# Community usage

- RFC1998
- Examples of SP applications

# RFC1998

- Informational RFC
- Describes how to implement loadsharing and backup on multiple inter-AS links

  BGP communities used to determine local preference in upstream's network

- Gives control to the customer
- Simplifies upstream's configuration

  simplifies network operation!

# RFC1998

- **Community values defined to have particular meanings:**

| | | |
|---|---|---|
| ASx:100 | set local pref 100 | preferred route |
| ASx:90 | set local pref 90 | backup route if dualhomed on ASx |
| ASx:80 | set local pref 80 | main link is to another ISP with same AS path length |
| ASx:70 | set local pref 70 | main link is to another ISP |

# RFC1998

- Sample Customer Router Configuration

```
router bgp 107
 neighbor x.x.x.x remote-as 109
 neighbor x.x.x.x description Backup ISP
 neighbor x.x.x.x route-map config-community out
 neighbor x.x.x.x send-community
!
ip as-path access-list 20 permit ^$
ip as-path access-list 20 deny .*
!
route-map config-community permit 10
 match as-path 20
 set community 109:90
```

# RFC1998

- **Sample ISP Router Configuration**

```
! Homed to another ISP
ip community-list 70 permit 109:70
! Homed to another ISP with equal ASPATH length
ip community-list 80 permit 109:80
! Customer backup routes
ip community-list 90 permit 109:90
!
route-map set-customer-local-pref permit 10
 match community 70
 set local-preference 70
```

# RFC1998

- Sample ISP Router Configuration

```
route-map set-customer-local-pref permit 20
 match community 80
 set local-preference 80
!
route-map set-customer-local-pref permit 30
 match community 90
 set local-preference 90
!
route-map set-customer-local-pref permit 40
 set local-preference 100
```

# RFC1998

- Supporting RFC1998

  many ISPs do, more should

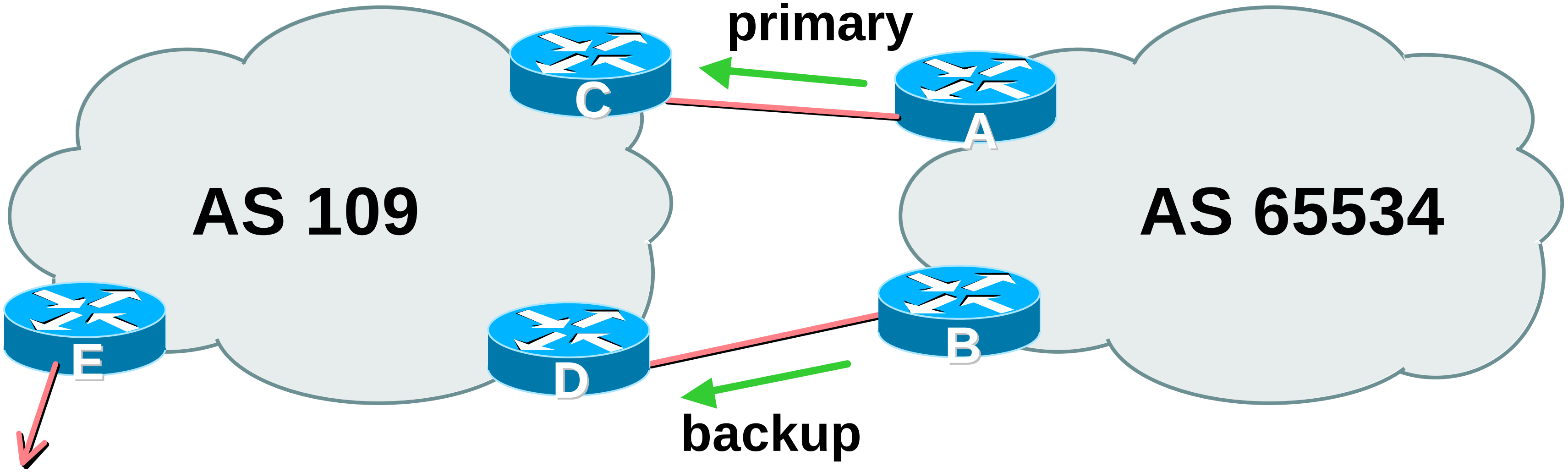
  check AS object in the Internet Routing Registry

  if you do, insert comment in AS object in the IRR

# Two links to the same ISP

## One link primary, the other link backup only

# Two links to the same ISP



- AS109 proxy aggregates for AS 65534

# Two links to the same ISP (one as backup only)

- Announce /19 aggregate on each link

  primary link makes standard announcement

  backup link sends community

- When one link fails, the announcement of the /19 aggregate via the other link ensures continued connectivity

