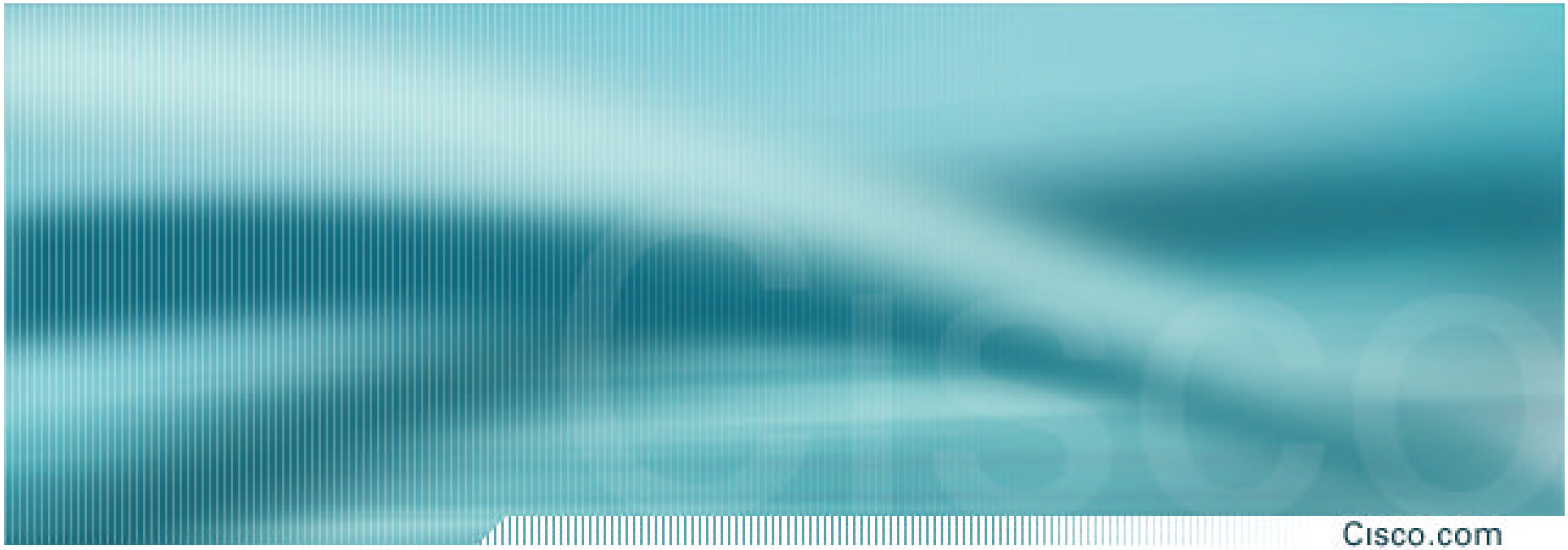


BGP for Internet Service Providers

Philip Smith <pfs@cisco.com>

SANOG I, Kathmandu, Nepal – January 2003

Presentation Slides

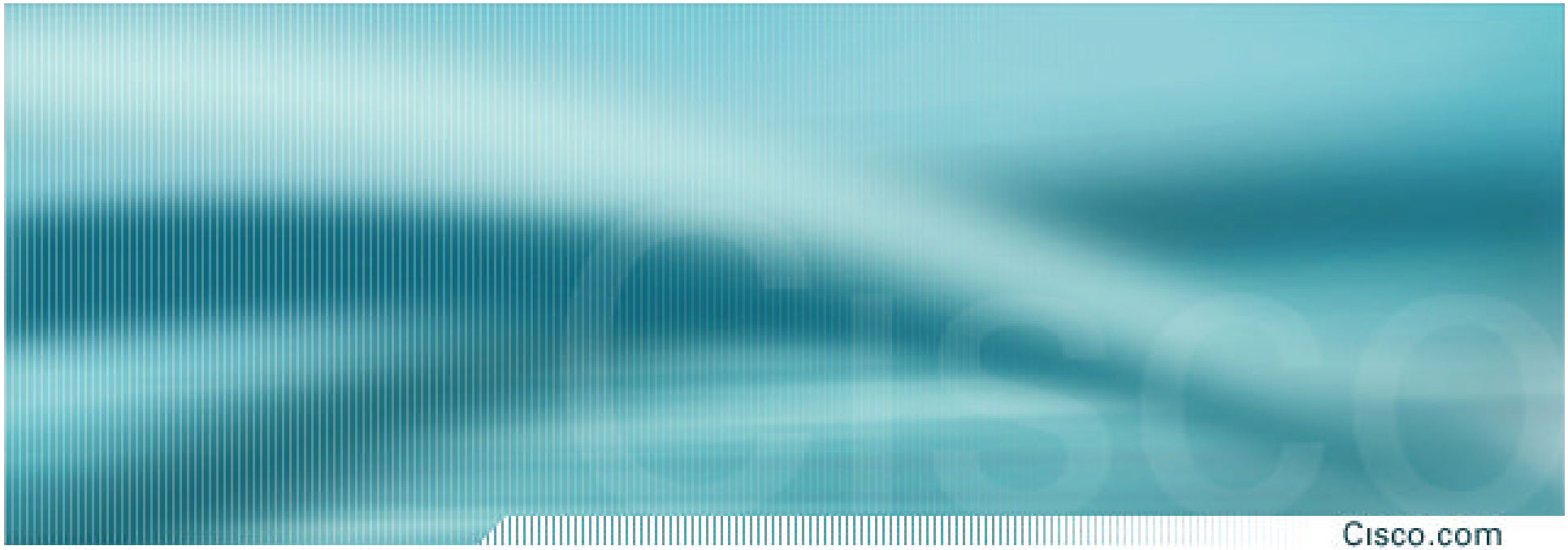
- **Will be available on**

  ftp://ftp-eng.cisco.com/pfs/seminars

- **Feel free to ask questions any time**

BGP for Internet Service Providers

- BGP Basics (recap)
- Scaling BGP
- Using Communities
- Deploying BGP in an ISP network



BGP Basics

What is this BGP thing?

Border Gateway Protocol

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

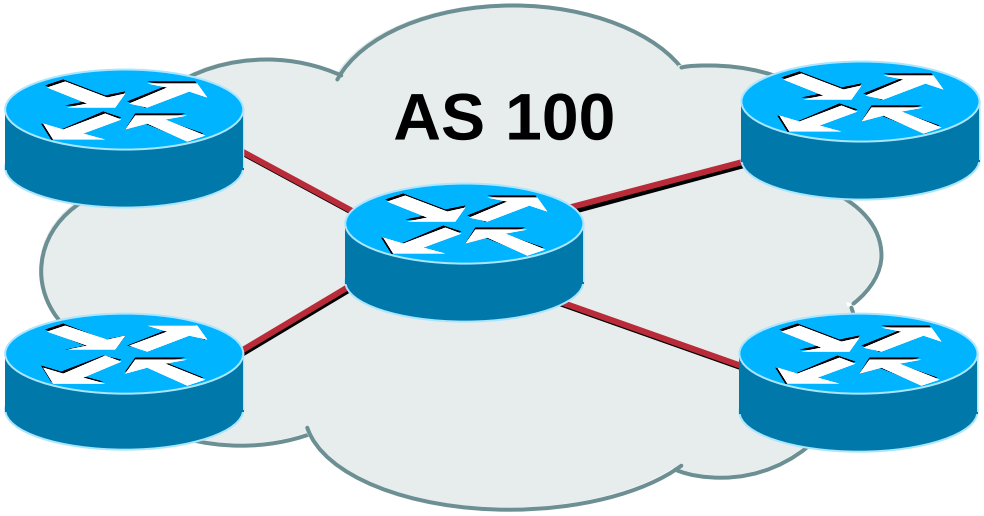
  exterior gateway protocol

- **Described in RFC1771**

  work in progress to update

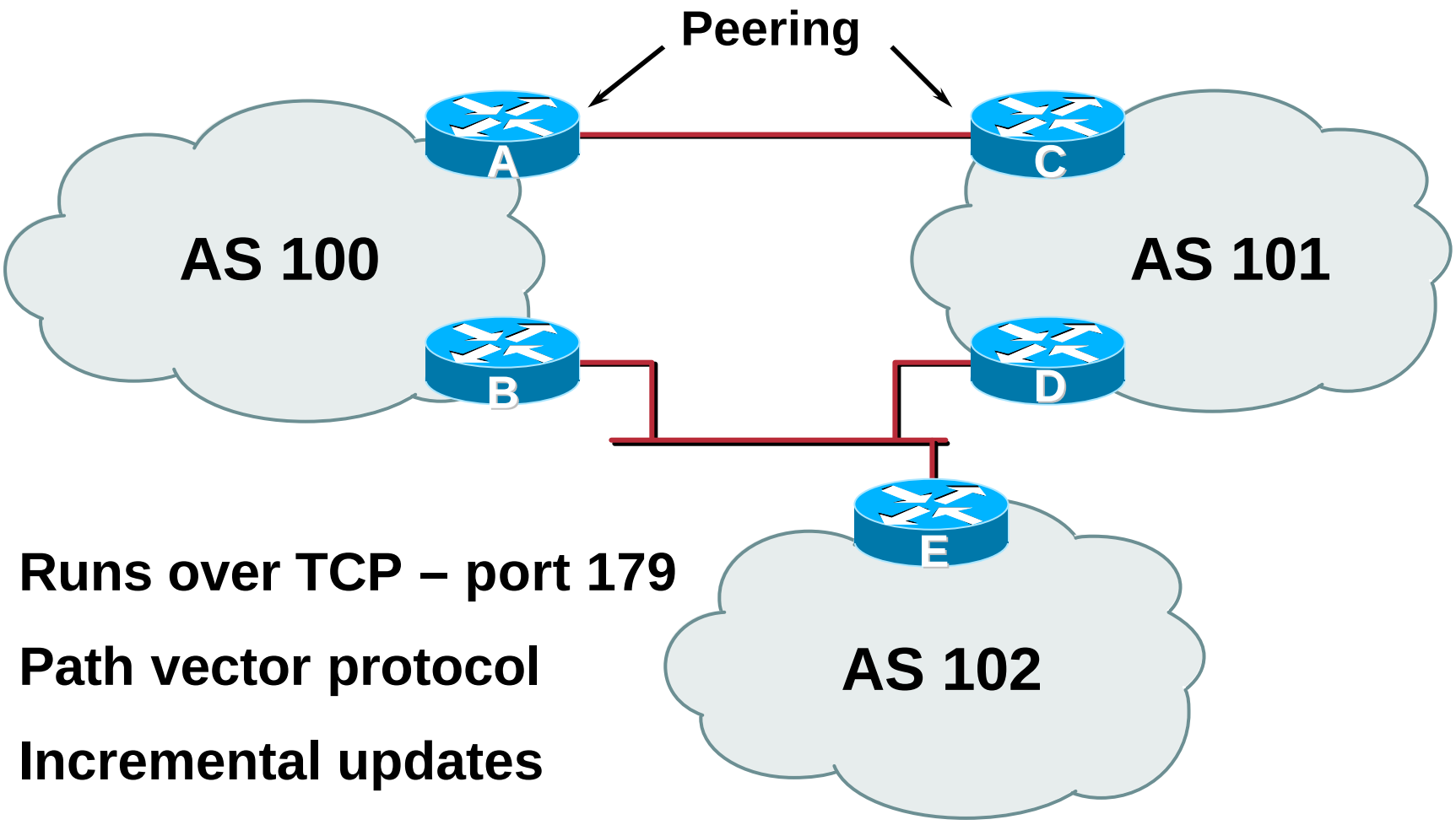
  www.ietf.org/internet-drafts/draft-ietf-idr-bgp4-18.txt

Autonomous System (AS)



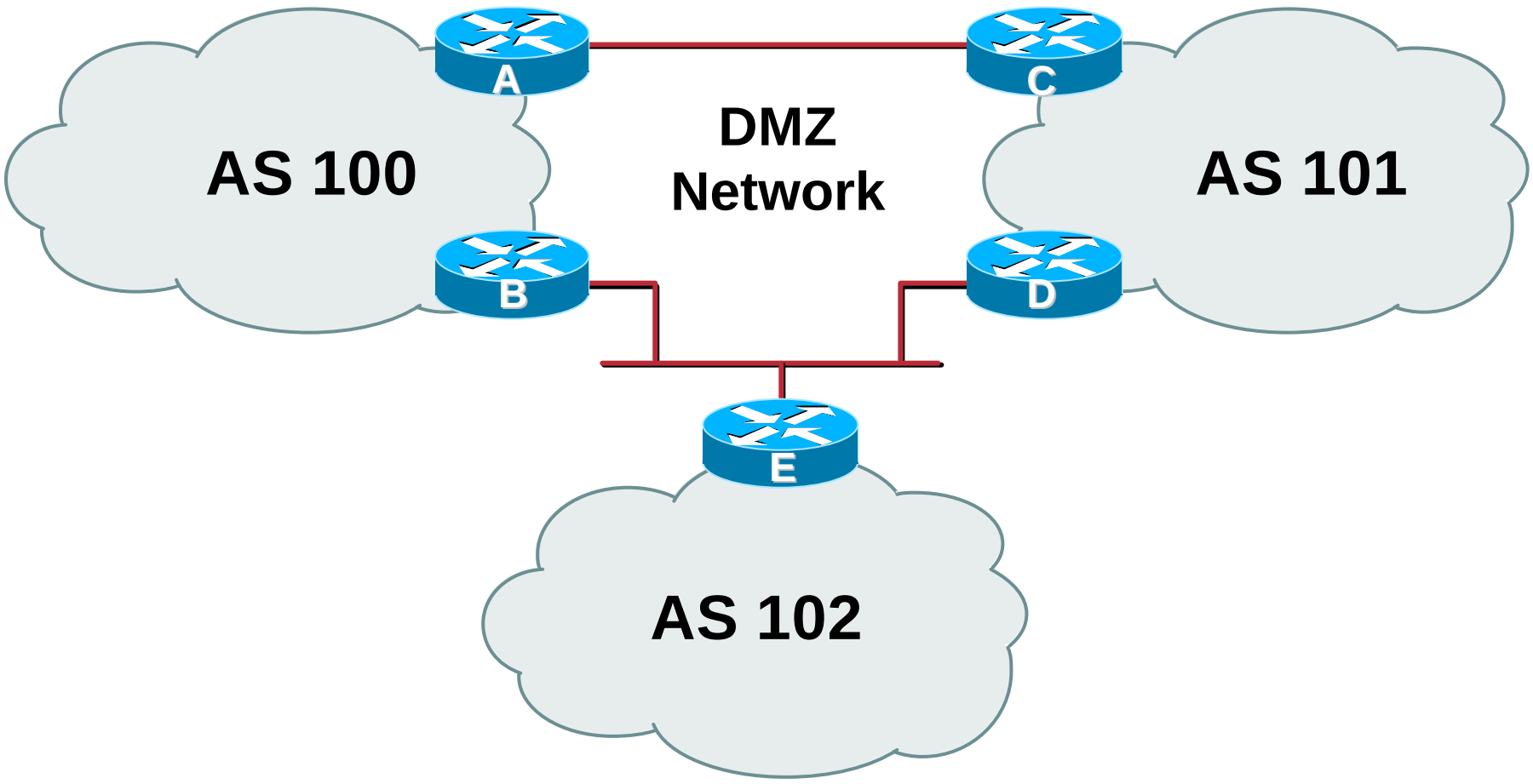
- **Collection of networks with same routing policy**
- **Single routing protocol**
- **Usually under single ownership, trust and administrative control**

BGP Basics



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

Demarcation Zone (DMZ)

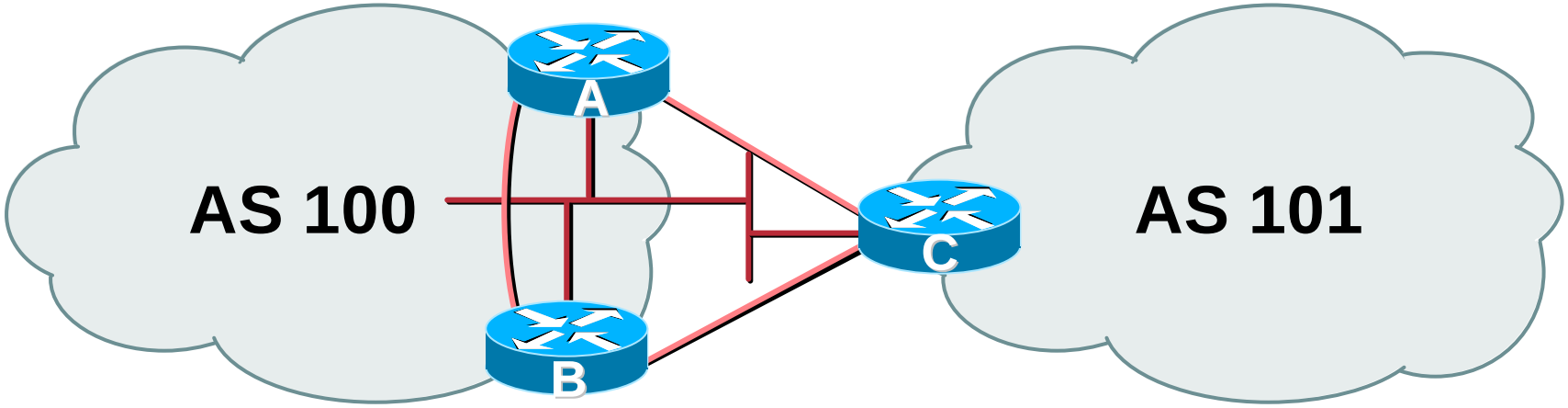


- **Shared network between ASes**

BGP General Operation

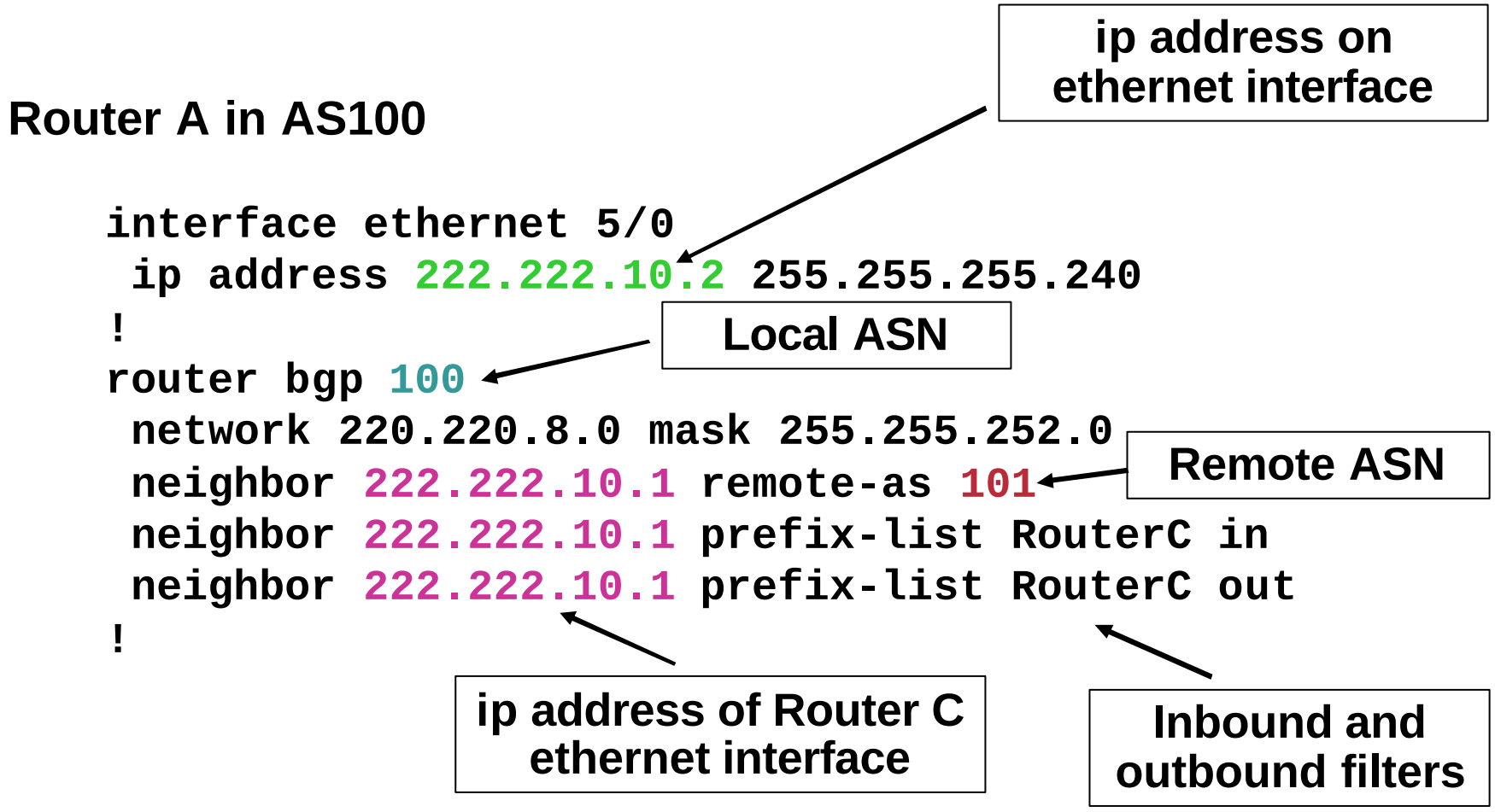
- Learns multiple paths via internal and external BGP speakers
- Picks the best path and installs in the forwarding table
- Best path is sent to external BGP neighbours
- Policies applied by influencing the best path selection

External BGP Peering (eBGP)

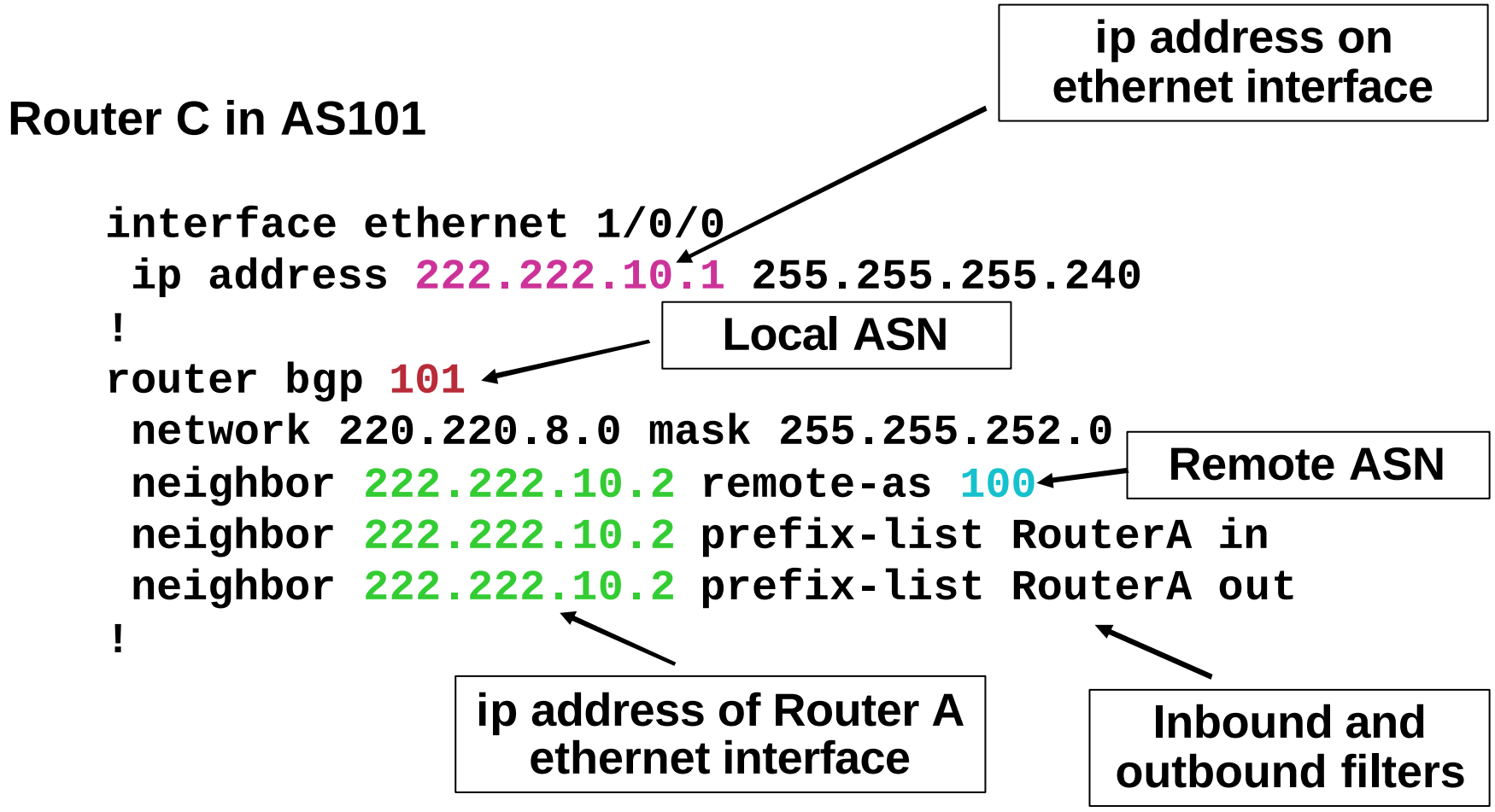


- Between BGP speakers in different AS
- Should be directly connected
- Never run an IGP between eBGP peers

Configuring External BGP



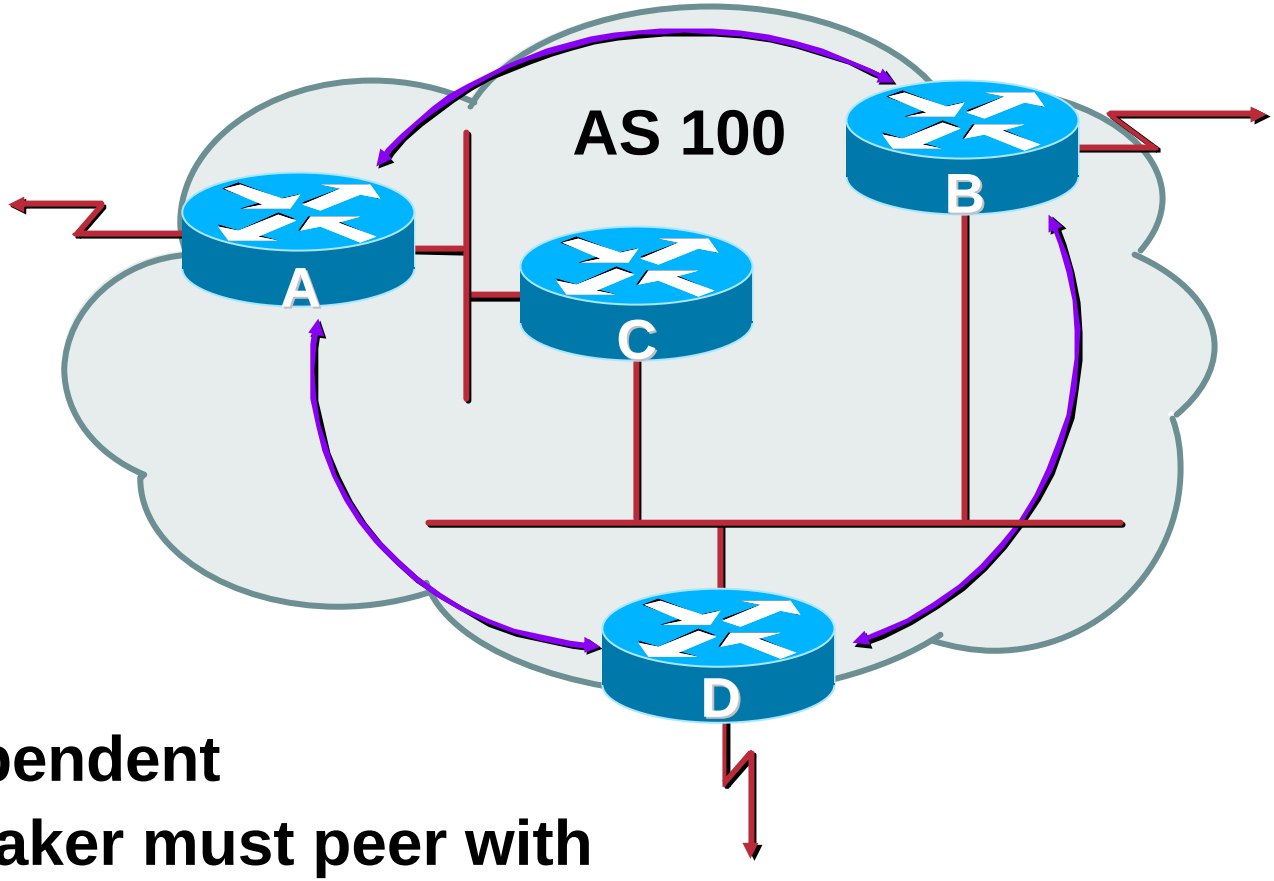
Configuring External BGP



Internal BGP (iBGP)

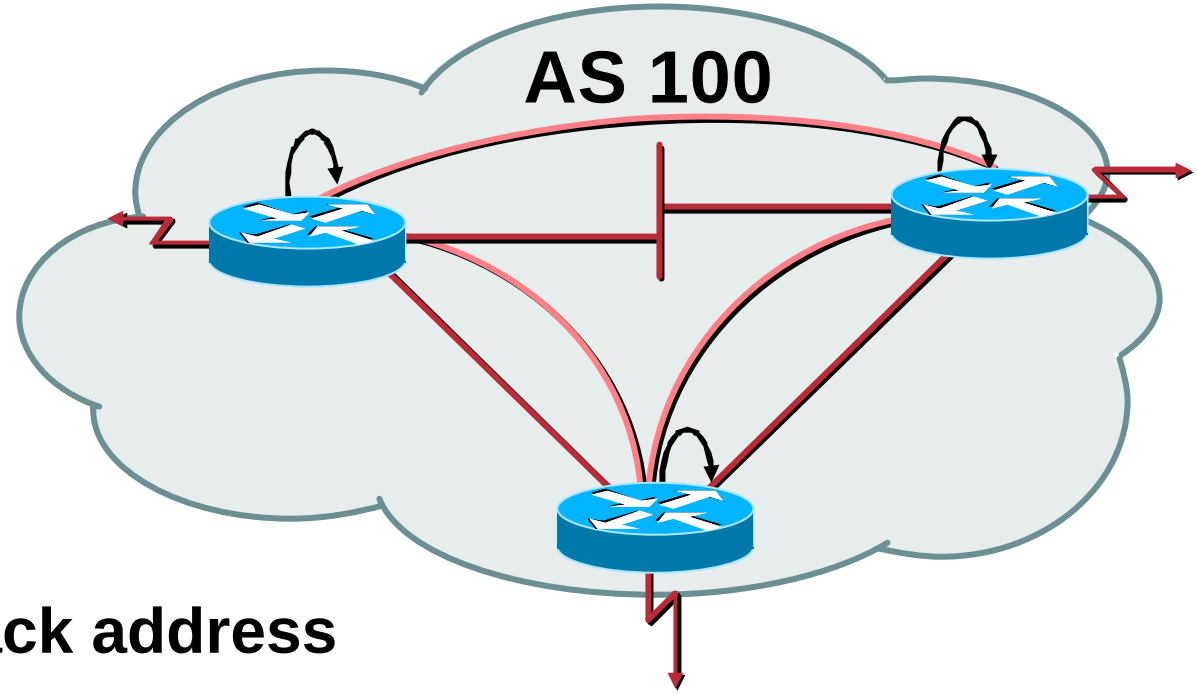
- BGP peer within the same AS
- Not required to be directly connected
  - IGP takes care of inter-BGP speaker connectivity
- iBGP speakers need to be fully meshed
  - they originate connected networks
  - they do not pass on prefixes learned from other iBGP speakers

Internal BGP Peering (iBGP)



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

Peering to Loop-back Address

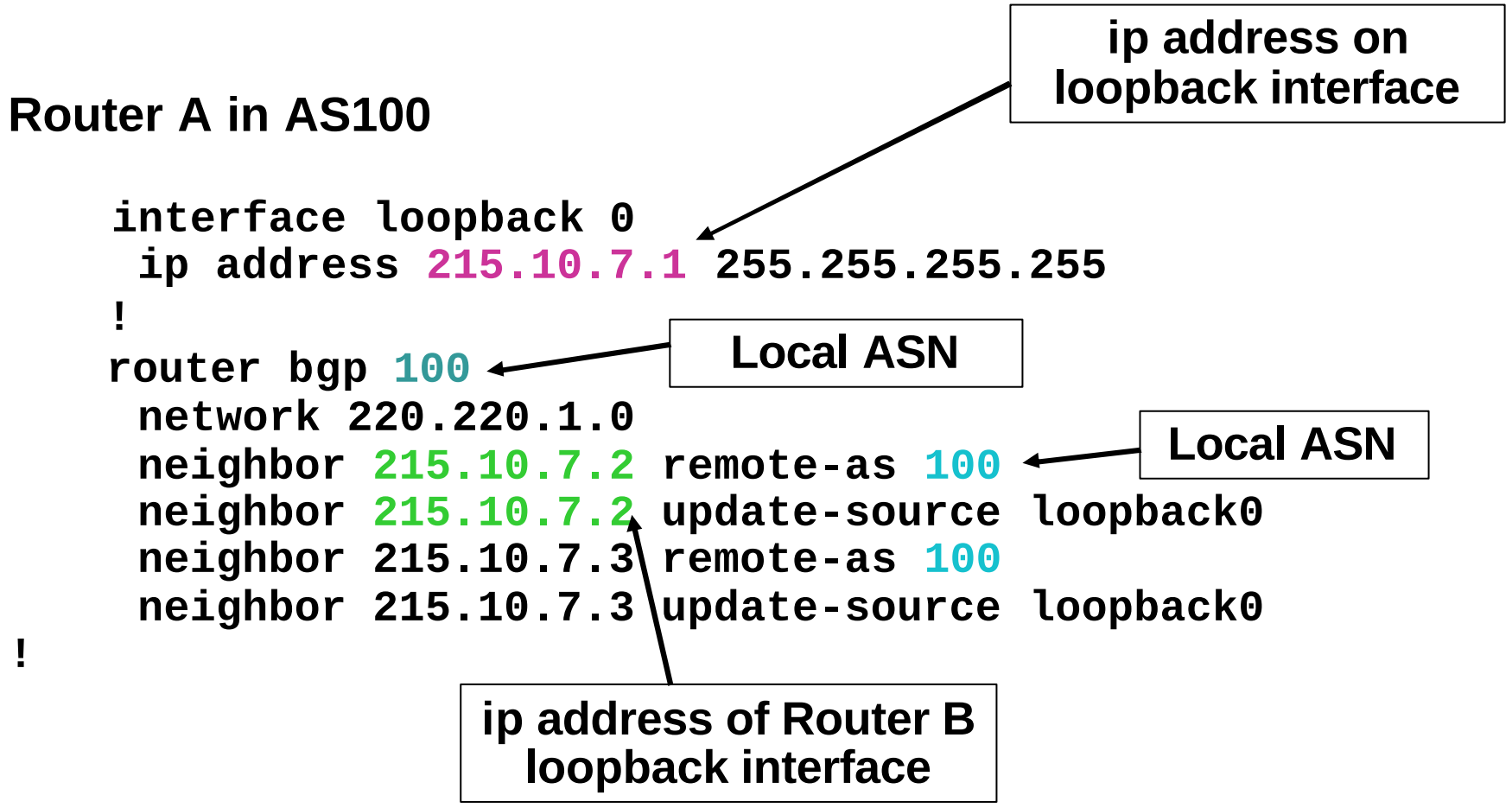


- **Peer with loop-back address**

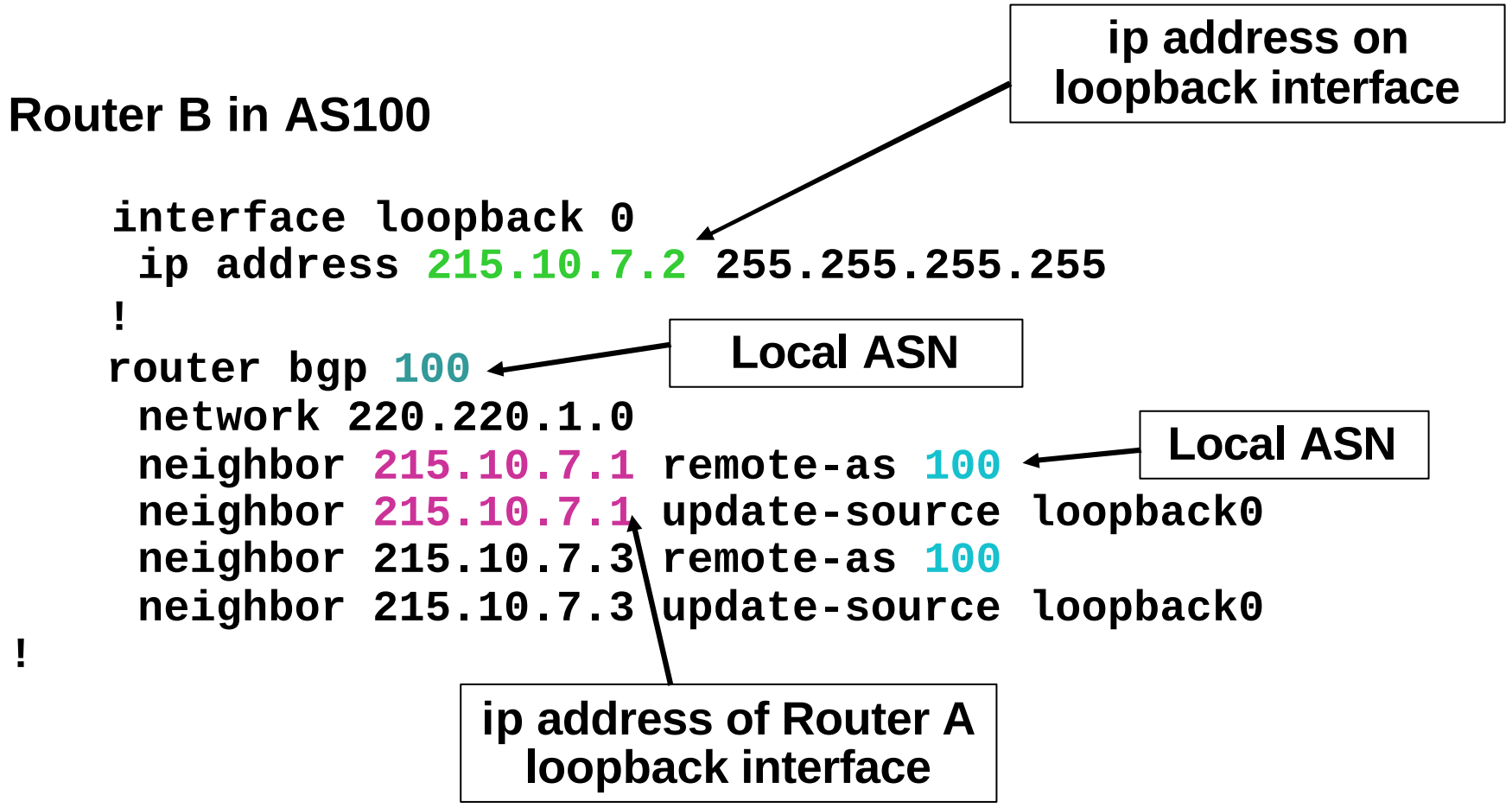
  Loop-back interface does not go down – ever!