# Two links to the same ISP (one as backup only)

- Router A Configuration

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 description RouterC
 neighbor 222.222.10.2 prefix-list aggregate out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
```

# Two links to the same ISP (one as backup only)

- **Router B Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.6 remote-as 109
 neighbor 222.222.10.6 description RouterD
 neighbor 222.222.10.6 send-community
 neighbor 222.222.10.6 prefix-list aggregate out
 neighbor 222.222.10.6 route-map routerD-out out
 neighbor 222.222.10.6 prefix-list default in
 neighbor 222.222.10.6 route-map routerD-in in
!
..next slide
```

# Two links to the same ISP (one as backup only)

```
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
route-map routerD-out permit 10
 match ip address prefix-list aggregate
 set community 109:90
route-map routerD-out permit 20
!
route-map routerD-in permit 10
 set local-preference 90
!
```

# Two links to the same ISP (one as backup only)

- Router C Configuration (main link)

```
router bgp 109
 neighbor 222.222.10.1 remote-as 65534
 neighbor 222.222.10.1 default-originate
 neighbor 222.222.10.1 prefix-list Customer in
 neighbor 222.222.10.1 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
```

# Two links to the same ISP (one as backup only)

- Router D Configuration (backup link)

```
router bgp 109
 neighbor 222.222.10.5 remote-as 65534
 neighbor 222.222.10.5 default-originate
 neighbor 222.222.10.5 prefix-list Customer in
 neighbor 222.222.10.5 route-map bgp-cust-in in
 neighbor 222.222.10.5 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
..next slide
```

# Two links to the same ISP (one as backup only)

```
  ip prefix-list Customer permit 221.10.0.0/19
  ip prefix-list default permit 0.0.0.0/0
  !
  ip community-list 90 permit 109:90
  !
<snip>
  route-map bgp-cust-in permit 30
   match community 90
   set local-preference 90
  route-map bgp-cust-in permit 40
   set local-preference 100
```

# Service Providers use of Communities

## Some working examples

# Background

- RFC1998 is okay for “simple” multihomed customers

  assumes that upstreams are interconnected

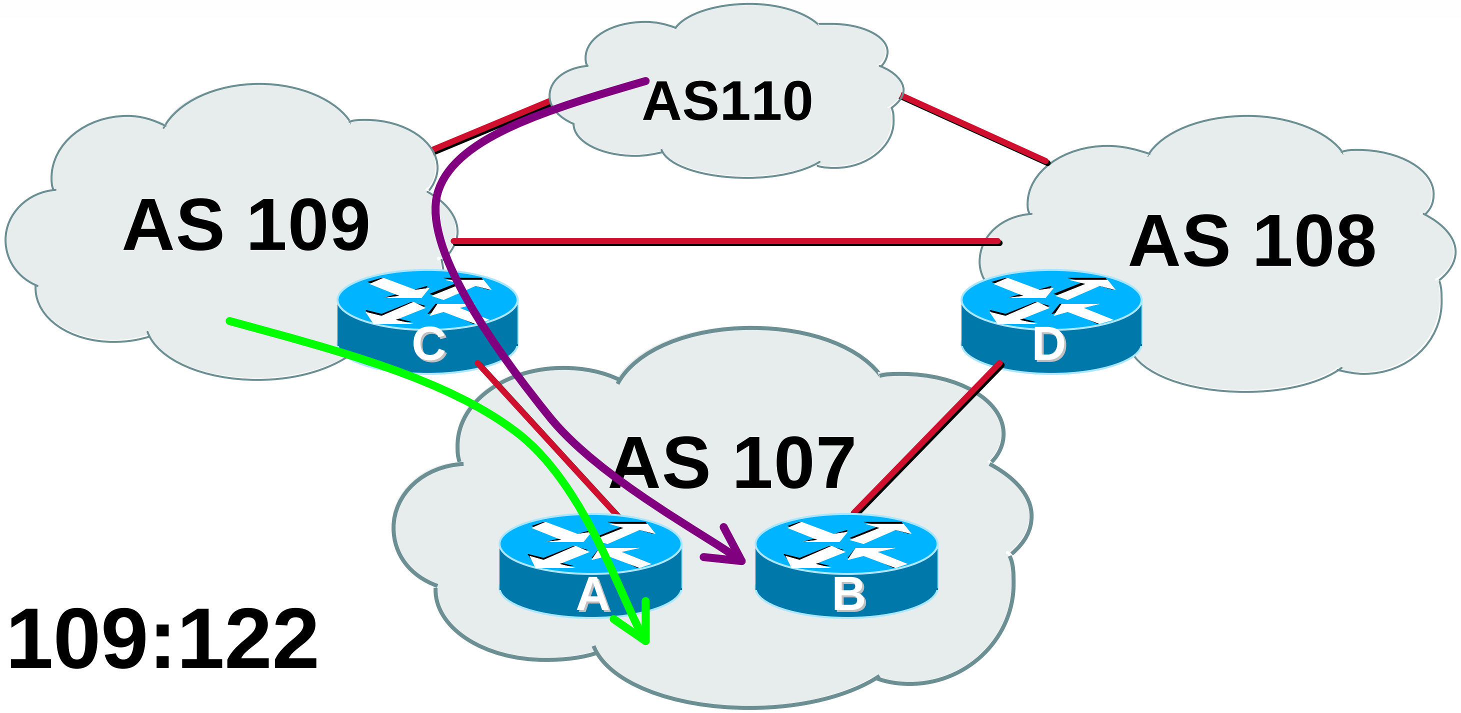
- ISPs create many other communities to handle more complex situations

# More community definitions

| | |
|---|---|
| ASx:122 | set local pref 120 and set local pref high on upstreams |
| ASx:121 | set local pref 120 and set local pref low on upstreams |
| ASx:120 | set local pref 120 (opposite to ASx:80) |
| ASx:82 | set local pref 80 and set local pref high on upstreams |
| ASx:81 | set local pref 80 and set local pref low on upstreams |
| ASx:21 | announce to customers with no-export |
| ASx:20 | announce only to backbone and customers |
| ASx:3 | set 3x as-path prepend on peer announcement |
| ASx:2 | set 2x as-path prepend on peer announcement |
| ASx:1 | set 1x as-path prepend on peer announcement |

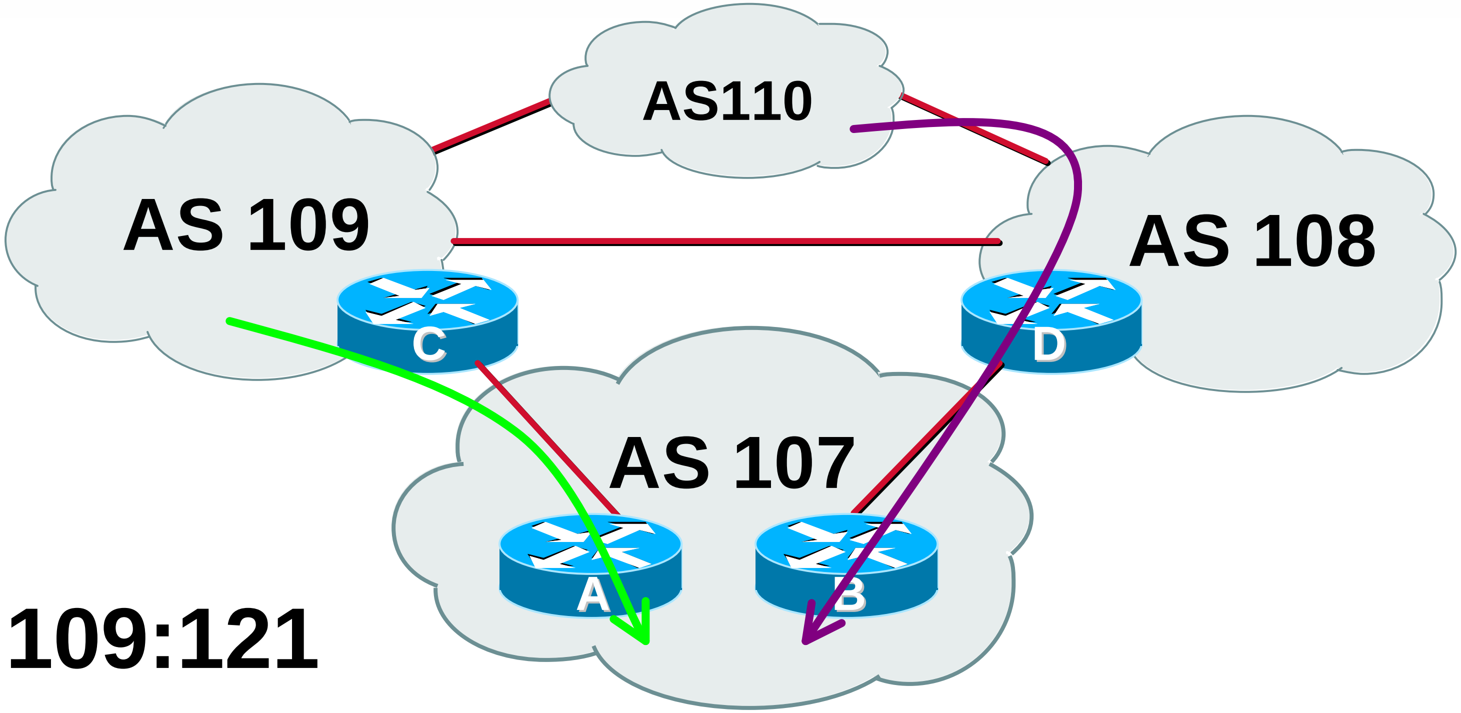
(and variations on this theme depending on local conditions, e.g. IXPs, domestic vs. international transit, etc.)