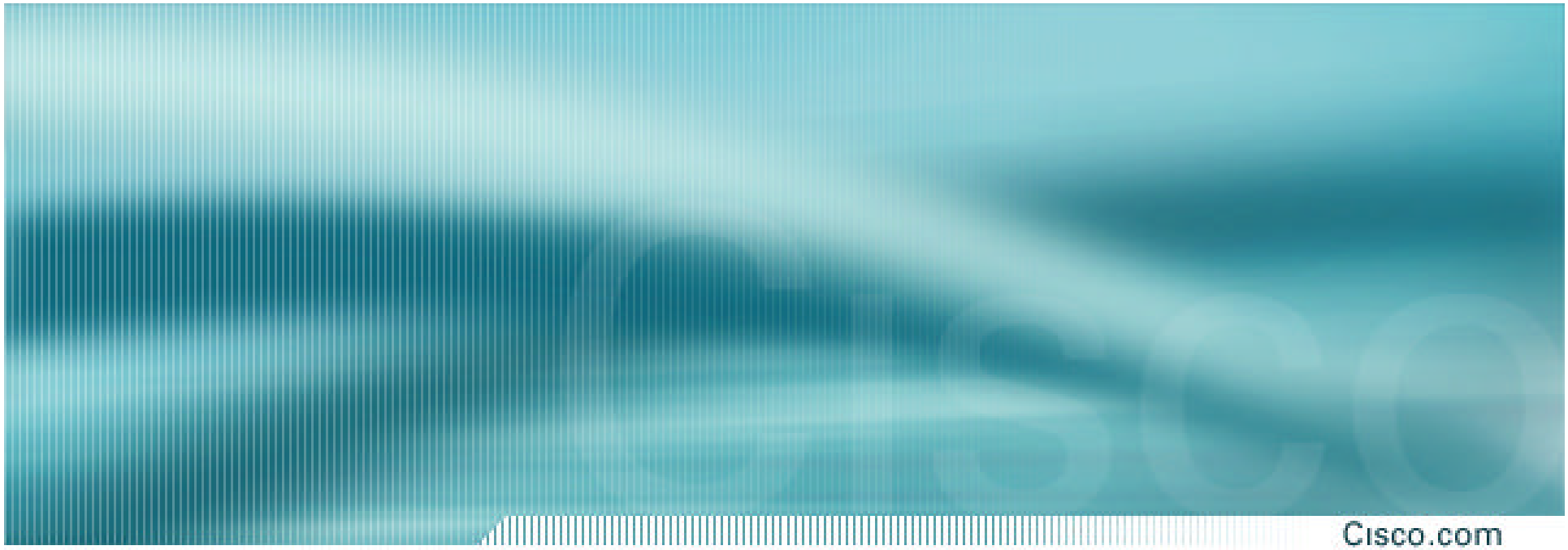
- **iBGP session is not dependent on state of a single interface**
- **iBGP session is not dependent on physical topology**

Configuring Internal BGP



Configuring Internal BGP



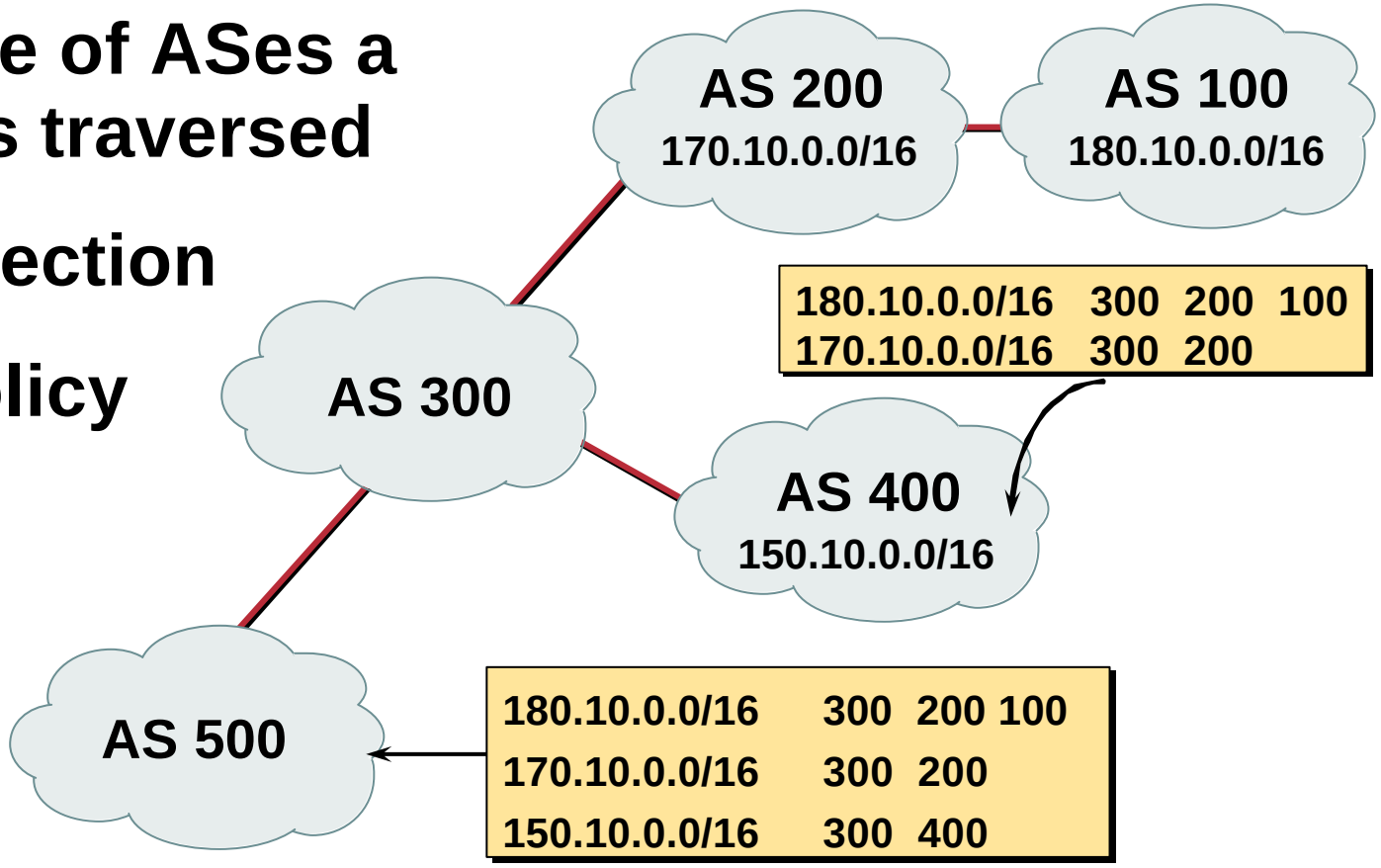


BGP Attributes

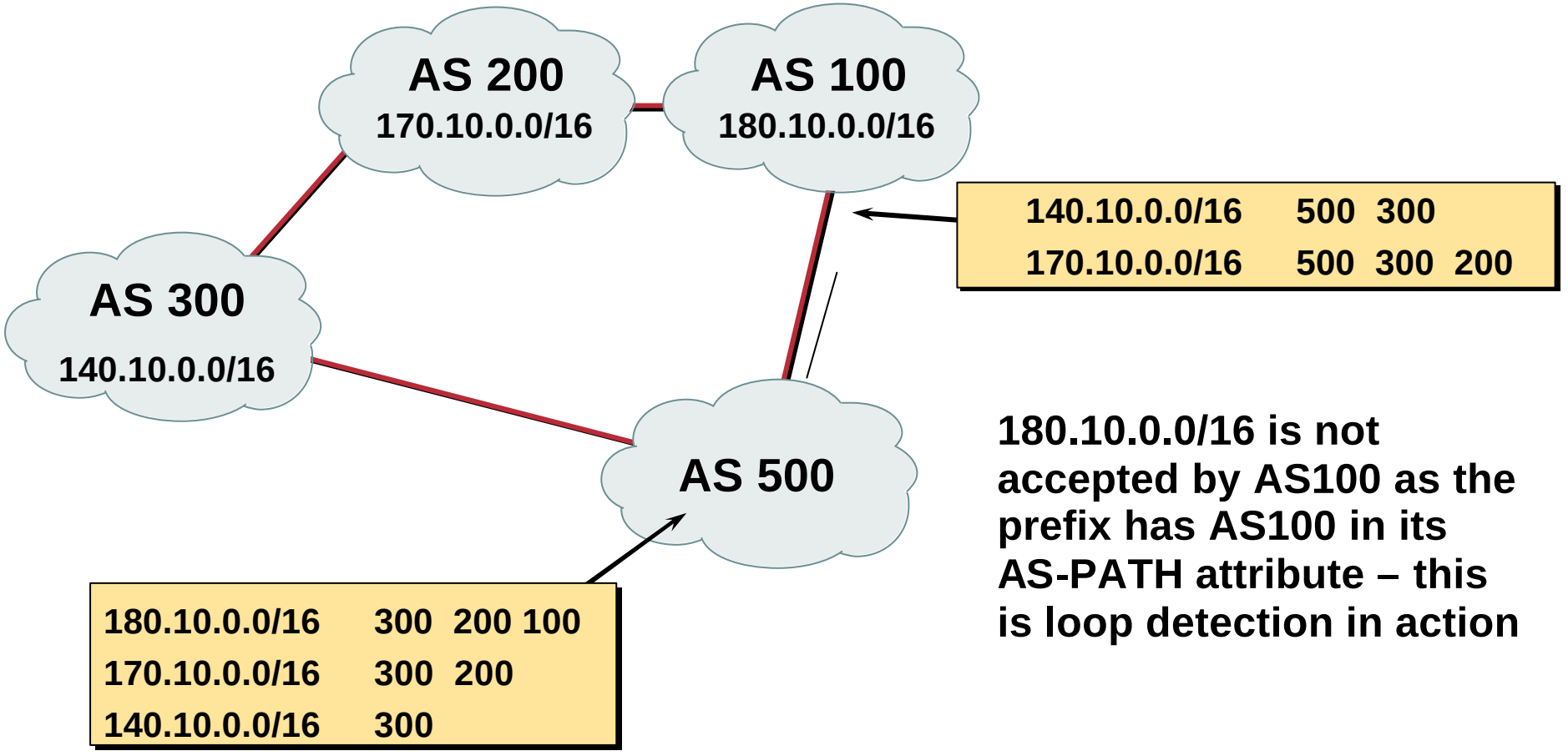
Recap

AS-Path

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

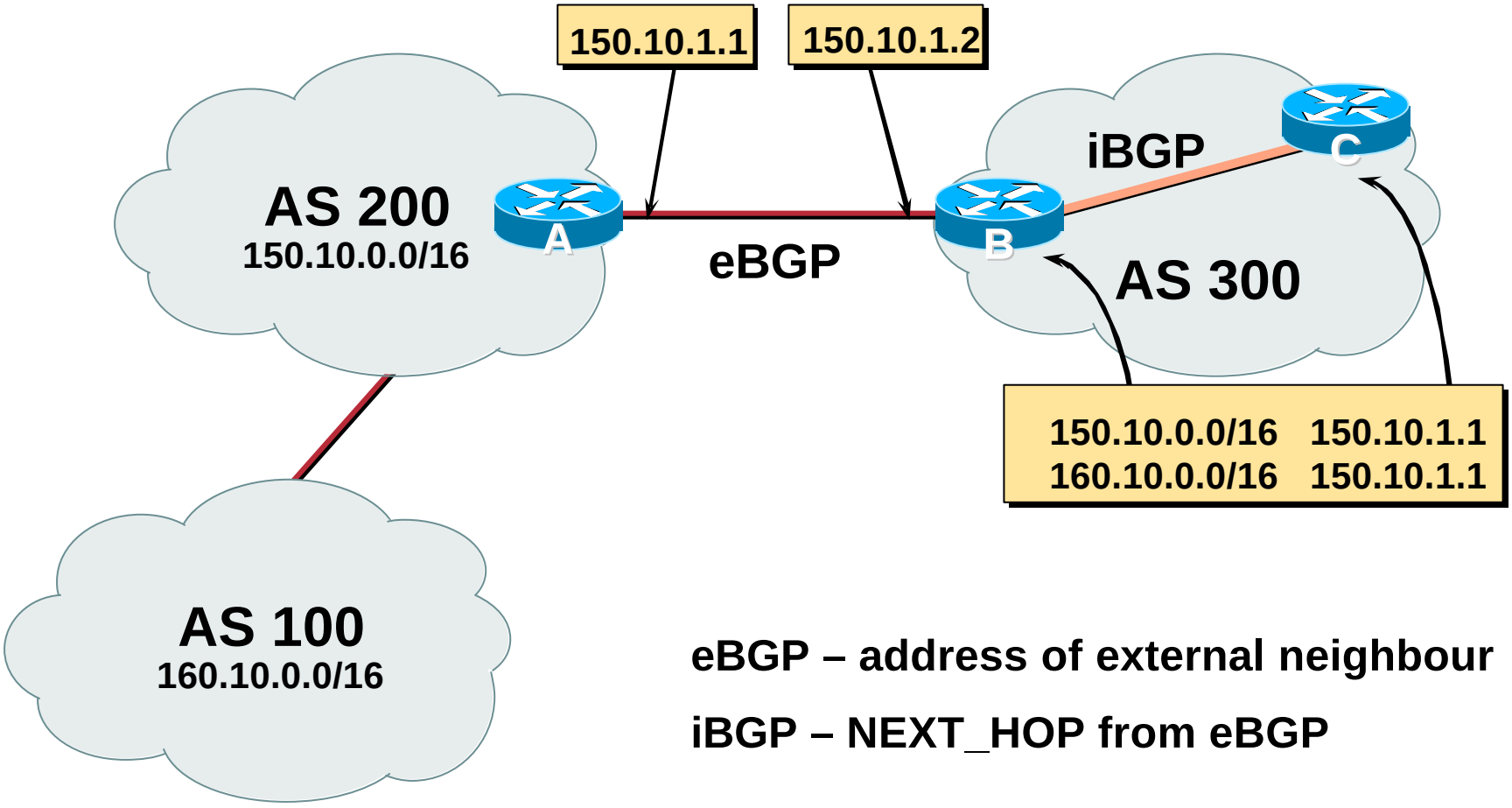


AS-Path loop detection



180.10.0.0/16 is not accepted by AS100 as the prefix has AS100 in its AS-PATH attribute – this is loop detection in action

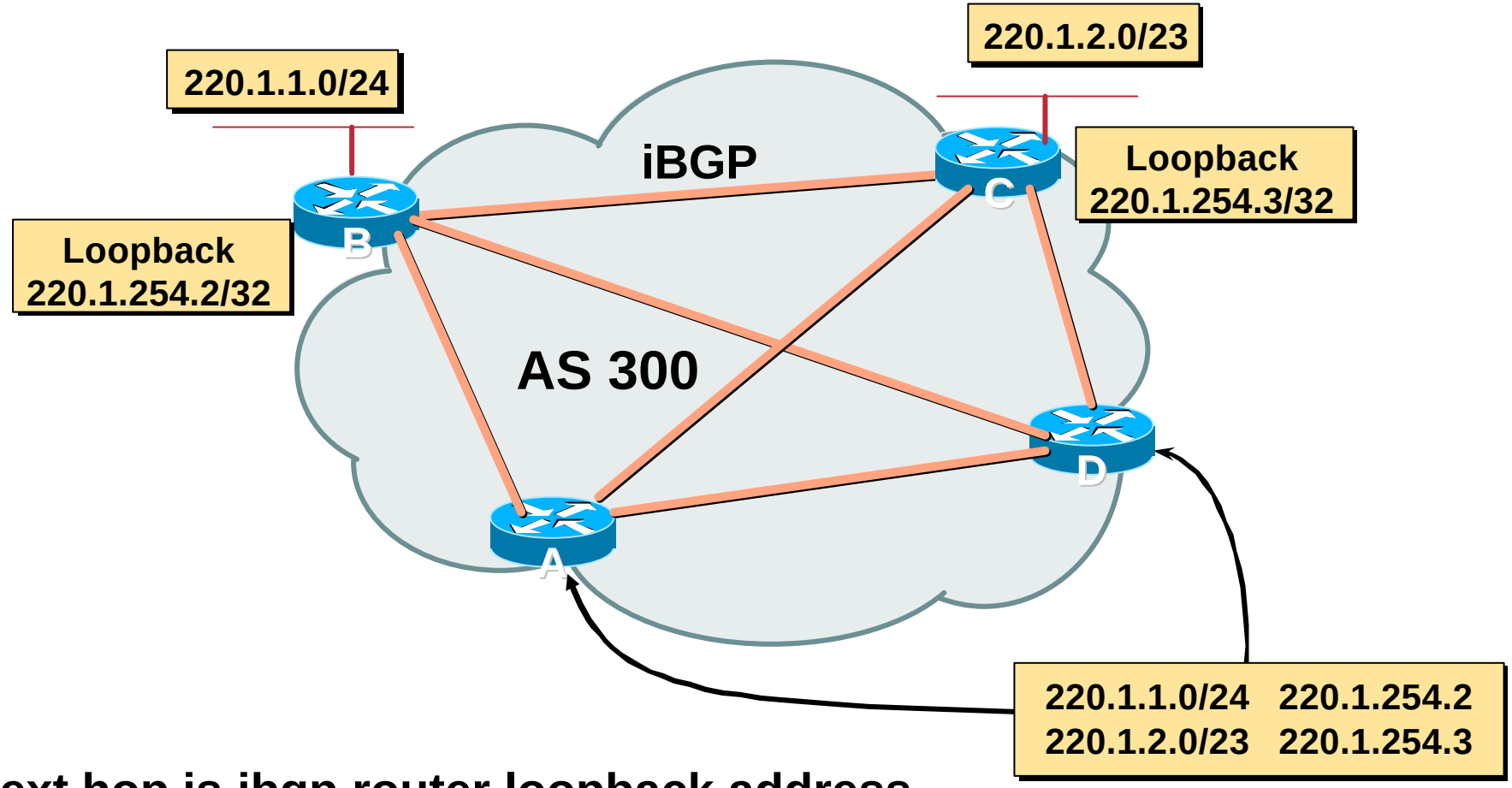
Next Hop



eBGP – address of external neighbour

iBGP – NEXT_HOP from eBGP

iBGP Next Hop



Next hop is ibgp router loopback address

Recursive route look-up

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

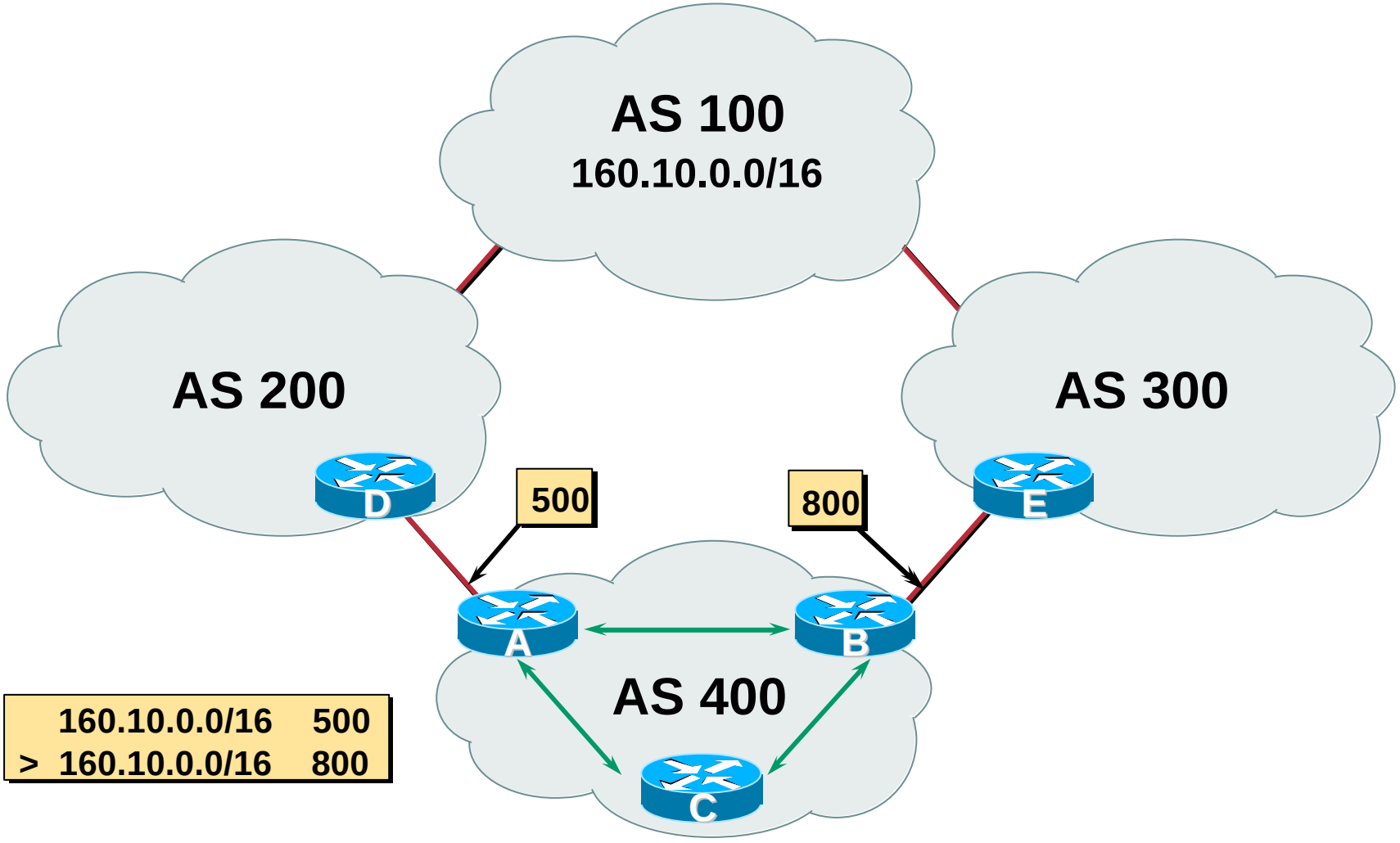
  EGP – generated by EGP

  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



Local Preference

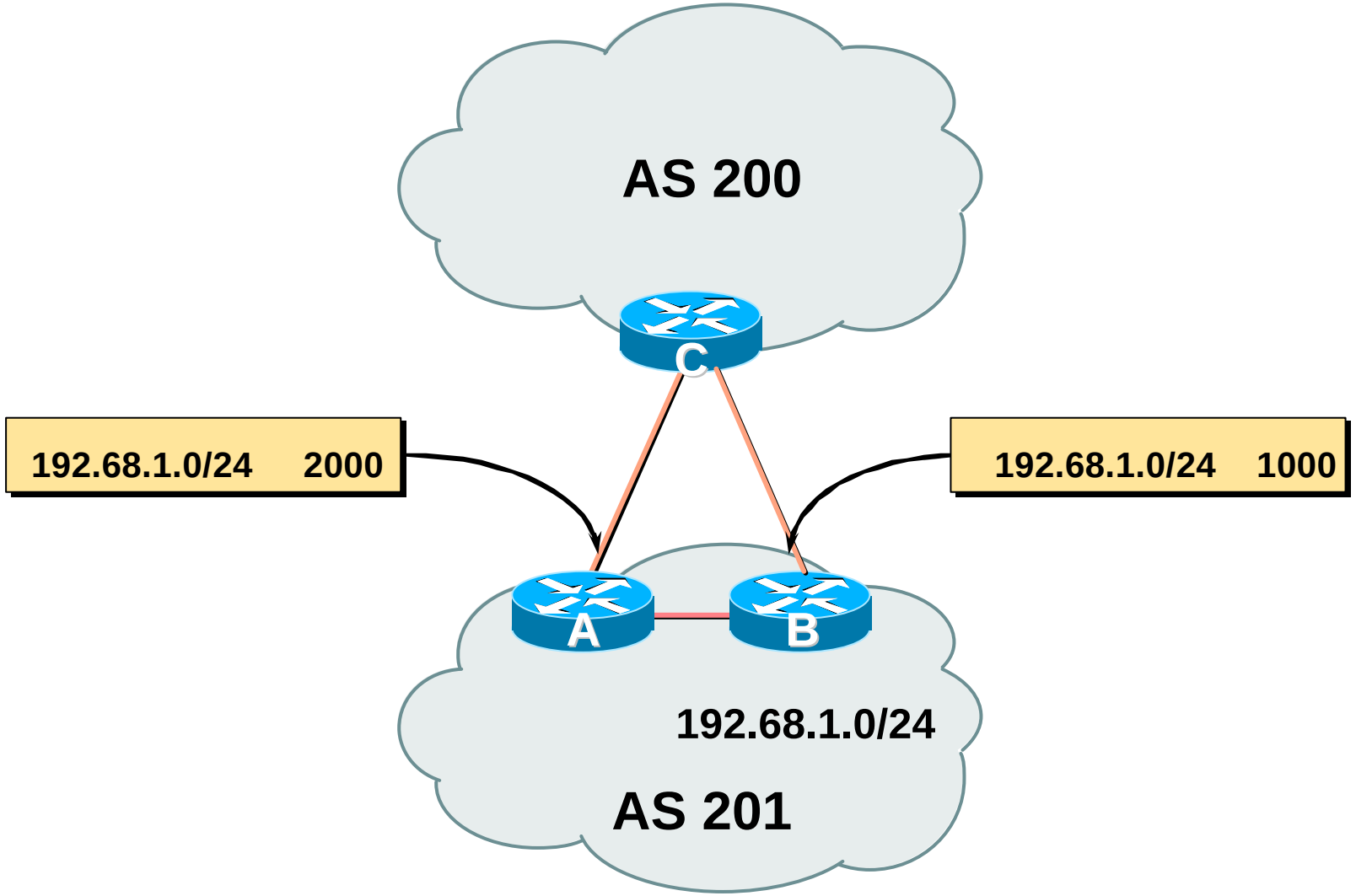
- **Local to an AS – non-transitive**
  - **Default local preference is 100 (IOS)**
- **Used to influence BGP path selection**
  - **determines best path for *outbound* traffic**
- **Path with highest local preference wins**