# Examples



- **109:122**
  - **traffic in AS109 comes directly to you**
  - **traffic in AS110 sent to AS109 rather than best path**

# Examples



- **109:121**

  **traffic in AS109 comes directly to you**

  **traffic in AS110 sent to AS108 rather than best path**

# Examples

- 109:3

  prepend any announcements to peers of AS109 with 109_109_109

  “AS109 is my backup transit AS”

- 109:20

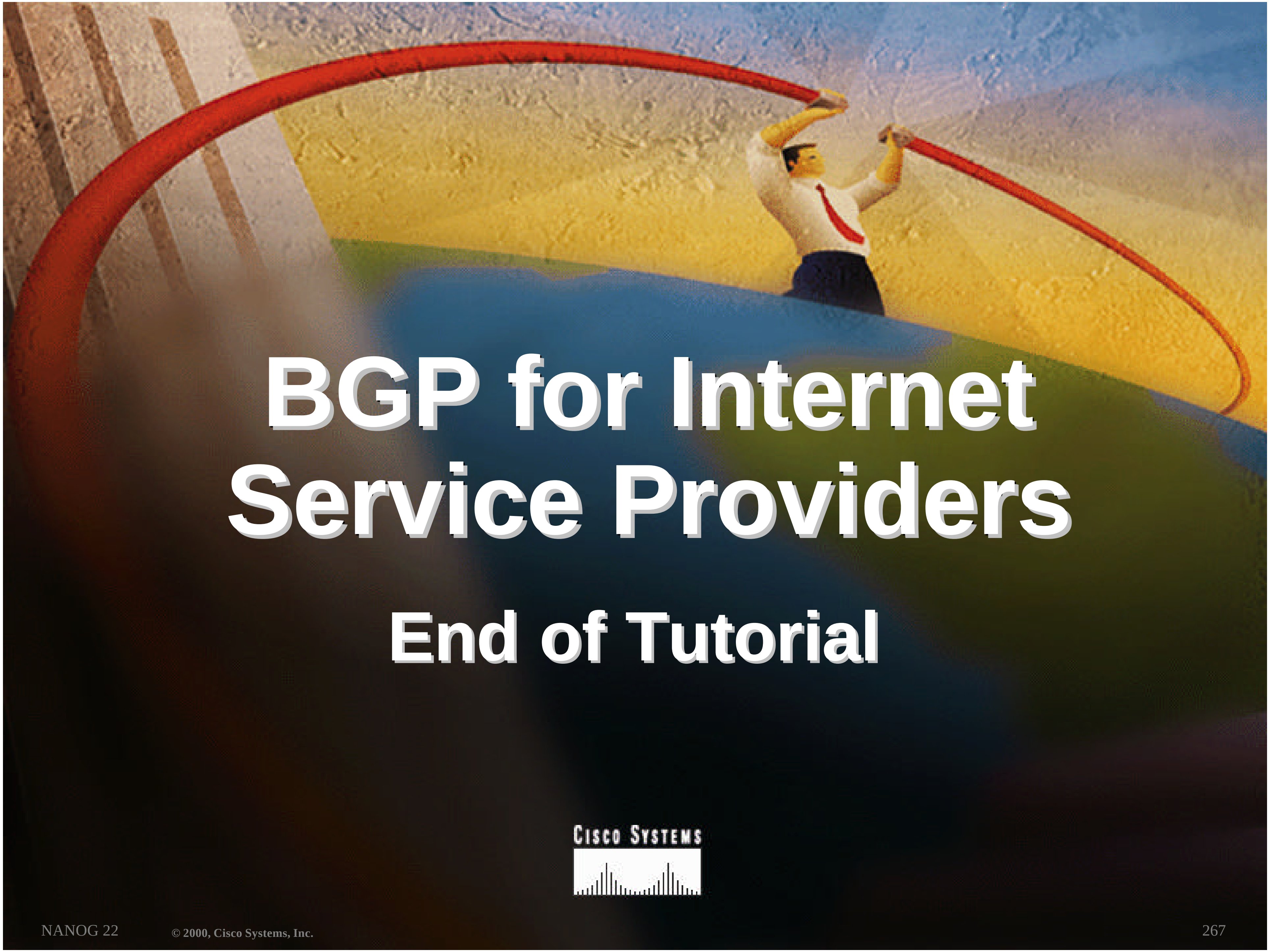
  Don’t announce outside upstream’s customer base

  “AS109 provides local connections only”

  109:21 is very similar

# BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Deploying BGP in an ISP network
- Trouble & Troubleshooting
- Multihoming Examples
- Using Communities



# BGP for Internet Service Providers

## End of Tutorial