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

- **Not really an attribute – local to router**

  Allows policy control, similar to local preference

- **Highest weight wins**

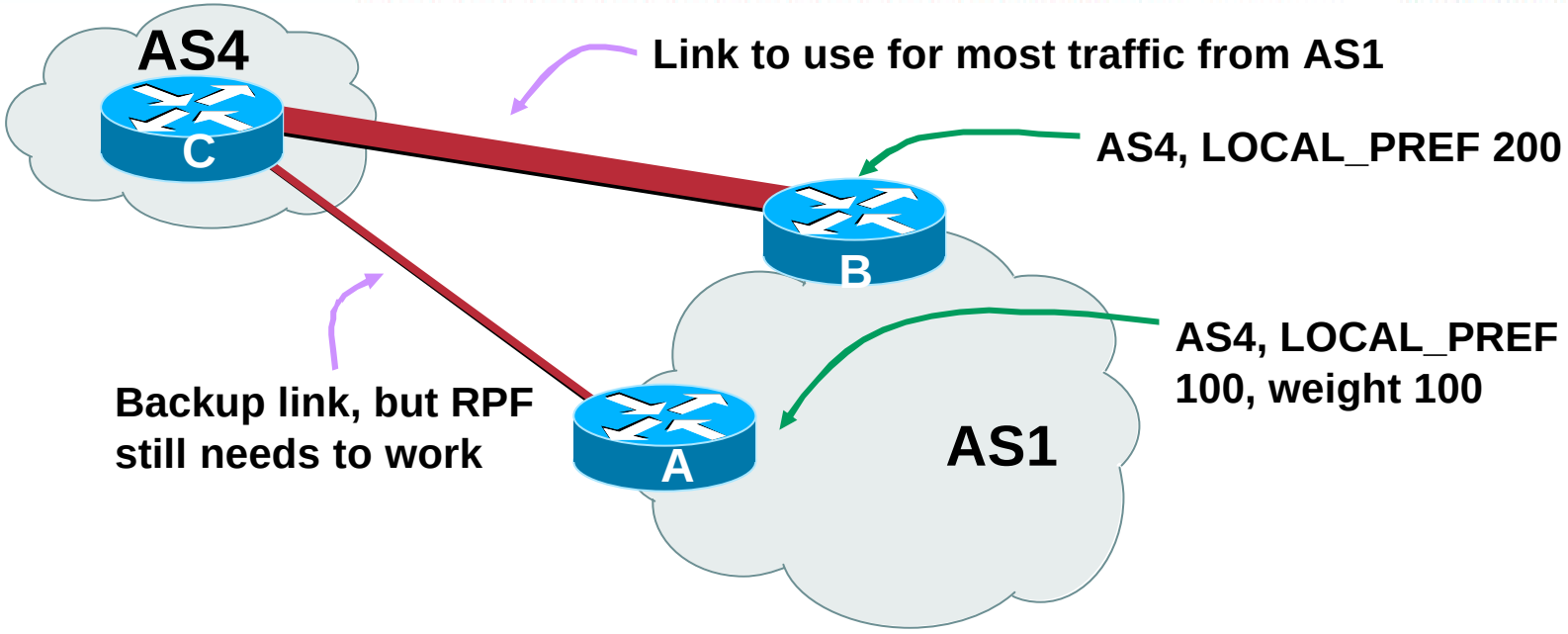
- **Applied to all routes from a neighbour**

```
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 help Deploy RPF

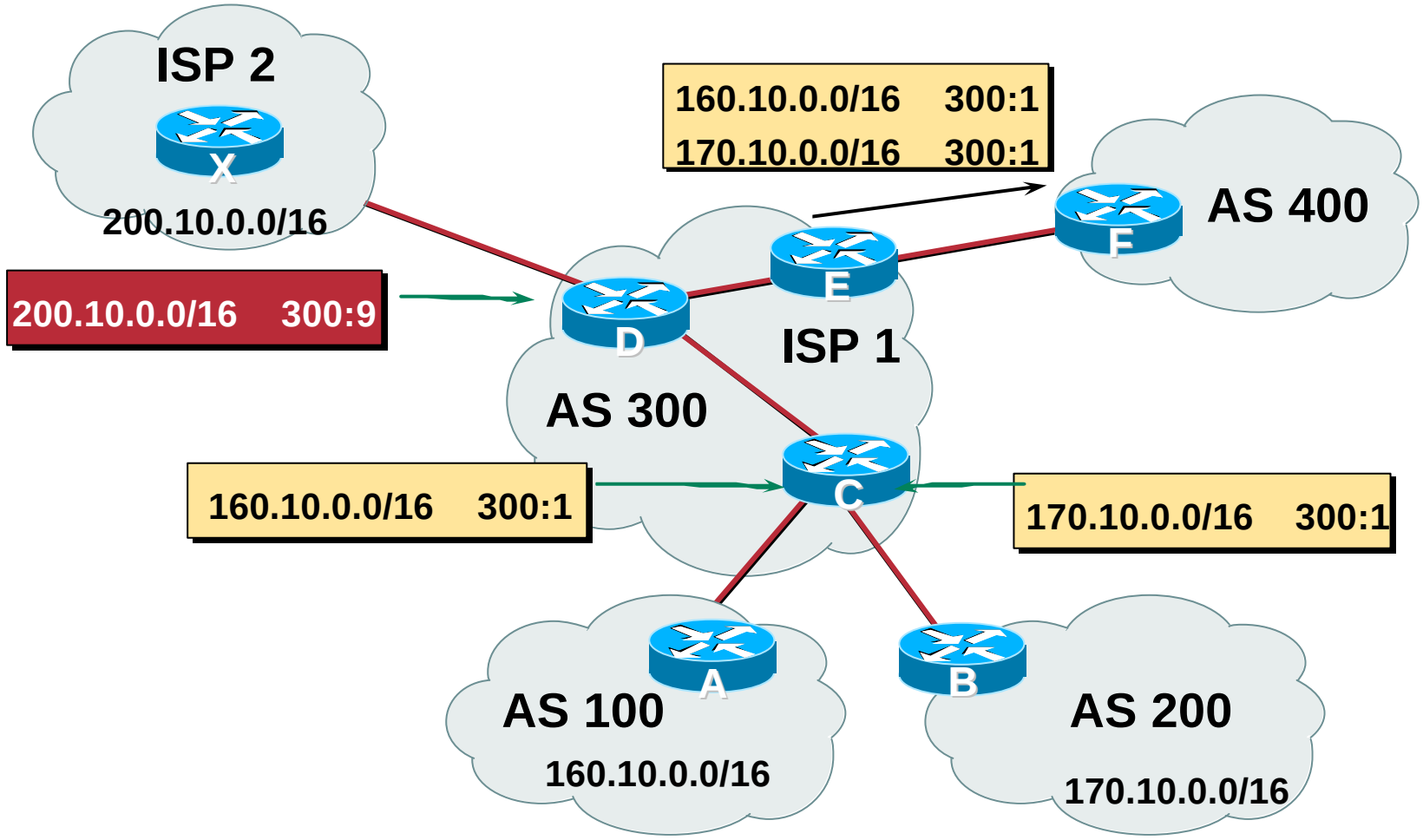


- Best path to AS4 from AS1 is always via B due to local-pref
- But packets arriving at A from AS4 over the direct C to A link will pass the RPF check as that path has a priority due to the weight being set
  - If weight was not set, best path would be via B, and the RPF check would fail

Community

- Communities are described in RFC1997
- 32 bit integer
  - 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



Well-Known Communities

Cisco.com

- **no-export**

  do not advertise to eBGP peers

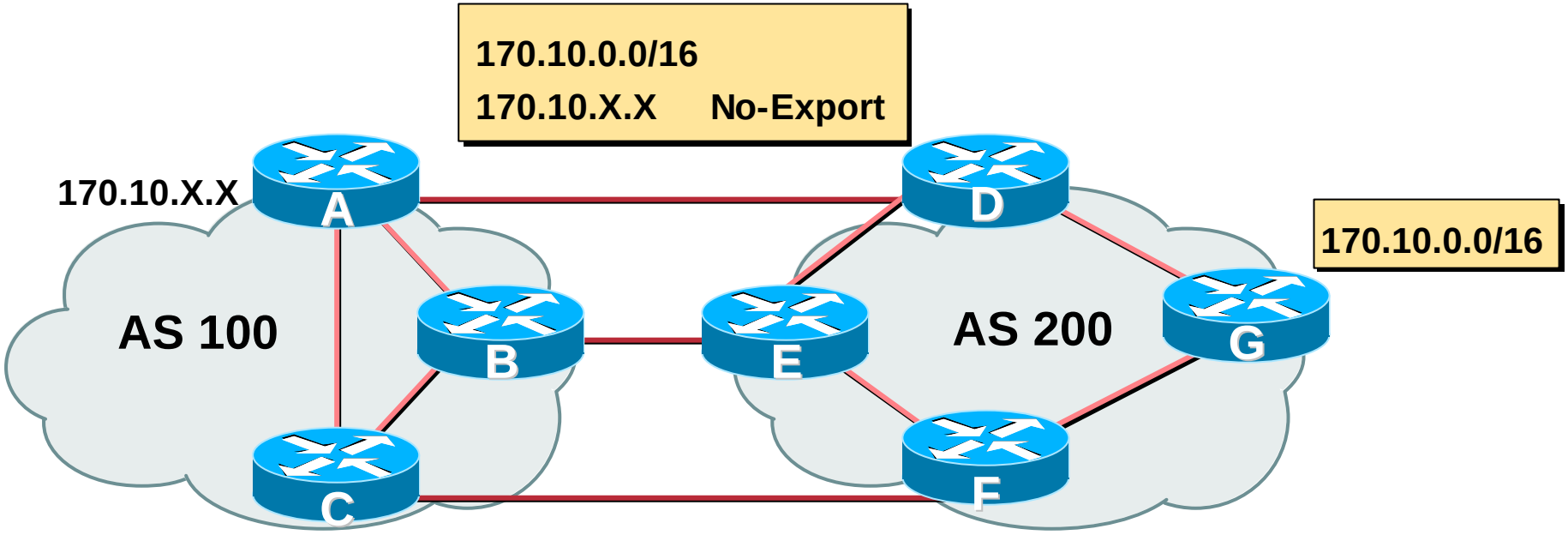
- **no-advertise**

  do not advertise to any peer

- **local-AS**

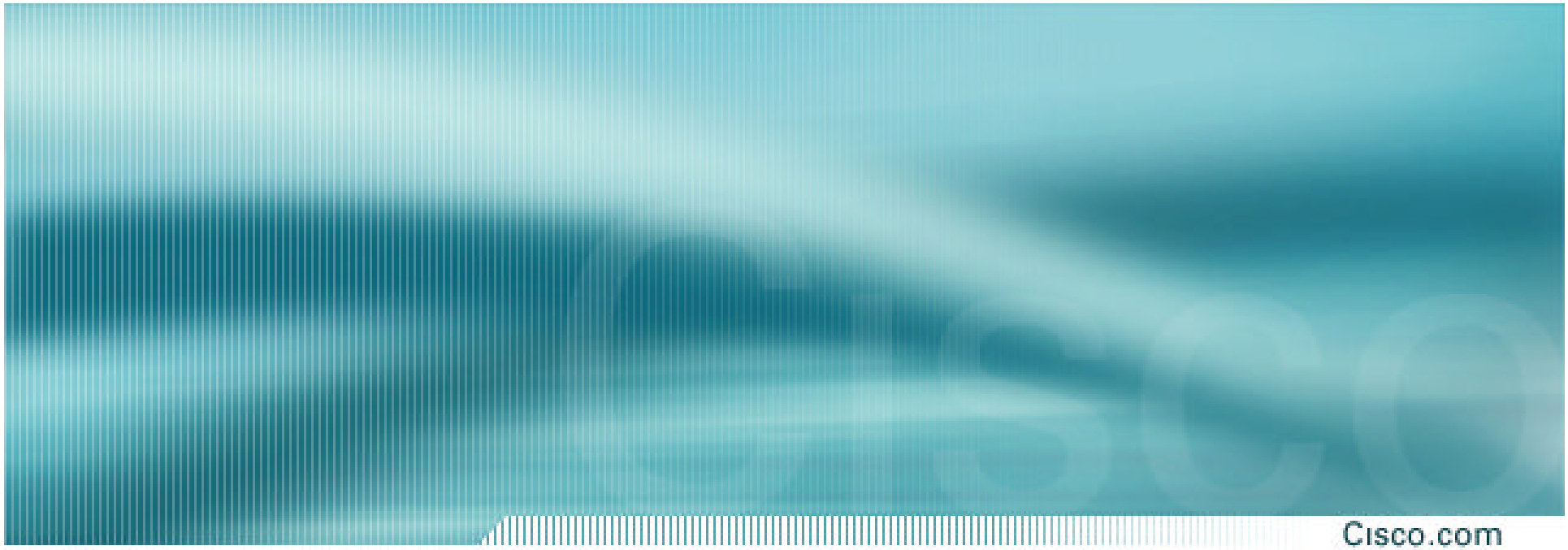
  do not advertise outside local AS (only used with confederations)

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 does not announce 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

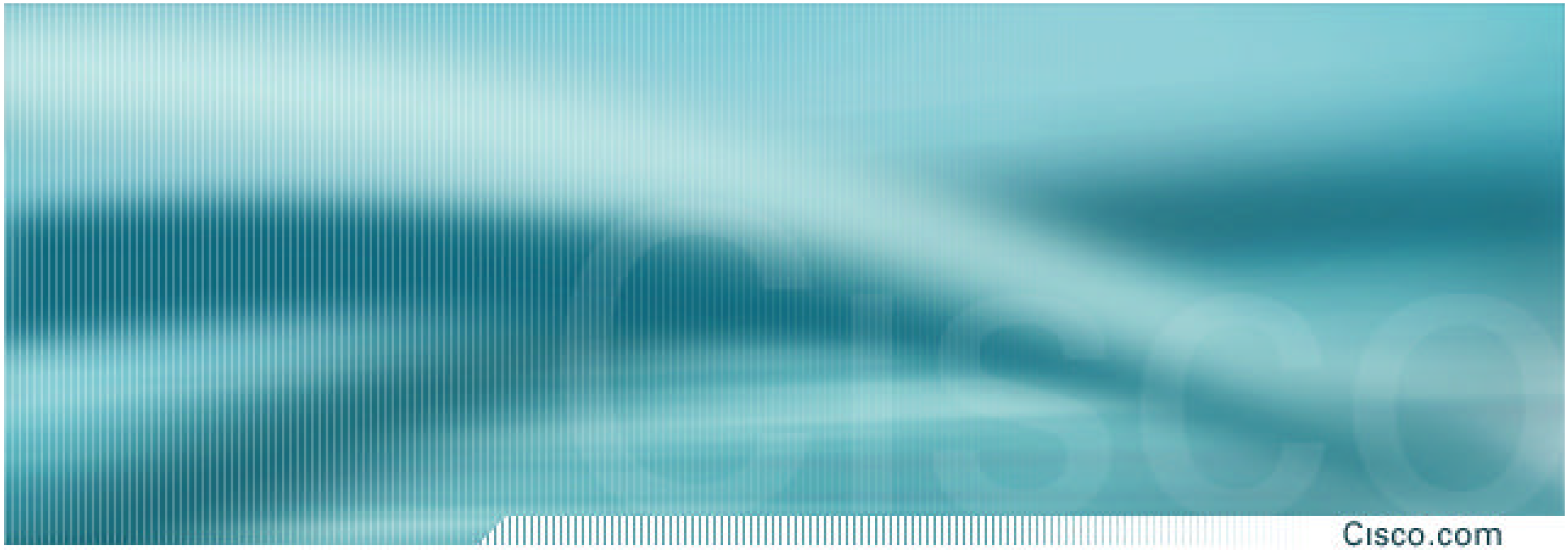
  If bgp always-compare-med, then compare for all paths

  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

Cisco.com

- **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

Cisco.com

- **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 – Regular Expressions

- **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

- 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
```

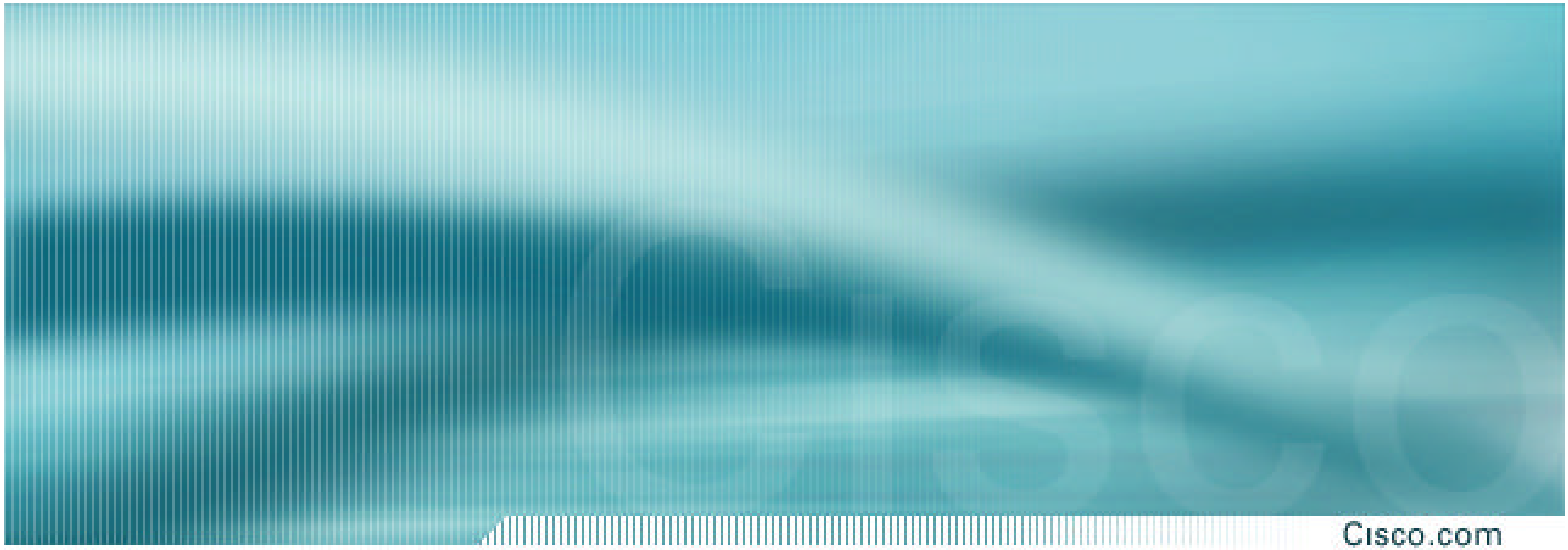
- **Use your own AS number when prepending**

  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
```



BGP Capabilities

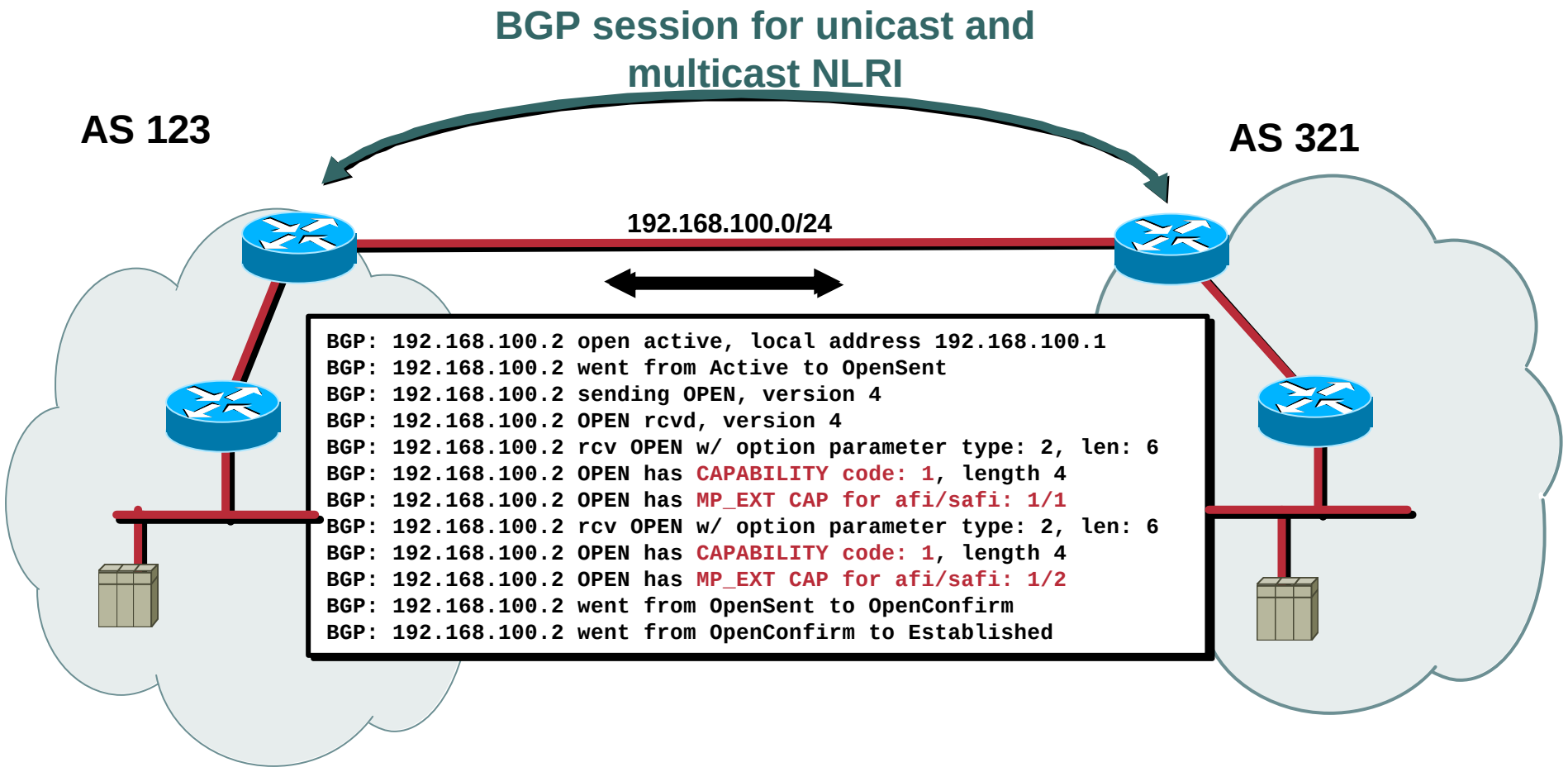
Extending BGP

BGP Capabilities

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

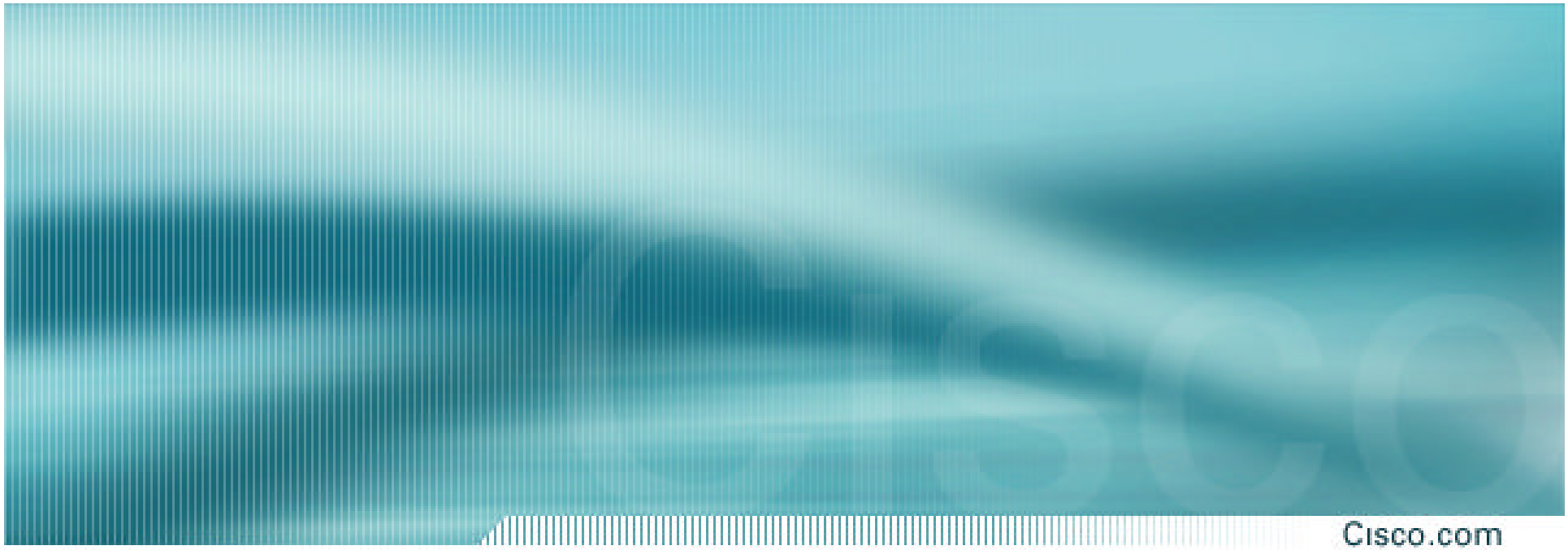
```
  0  Reserved                                        [RFC2842]
  1  Multiprotocol Extensions for BGP-4              [RFC2858]
  2  Route Refresh Capability for BGP-4              [RFC2918]
  3  Cooperative Route Filtering Capability          []
  4  Multiple routes to a destination capability   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

BGP Capabilities Negotiation



BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Using Communities
- Deploying BGP in an ISP network



BGP Scaling Techniques

BGP Scaling Techniques

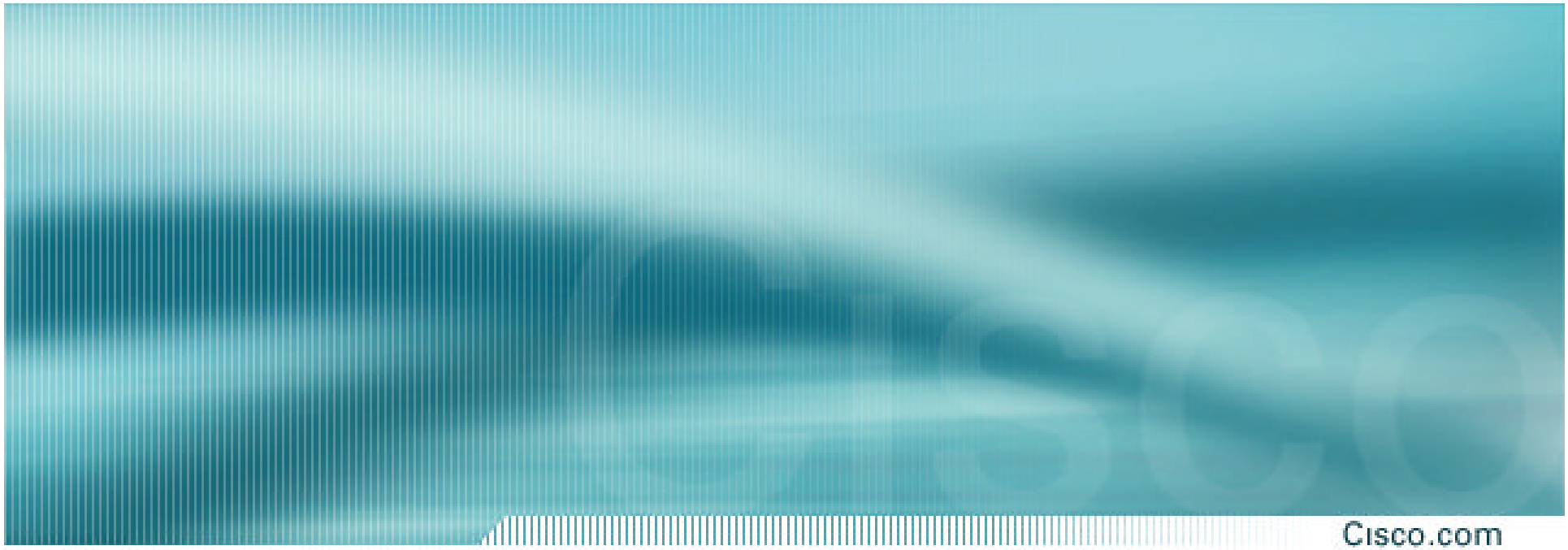
Cisco.com

- **How does a service provider:**
  - **Scale the iBGP mesh beyond a few peers?**
  - **Implement new policy without causing flaps and route churning?**
  - **Reduce the overhead on the routers?**
  - **Keep the network stable, scalable, as well as simple?**

BGP Scaling Techniques

Cisco.com

- Route Refresh
- Peer groups
- Route flap damping
- Route Reflectors & Confederations



Route Refresh

Route Refresh

Problem:

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

Solution:

- **Route Refresh**

Route Refresh Capability

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

Dynamic Reconfiguration

- **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 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

Configuring Soft Reconfiguration

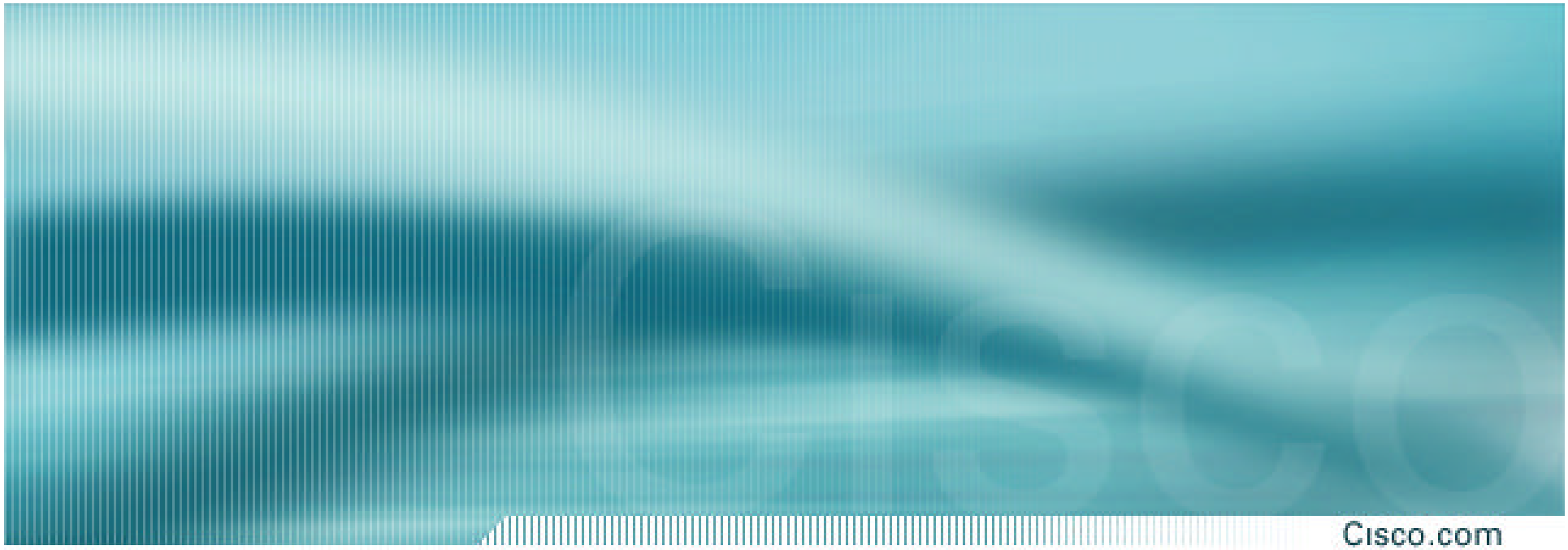
Cisco.com

```
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]
```



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

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* !

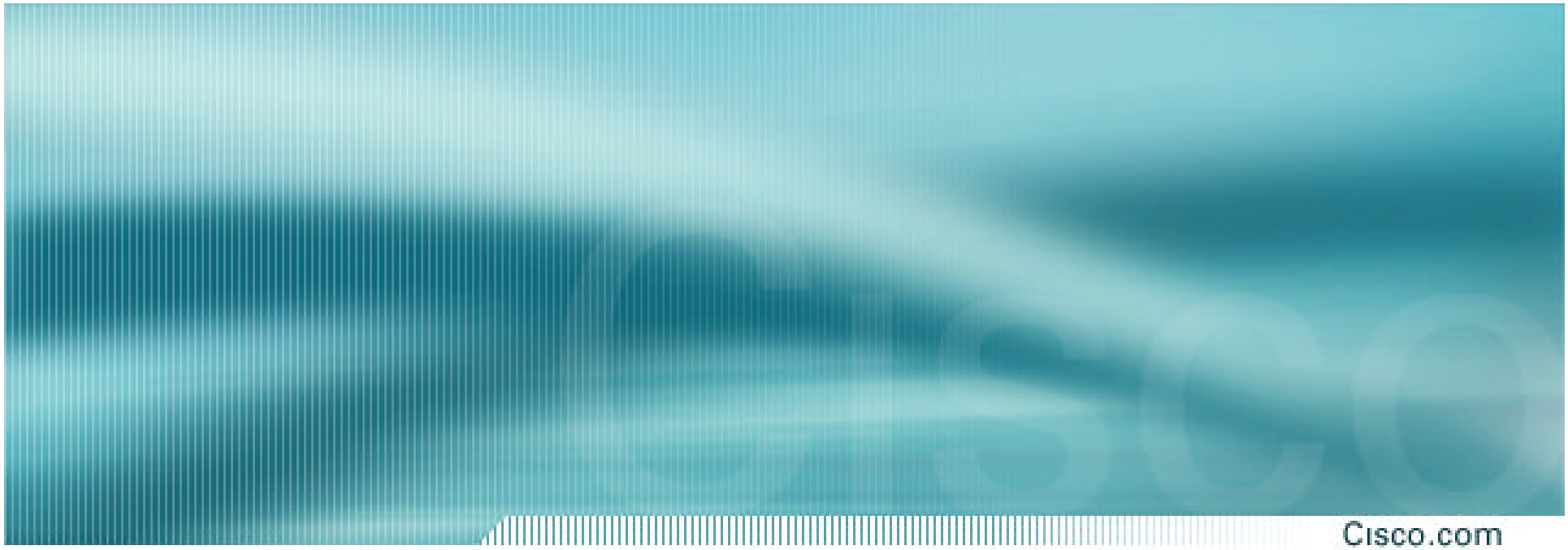
Configuring Peer Group

Cisco.com

```
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**
  - **Makes template configuration much easier**
- **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 peering reset is NOT a flap**
  - **Ripples through the entire Internet**
  - **Wastes CPU**
- **Damping aims to reduce scope of route flap propagation**

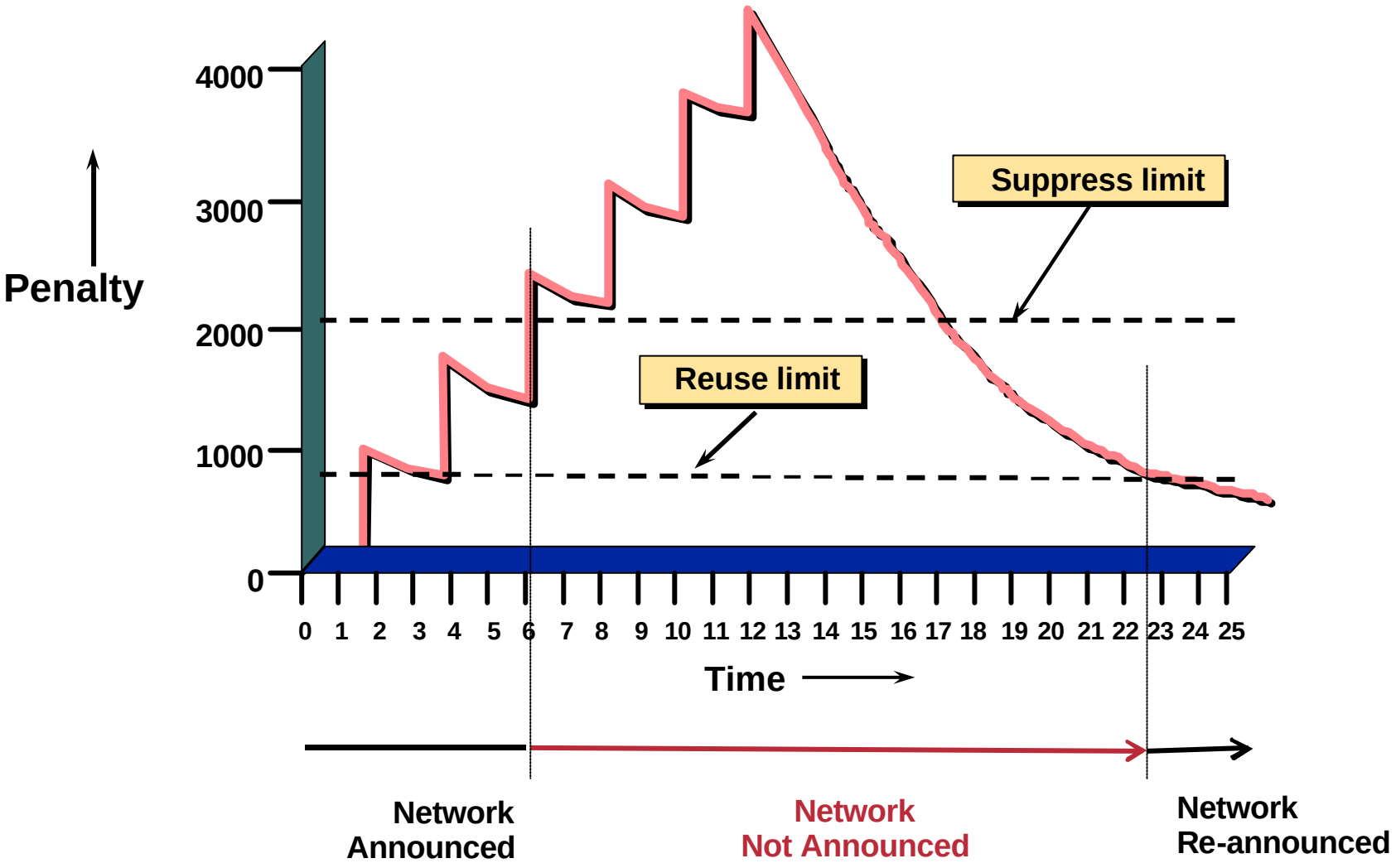
Route Flap Damping (continued)

- **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



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-229.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

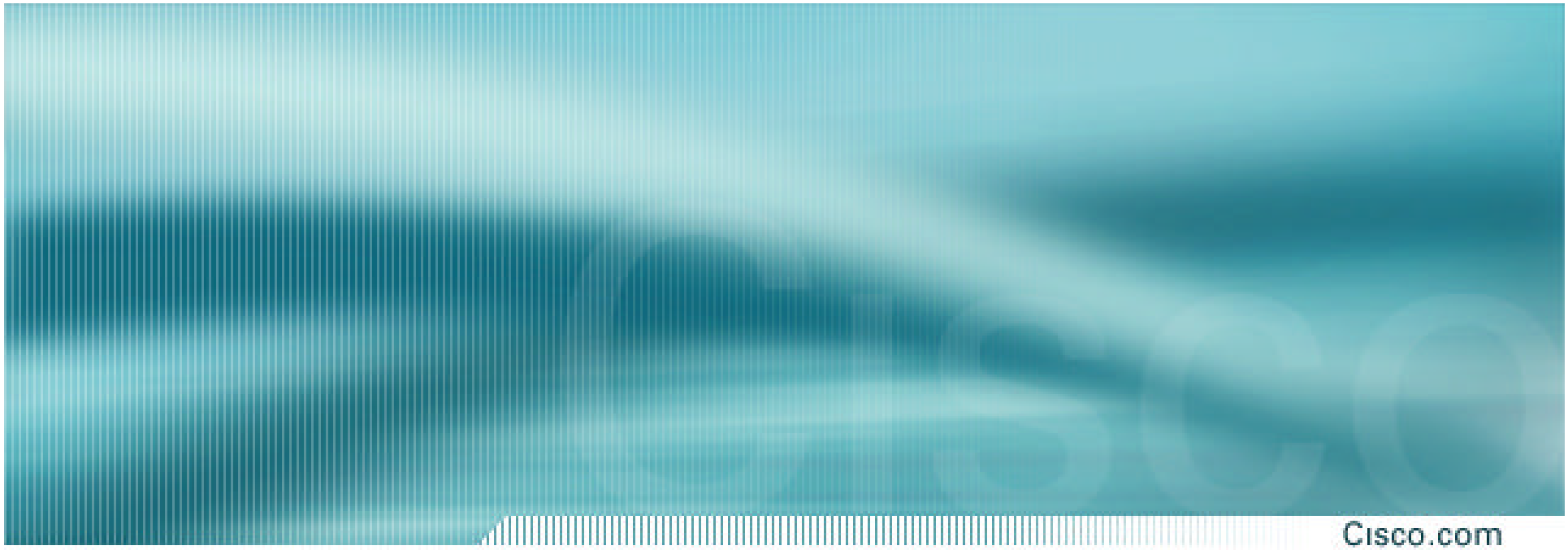
  reuse-limit of 2000 means maximum possible penalty is 8000 – suppress limit is easily reached

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**

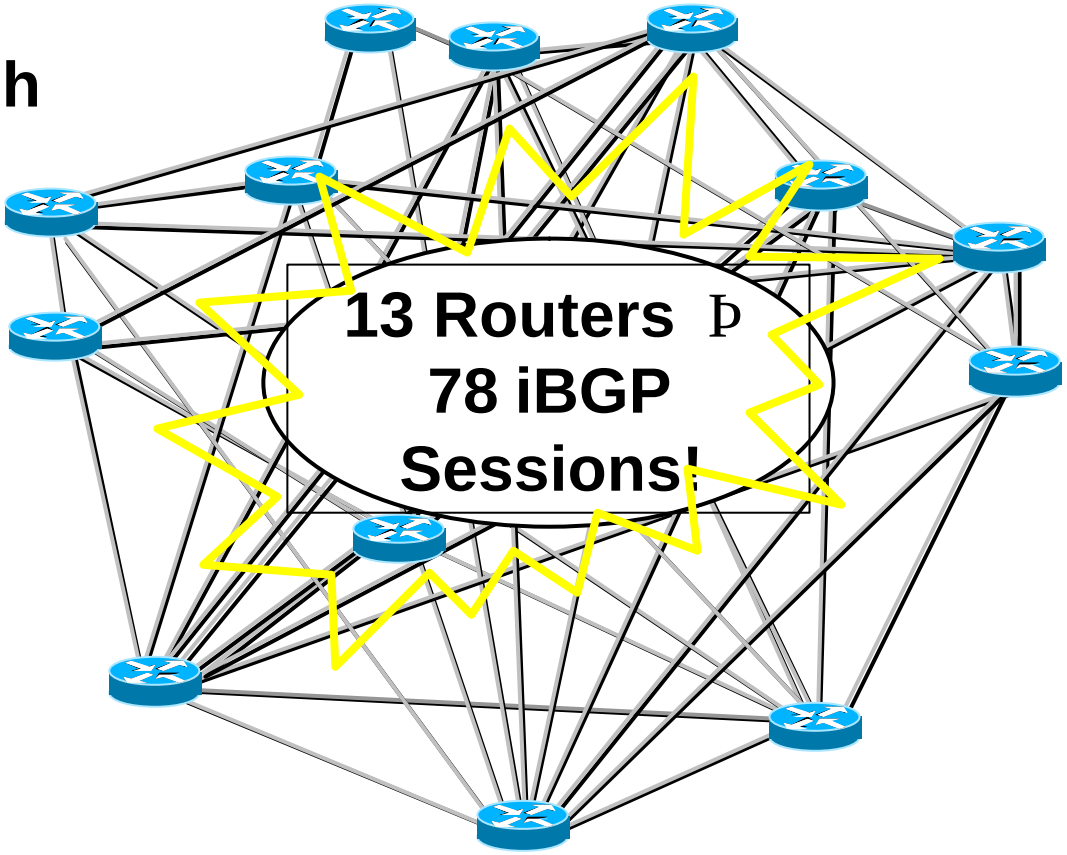


Route Reflectors and Confederations

Scaling iBGP mesh

Avoid ½n(n-1) iBGP mesh

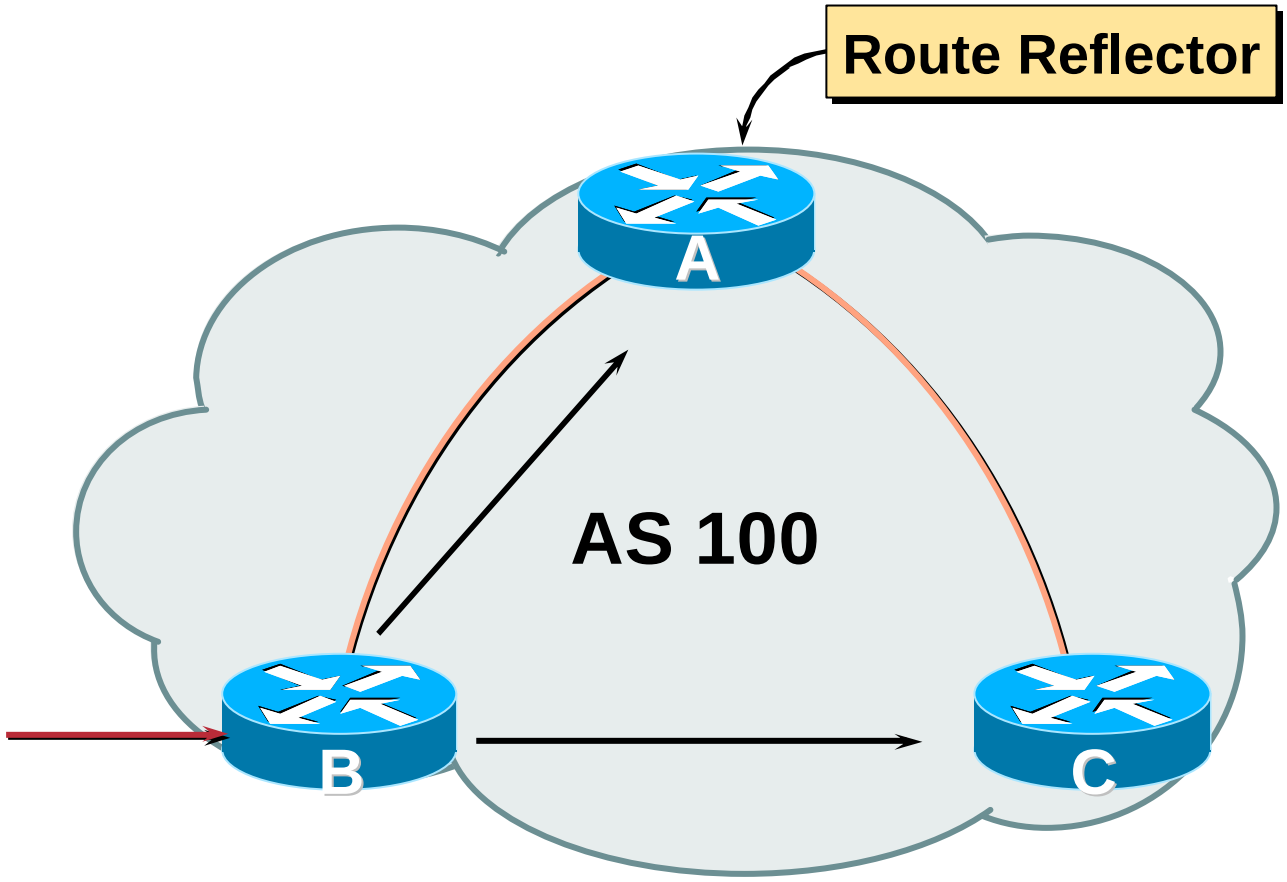
n=1000 Þ nearly half a million ibgp sessions!



Two solutions

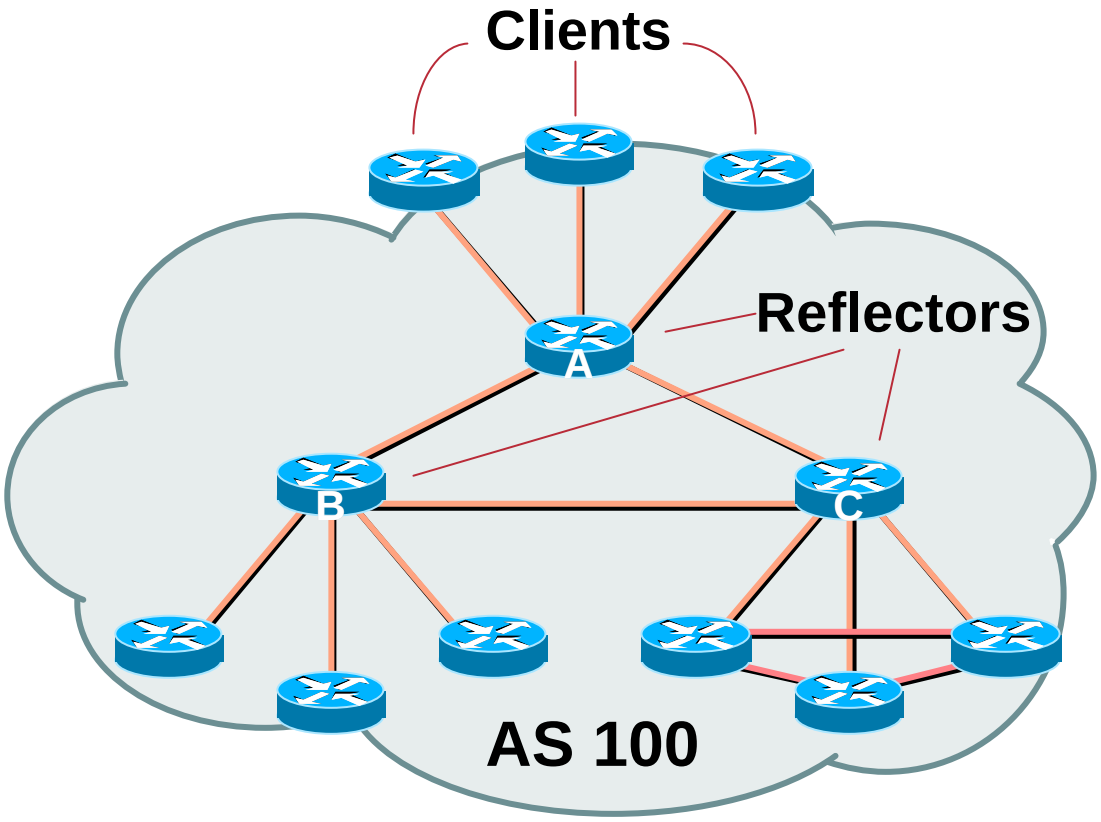
- **Route reflector – simpler to deploy and run**
- **Confederation – more complex, corner case benefits**

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

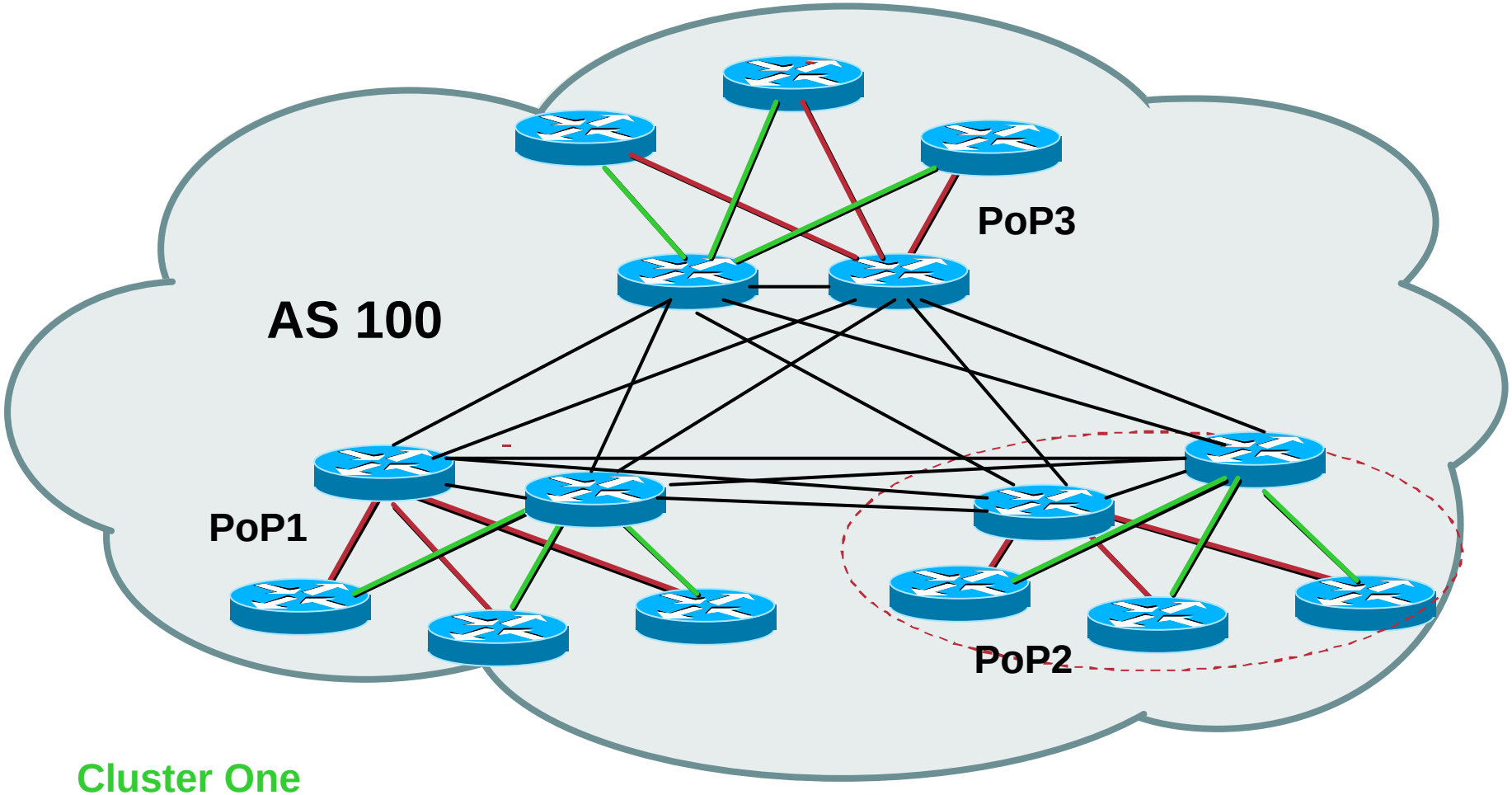
  Cluster-id is automatically set from router-id (address of loopback)

  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

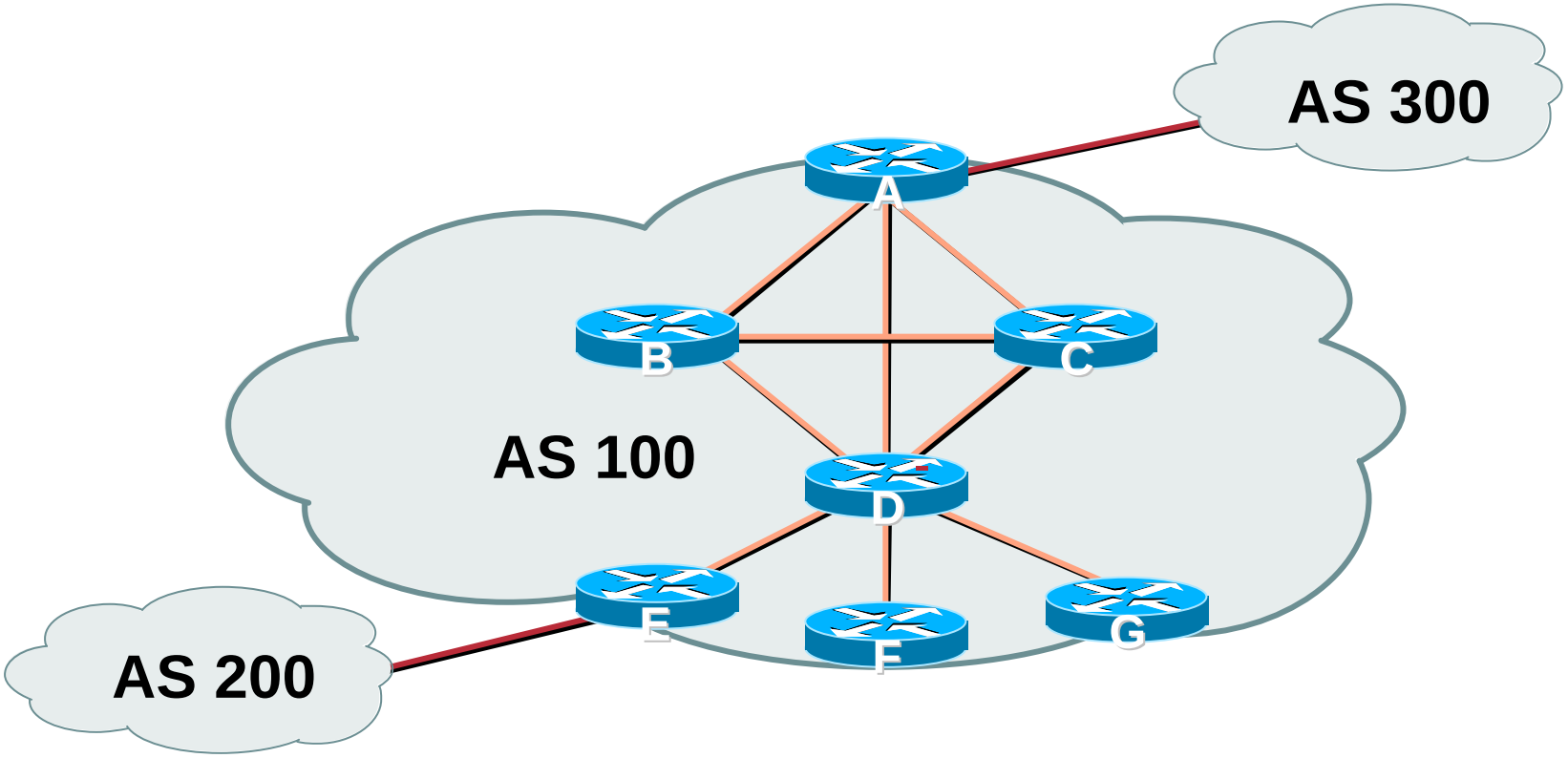
  Core routers are RR for the PoP

  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

Cisco.com

```
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
 neighbor 4.4.4.4 remote-as 100
 neighbor 4.4.4.4 route-reflector-client
```

Confederations

- **Divide the AS into sub-ASes**
  - **eBGP between sub-ASes, 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"**

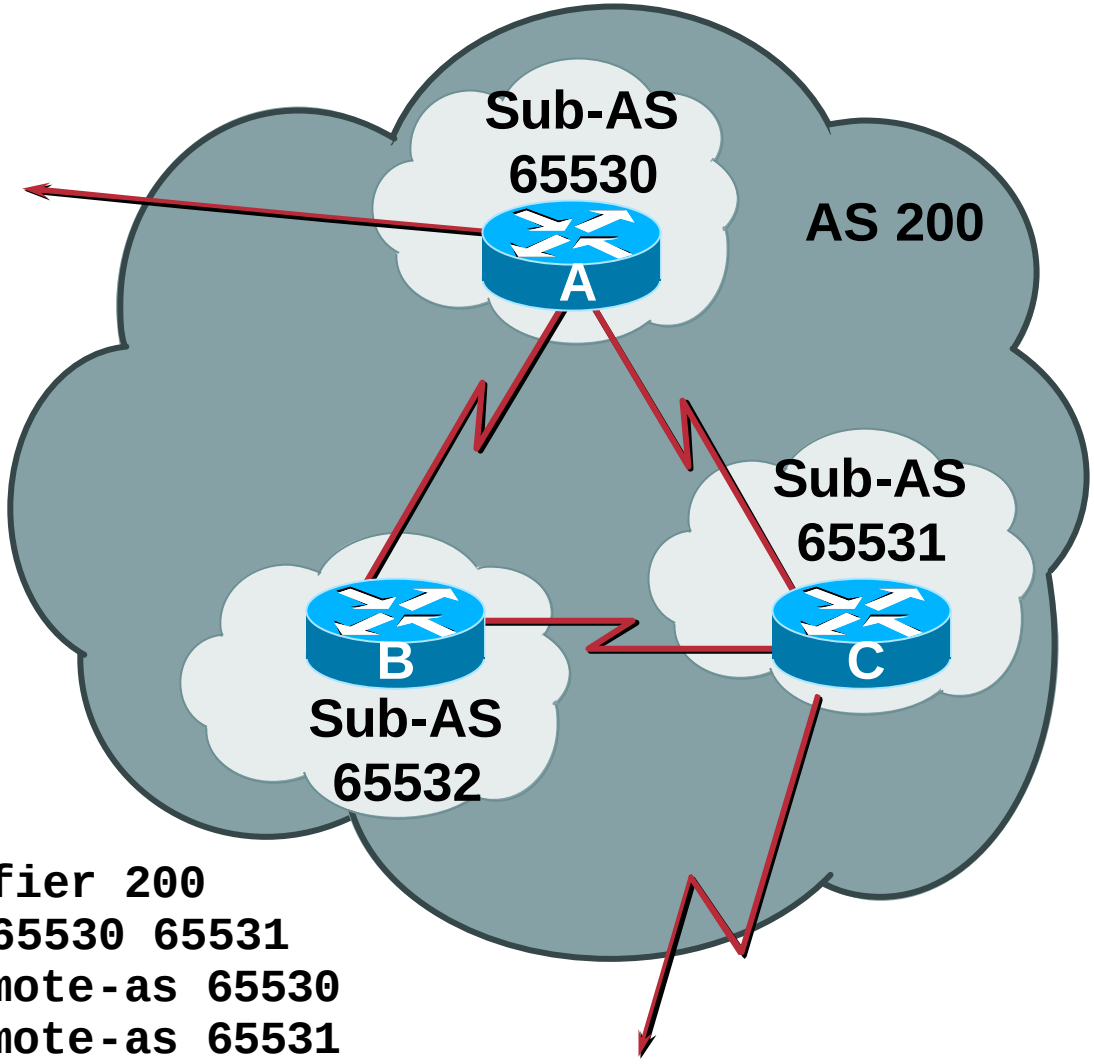
  Each sub-AS uses a number from the private AS range (64512-65534)

- **iBGP speakers in each 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

  Can also use Route-Reflector within sub-AS

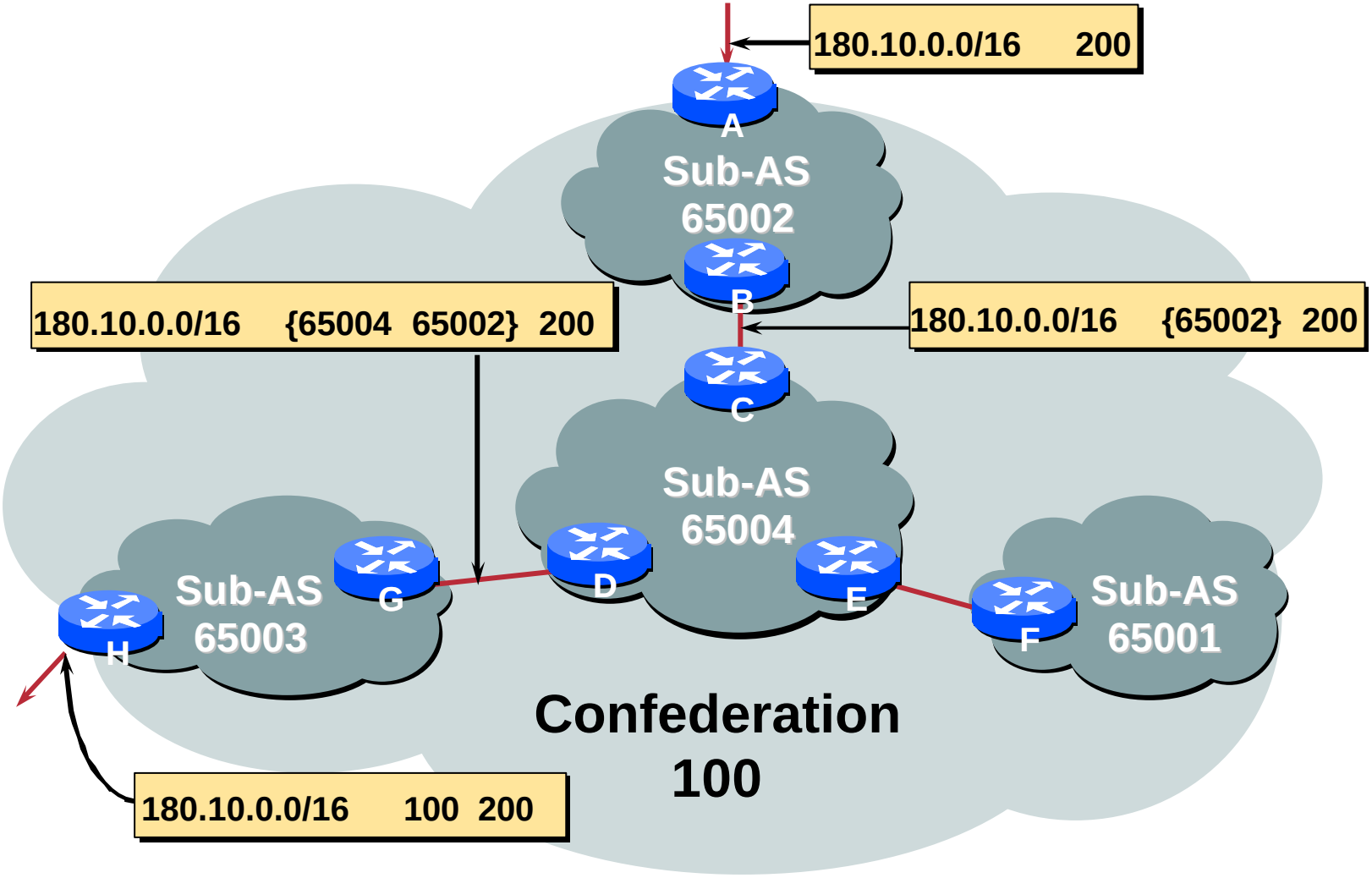
Confederations (cont.)



- **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
```

Confederations: AS-Sequence



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
```

Route Reflectors 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	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**

  Or can use local-as feature to do a similar thing

- **Can use route-reflectors with confederation sub-AS to reduce the sub-AS iBGP mesh**

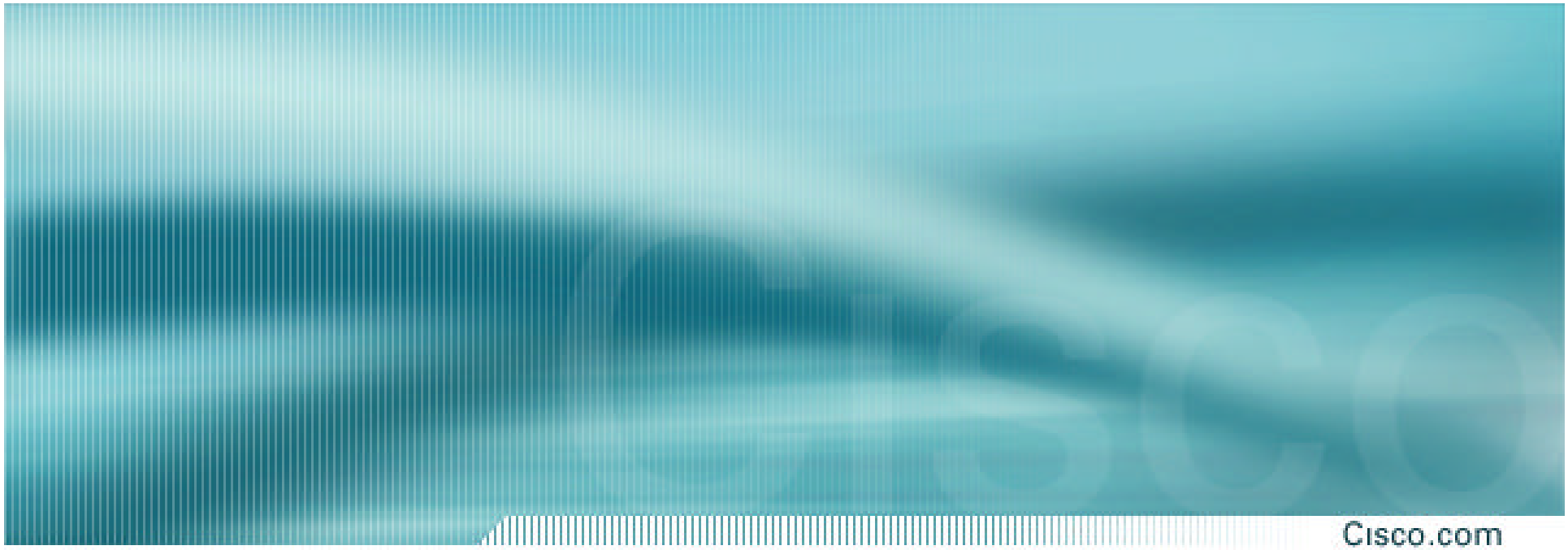
BGP Scaling Techniques

Cisco.com

- **These 4 techniques should be core requirements in all ISP networks**
  - **Route Refresh**
  - **Peer groups**
  - **Route flap damping**
  - **Route reflectors**

BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Using Communities
- Deploying BGP in an ISP network



Service Providers use of Communities

Some examples of how ISPs make life easier for themselves

BGP Communities

- Another ISP “scaling technique”
- Prefixes are grouped into different “classes” or communities within the ISP network
- Each community means a different thing, has a different result in the ISP network

BGP Communities

- **Communities are generally set at the edge of the ISP network**

  Customer edge: customer prefixes belong to different communities depending on the services they have purchased

  Internet edge: transit provider prefixes belong to difference communities, depending on the loadsharing or traffic engineering requirements of the local ISP, or what the demands from its BGP customers might be

- **Two simple examples follow to explain the concept**

Community Example – Customer Edge

- This demonstrates how communities might be used at the customer edge of an ISP network
- ISP has three connections to the Internet:
  - IXP connection, for local peers
  - Private peering with a competing ISP in the region
  - Transit provider, who provides visibility to the entire Internet
- Customers have the option of purchasing combinations of the above connections

Community Example – Customer Edge

- **Community assignments:**

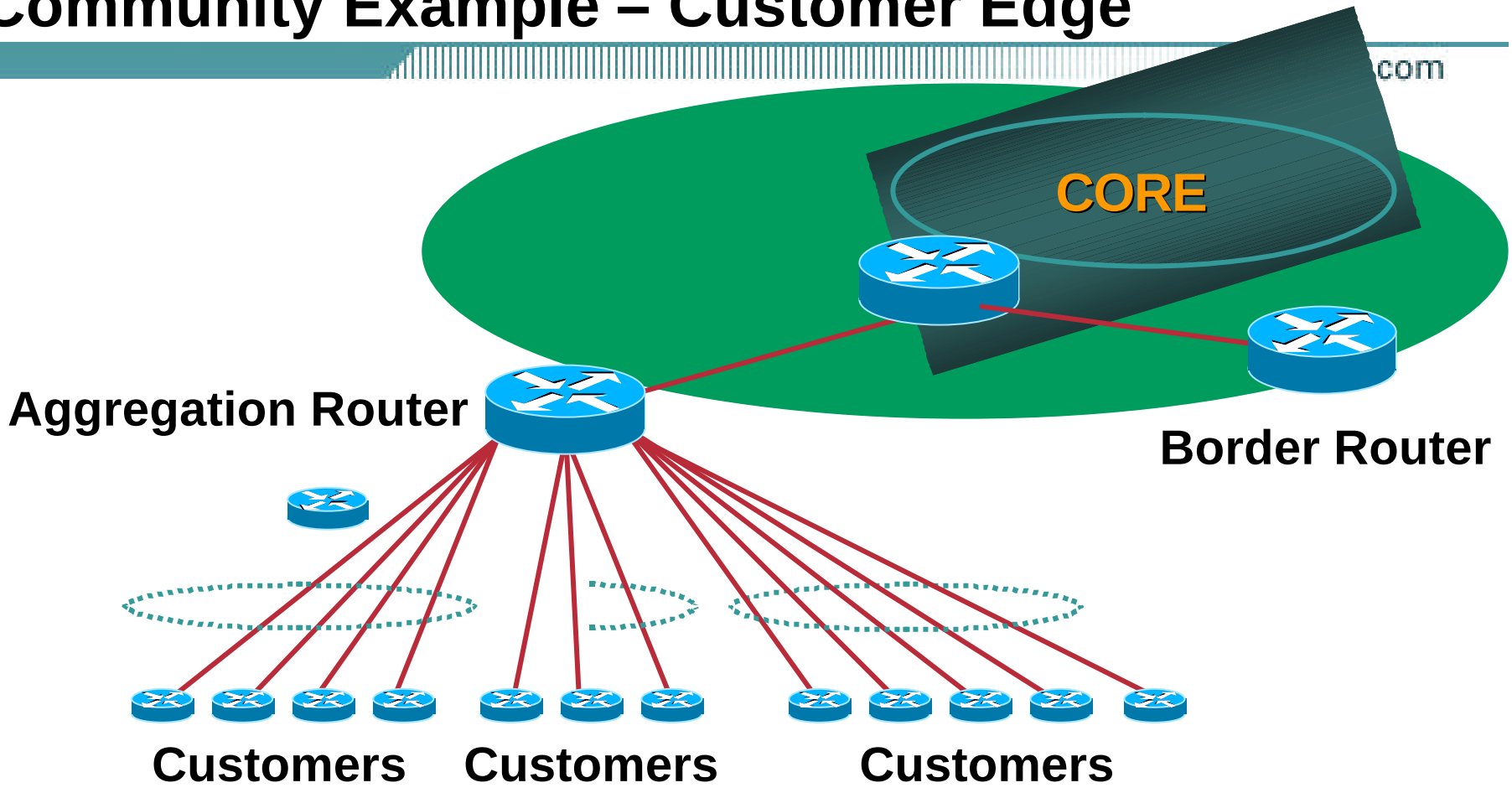
  IXP connection: community 100:2100

  Private peer: community 100:2200

- **Customer who buys local connectivity (via IXP) is put in community 100:2100**
- **Customer who buys peer connectivity is put in community 100:2200**
- **Customer who wants both IXP and peer connectivity is put in 100:2100 and 100:2200**
- **Customer who wants “the Internet” has no community set**

  We are going to announce his prefix everywhere

Community Example – Customer Edge

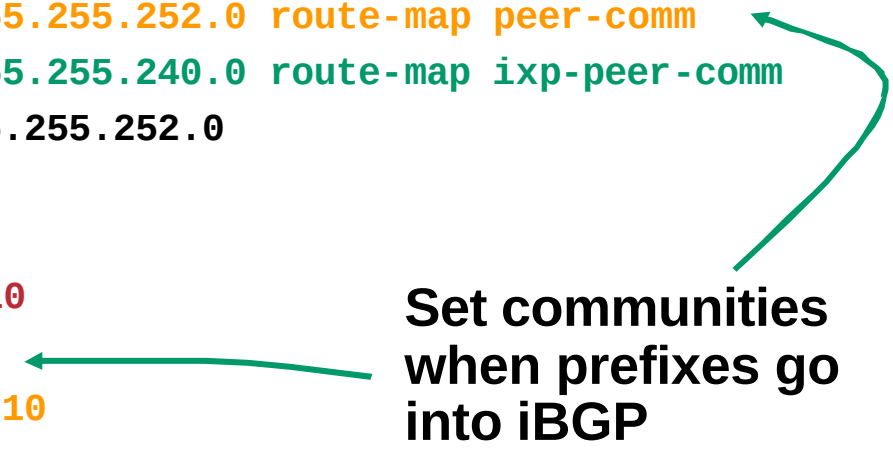


Communities set at the aggregation router where the prefix is injected into the ISP's iBGP

Community Example – Customer Edge

Aggregation Router configuration

```
ip route 222.1.20.0 255.255.255.0 serial 0 ! IXP only
ip route 222.1.28.0 255.255.252.0 serial 1 ! Peer only
ip route 222.1.64.0 255.255.240.0 serial 3 ! IXP+Peer
ip route 222.1.0.0  255.255.252.0 serial 4 ! everything
!
router bgp 100
 network 222.1.20.0 mask 255.255.255.0 route-map ixp-comm
 network 222.1.28.0 mask 255.255.252.0 route-map peer-comm
 network 222.1.64.0 mask 255.255.240.0 route-map ixp-peer-comm
 network 222.1.0.0 mask 255.255.252.0
 neighbor ...
!
route-map ixp-comm permit 10
 set community 100:2100
route-map peer-comm permit 10
 set community 100:2200
route-map ixp-peer-comm permit 10
 set community 100:2100 100:2200
```

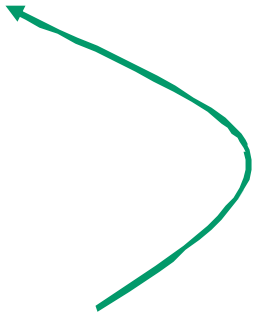


Set communities when prefixes go into iBGP

Community Example – Customer Edge

Border Router configuration

```
router bgp 100
 network 221.1.0.0 mask 255.255.0.0
 neighbor ixp-peer peer-group
 neighbor ixp-peer route-map ixp-out out
 neighbor private-peer peer-group
 neighbor private-peer route-map ppeer-out out
 neighbor upstream peer-group
 neighbor upstream prefix-list aggregate out
 neighbor ...
!
route-map ixp-out permit 10
 match community 11
route-map ppeer-out permit 10
 match community 12
!
ip community-list 11 permit 100:2100
ip community-list 12 permit 100:2200
ip prefix-list aggregate permit 221.1.0.0/16
```



Filter outgoing announcements based on communities set

Community Example – Customer Edge

- **No need to alter filters at the network border when adding a new customer**
- **New customer simply is added to the appropriate community**
  - **Border filters already in place take care of announcements**
    - **Ease of operation!**

Community Example – Internet Edge

- This demonstrates how communities might be used at the peering edge of an ISP network
- ISP has four types of BGP peers:
  - Customer
  - IXP peer
  - Private peer
  - Transit provider
- The prefixes received from each can be classified using communities
- Customers can opt to receive any or all of the above

Community Example – Internet Edge

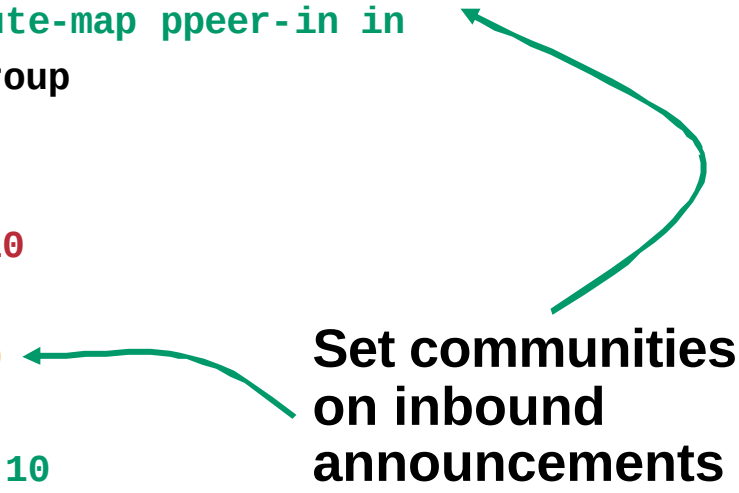
Cisco.com

- **Community assignments:**
  - **Customer prefix: community 100:3000**
  - **IXP prefix: community 100:3100**
  - **Private peer prefix: community 100:3200**
- **BGP customer who buys local connectivity gets 100:3000**
- **BGP customer who buys local and IXP connectivity receives community 100:3000 and 100:3100**
- **BGP customer who buys full peer connectivity receives community 100:3000, 100:3100, and 100:3200**
- **Customer who wants “the Internet” gets everything**
  - **Gets default route via “default-originate”**
  - **Or pays money to get all 120k prefixes**

Community Example – Internet Edge

Border Router configuration

```
router bgp 100
 neighbor customer peer-group
 neighbor customer route-map cust-in in
 neighbor ixp-peer peer-group
 neighbor ixp-peer route-map ixp-in in
 neighbor private-peer peer-group
 neighbor private-peer route-map ppeer-in in
 neighbor upstream peer-group
 neighbor ...
!
route-map cust-in permit 10
 set community 100:3000
route-map ixp-in permit 10
 set community 100:3100
route-map ppeer-in permit 10
 set community 100:3200
!
```

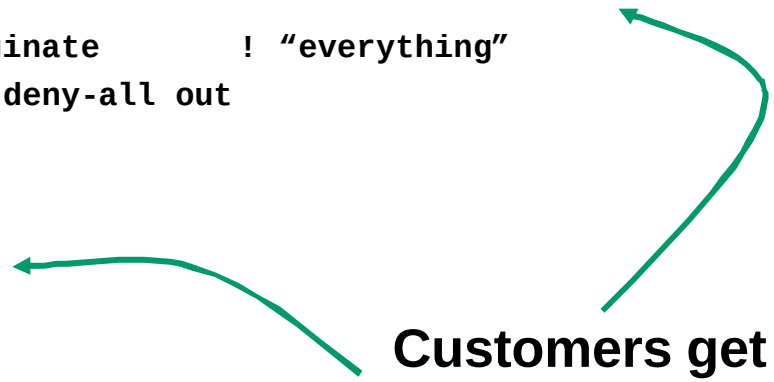


Set communities on inbound announcements

Community Example – Internet Edge

Aggregation Router configuration

```
router bgp 100
 neighbor customer1 peer-group
 neighbor customer1 route-map cust1-out     ! local routes
 neighbor customer2 peer-group
 neighbor customer2 route-map cust2-out     ! local+IXP routes
 neighbor customer3 peer-group
 neighbor customer3 route-map cust3-out     ! all routes except internet
 neighbor customer4 peer-group
 neighbor customer4 default-originate       ! “everything”
 neighbor customer4 prefix-list deny-all out
!
route-map cust1-out permit 10
 match community 23
route-map cust2-out permit 10
 match community 24
route-map cust3-out permit 10
 match community 25
!
ip community-list 23 permit 100:3000
ip community-list 24 permit 100:3000
ip community-list 24 permit 100:3100
...
```



Customers get prefixes according to community matches

Community Example – Internet Edge

- **No need to create customised filters when adding customers**
  - **Border router already sets communities**
  - **Installation engineers pick the appropriate community set when establishing the customer BGP session**
    - **Ease of operation!**

Community Example – Summary

- **Two examples of customer edge and internet edge can be combined to form a simple community solution for ISP prefix policy control**
- **More experienced operators tend to have more sophisticated options available**

  Advice is to start with the easy examples given, and then proceed onwards as experience is gained

Some ISP Examples

- ISPs also create communities to give customers bigger routing policy control
- Public policy is usually listed in the IRR
  - Following examples are all in the IRR
  - Examples build on the configuration concepts from the introductory example
- Consider creating communities to give policy control to customers
  - Reduces technical support burden
  - Reduces the amount of router reconfiguration, and the chance of mistakes

Some ISP Examples
Connect.com.au

- Australian ISP
- Run their own Routing Registry

  Whois.connect.com.au

- Permit customers to send up 8 types of communities to allow traffic engineering

Some ISP
Conne

```
aut-num:        AS2764
as-name:        ASN-CONNECT-NET
descr:          connect.com.au pty ltd
admin-c:        CC89
tech-c:         MP151
remarks:        Community  Definition
remarks:        ----------------------------------------
remarks:        2764:1 Announce to "domestic" rate ASes only
remarks:        2764:2 Don't announce outside local POP
remarks:        2764:3 Lower local preference by 25
remarks:        2764:4 Lower local preference by 15
remarks:        2764:5 Lower local preference by 5
remarks:        2764:6 Announce to non customers with "no-export"
remarks:        2764:7 Only announce route to customers
remarks:        2764:8 Announce route over satellite link
notify:         routing@connect.com.au
mnt-by:         CONNECT-AU
changed:        mrp@connect.com.au 19990506
source:         CCAIR
```

Some ISP Examples
UUNET Europe

- UUNET's European operation
- Permits customers to send communities which determine
  - local preferences within UUNET's network
  - Reachability of the prefix
  - How the prefix is announced outside of UUNET's network

Some IS
UUN

```
aut-num: AS702
as-name: AS702
descr:   UUNET - Commercial IP service provider in Europe
remarks: ------------------------------------------------------------
remarks: UUNET uses the following communities with its customers:
remarks: 702:80  Set Local Pref 80 within AS702
remarks: 702:120 Set Local Pref 120 within AS702
remarks: 702:20  Announce only to UUNET AS'es and UUNET customers
remarks: 702:30  Keep within Europe, don't announce to other UUNET AS's
remarks: 702:1   Prepend AS702 once at edges of UUNET to Peers
remarks: 702:2   Prepend AS702 twice at edges of UUNET to Peers
remarks: 702:3   Prepend AS702 thrice at edges of UUNET to Peers
remarks: Details of UUNET's peering policy and how to get in touch with
remarks: UUNET regarding peering policy matters can be found at:
remarks: http://www.uu.net/peering/
remarks: ------------------------------------------------------------
mnt-by:  UUNET-MNT
changed: eric-apps@eu.uu.net 20010928
source:  RIPE
```

Some ISP Examples
BT Ignite

- Formerly Concert's European network
- One of the most comprehensive community lists around
  - Seems to be based on definitions originally used in Tiscali's network
  - whois –h whois.ripe.net AS5400 reveals all
- Extensive community definitions allow sophisticated traffic engineering by customers

Some ISP
BT Ig

```
aut-num:        AS5400
as-name:        CIPCORE
descr:          BT Ignite European Backbone
remarks:        The following BGP communities can be set by BT Ignite
remarks:        BGP customers to affect announcements to major peers.
remarks:
remarks:        Community to                                Community to
remarks:        Not announce      To peer:                  AS prepend 5400
remarks:
remarks:        5400:1000 European peers                    5400:2000
remarks:        5400:1001 Sprint (AS1239)                   5400:2001
remarks:        5400:1003 Unisource (AS3300)                5400:2003
remarks:        5400:1005 UUnet (AS702)                     5400:2005
remarks:        5400:1006 Carrier1 (AS8918)                 5400:2006
remarks:        5400:1007 SupportNet (8582)                 5400:2007
remarks:        5400:1008 AT&T (AS2686)                     5400:2008
remarks:        5400:1009 Level 3 (AS9057)                  5400:2009
remarks:        5400:1010 RIPE (AS3333)                     5400:2010
<snip>
remarks:        5400:1100         US peers                  5400:2100
notify:         notify@eu.ignite.net
mnt-by:         CIP-MNT
source:         RIPE
```

And many many more!

Some ISP Examples
Carrier1

- European ISP
- Another very comprehensive list of community definitions

  whois –h whois.ripe.net AS8918 reveals all

Some ISP
Carrier

```
aut-num:        AS8918
descr:          Carrier1 Autonomous System
<snip>
remarks:        Community Support Definitions:
remarks:        Communities that determine the geographic
remarks:        entry point of routes into the Carrier1 network:
remarks:        *
remarks:        Community     Entry Point
remarks:        --------------------------------
remarks:        8918:10       London
remarks:        8918:15       Hamburg
remarks:        8918:18       Chicago
remarks:        8918:20       Amsterdam
remarks:        8918:25       Milan
remarks:        8918:28       Berlin
remarks:        8918:30       Frankfurt
remarks:        8918:35       Zurich
remarks:        8918:40       Geneva
remarks:        8918:45       Stockholm
<snip>
notify:         inoc@carrier1.net
mnt-by:         CARRIER1-MNT
source:         RIPE
```

Some ISP Examples
Level 3

- **Highly detailed AS object held on the RIPE Routing Registry**
- **Also a very comprehensive list of community definitions**

  whois –h whois.ripe.net AS3356 reveals all

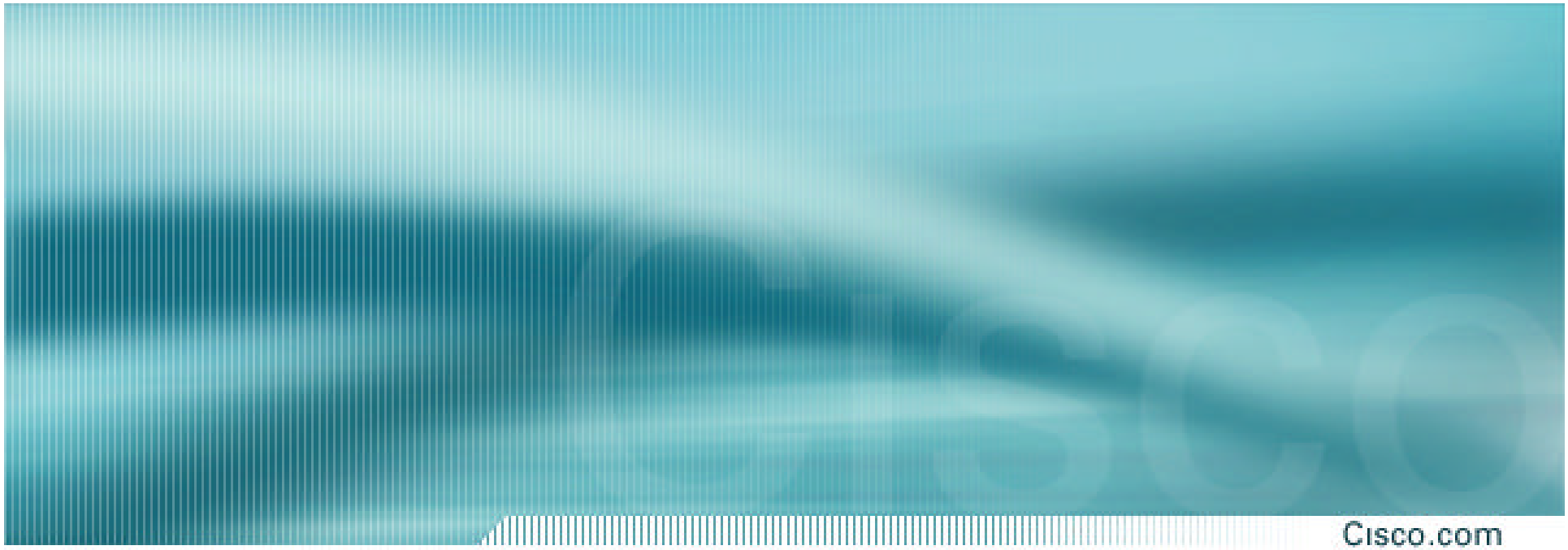
Some IS
Leve

```
aut-num:      AS3356
descr:        Level 3 Communications
<snip>
remarks:      ------------------------------------------------------
remarks:      customer traffic engineering communities - Suppression
remarks:      ------------------------------------------------------
remarks:      64960:XXX - announce to AS XXX if 65000:0
remarks:      65000:0   - announce to customers but not to peers
remarks:      65000:XXX - do not announce at peerings to AS XXX
remarks:      ------------------------------------------------------
remarks:      customer traffic engineering communities - Prepending
remarks:      ------------------------------------------------------
remarks:      65001:0   - prepend once  to all peers
remarks:      65001:XXX - prepend once  at peerings to AS XXX
remarks:      65002:0   - prepend twice to all peers
remarks:      65002:XXX - prepend twice at peerings to AS XXX
remarks:      65003:0   - prepend 3x to all peers
remarks:      65003:XXX - prepend 3x    at peerings to AS XXX
remarks:      65004:0   - prepend 4x to all peers
remarks:      65004:XXX - prepend 4x    at peerings to AS XXX
<snip>
mnt-by:       LEVEL3-MNT
source:       RIPE
```

And many many more!

BGP for Internet Service Providers

- BGP Basics (quick recap)
- Scaling BGP
- Using Communities
- Deploying BGP in an ISP network



Deploying BGP in an ISP Network

Best 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 Configuration Example

Cisco.com

```
router bgp 34567
 neighbor core-ibgp peer-group
 neighbor core-ibgp remote-as 34567
 neighbor core-ibgp update-source Loopback0
 neighbor core-ibgp send-community
 neighbor core-ibgp-partial peer-group
 neighbor core-ibgp-partial remote-as 34567
 neighbor core-ibgp-partial update-source Loopback0
 neighbor core-ibgp-partial send-community
 neighbor core-ibgp-partial prefix-list network-ibgp out
 neighbor 222.1.9.10 peer-group core-ibgp
 neighbor 222.1.9.13 peer-group core-ibgp-partial
 neighbor 222.1.9.14 peer-group core-ibgp-partial
```

BGP versus OSPF/ISIS

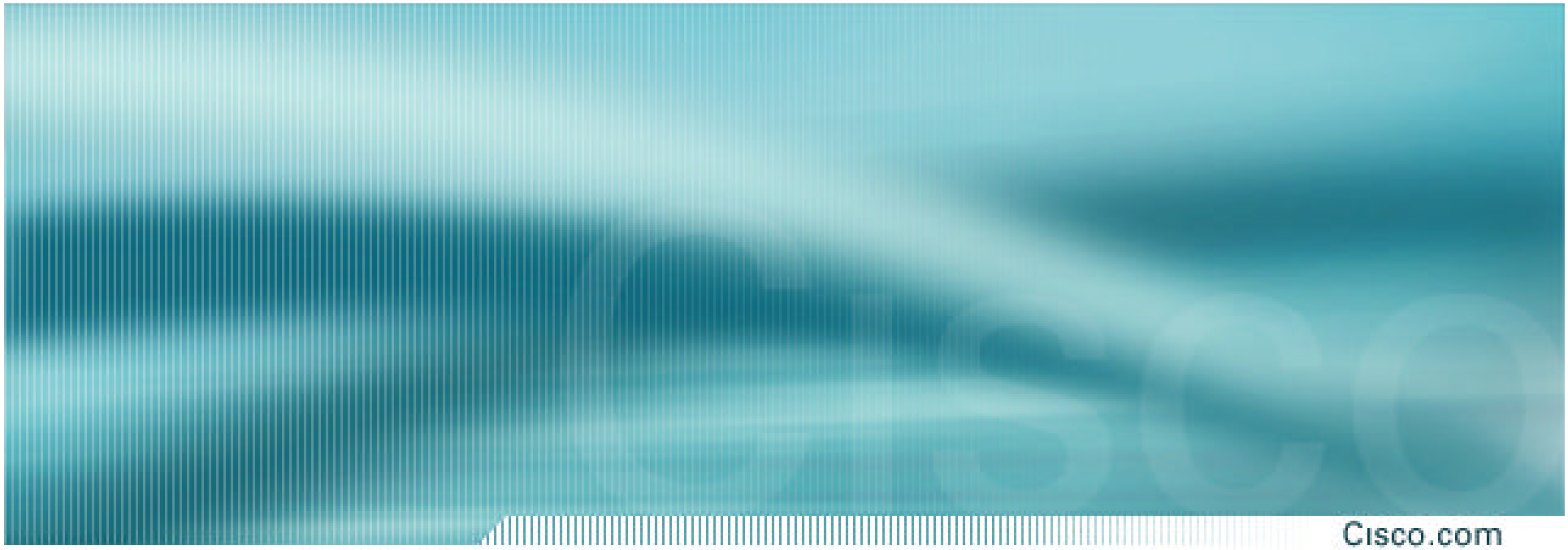
- **DO NOT:**

  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

Cisco.com

- 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”

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 >65000 /24s!

The Internet Today (January 2003)

- **Current Internet Routing Table Statistics**

BGP Routing Table Entries	119544
Prefixes after maximum aggregation	76260
Unique prefixes in Internet	57040
Prefixes smaller than registry alloc	55563
/24s announced	66125
only 5406 /24s are from 192.0.0.0/8	
ASes in use	14361

Efforts to improve aggregation

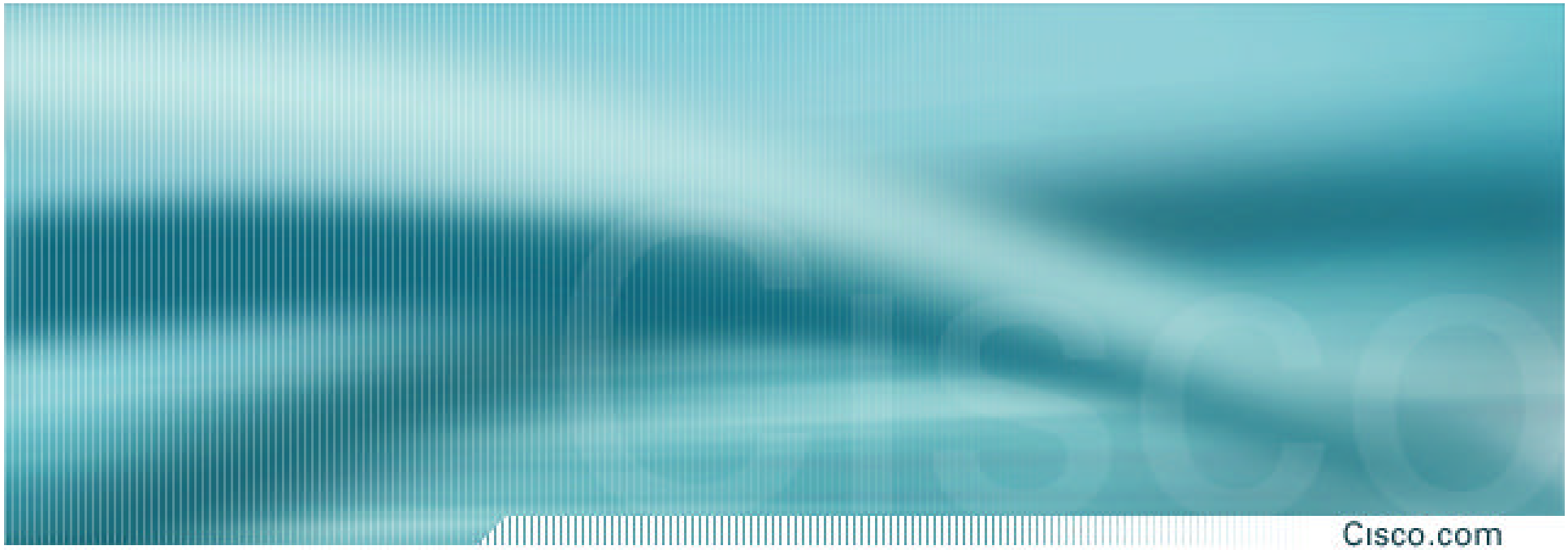
- **The CIDR Report**
  - **Initiated and operated for many years by Tony Bates**
  - **Now combined with Geoff Huston's routing analysis**
    - **www.cidr-report.org**
  - **Results e-mailed on a weekly basis to most operations lists around the world**
  - **Lists the top 30 service providers who could do better at aggregating**
  - **Also computes the size of the routing table assuming ISPs performed optimal aggregation**
  - **Website allows searches and computations to be made on a per AS basis – flexible and powerful tool to aid ISPs**

The CIDR Report

Cisco.com

--- 10Jan03 ---

ASnum	NetsNow	NetsAggr	NetGain	% Gain	Description
Table	118165	85303	32862	27.8%	All ASes
AS3908	1175	684	491	41.8%	SuperNet, Inc.
AS18566	422	5	417	98.8%	Covad Communications
AS7018	1450	1035	415	28.6%	AT&T WorldNet
AS701	1606	1193	413	25.7%	UUNET Technologies
AS4323	526	188	338	64.3%	Time Warner Communications
AS7843	628	291	337	53.7%	Adelphia Corp.
AS6197	458	150	308	67.2%	BellSouth Network Solutions
AS1221	1145	844	301	26.3%	Telstra Pty Ltd
AS1239	968	679	289	29.9%	Sprint
AS6347	369	85	284	77.0%	DIAMOND SAVVIS Communications Corp
AS4355	406	135	271	66.7%	EARTHLINK, INC
AS7046	554	286	268	48.4%	UUNET Technologies
AS22927	289	22	267	92.4%	TELEFONICA DE ARGENTINA
AS705	426	186	240	56.3%	UUNET Technologies, Inc.
AS4814	251	15	236	94.0%	China Telecom (Group)
AS1	661	439	222	33.6%	Genuity
AS6198	422	200	222	52.6%	BellSouth Network Solutions, Inc
AS17676	229	24	205	89.5%	GIGAINFRA XTAGE CORPORATION
AS22291	227	29	198	87.2%	Charter Communications
AS690	513	319	194	37.8%	Merit Network Inc.



Receiving Prefixes

Receiving Prefixes: From Downstreams

- ISPs should only accept prefixes which have been assigned or allocated to their downstream customer
- 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 Upstreams

- **Not desirable unless really necessary**
  - **special circumstances – see later**
- **Ask upstream to either:**
  - **originate a default-route**
    - **-or-**
  - **announce one prefix you can use as default**

Receiving Prefixes: From Upstreams

- **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 Upstreams

- **Upstream Router Configuration**

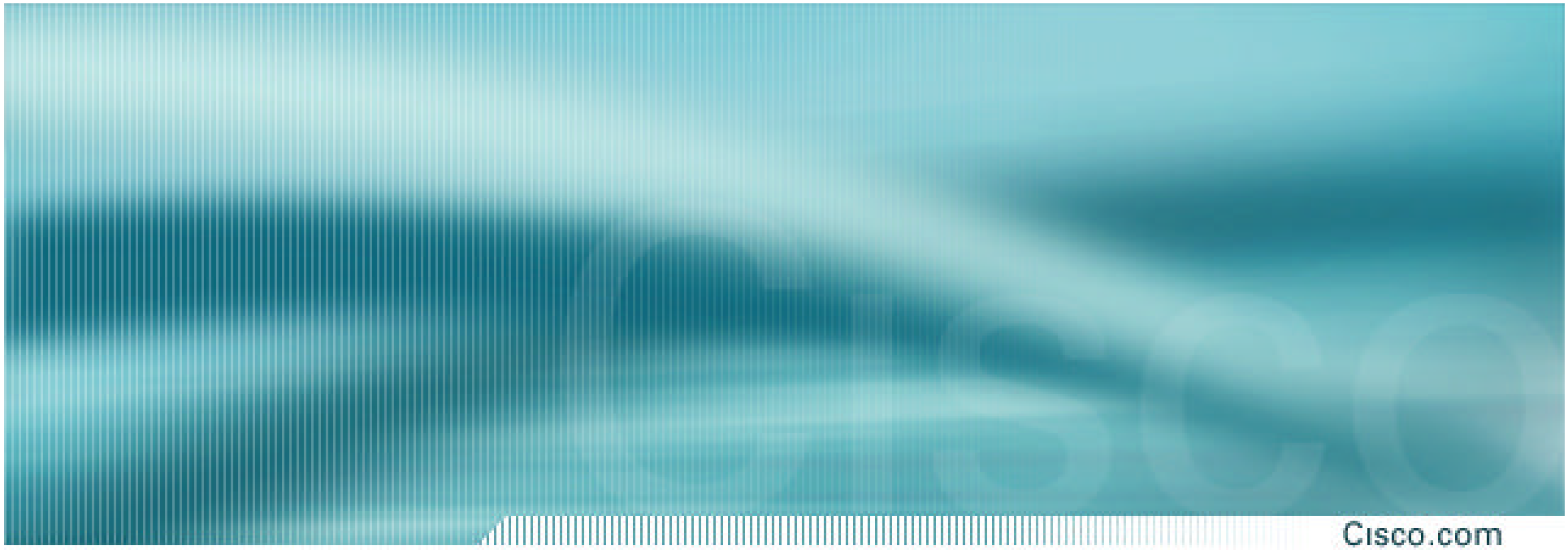
```
router bgp 101
 neighbor 221.5.7.2 remote-as 100
 neighbor 221.5.7.2 default-originate
 neighbor 221.5.7.2 prefix-list cust-in in
 neighbor 221.5.7.2 prefix-list cust-out out
!
ip prefix-list cust-in permit 221.10.0.0/19
!
ip prefix-list cust-out permit 0.0.0.0/0
```

Receiving Prefixes: From Peers and Upstreams

- **If necessary to receive prefixes from any provider, care is required**
  - **don't accept RFC1918 etc prefixes**
    - http://www.ietf.org/internet-drafts/draft-manning-dsua-08.txt
    - ftp://ftp.rfc-editor.org/in-notes/rfc3330.txt
  - **don't accept your own prefix**
  - **don't accept default (unless you need it)**
  - **don't accept prefixes longer than /24**
- **Check Rob Thomas' list of "bogons"**
  - http://www.cymru.org/Documents/bogon-list.html

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

Cisco.com

- **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

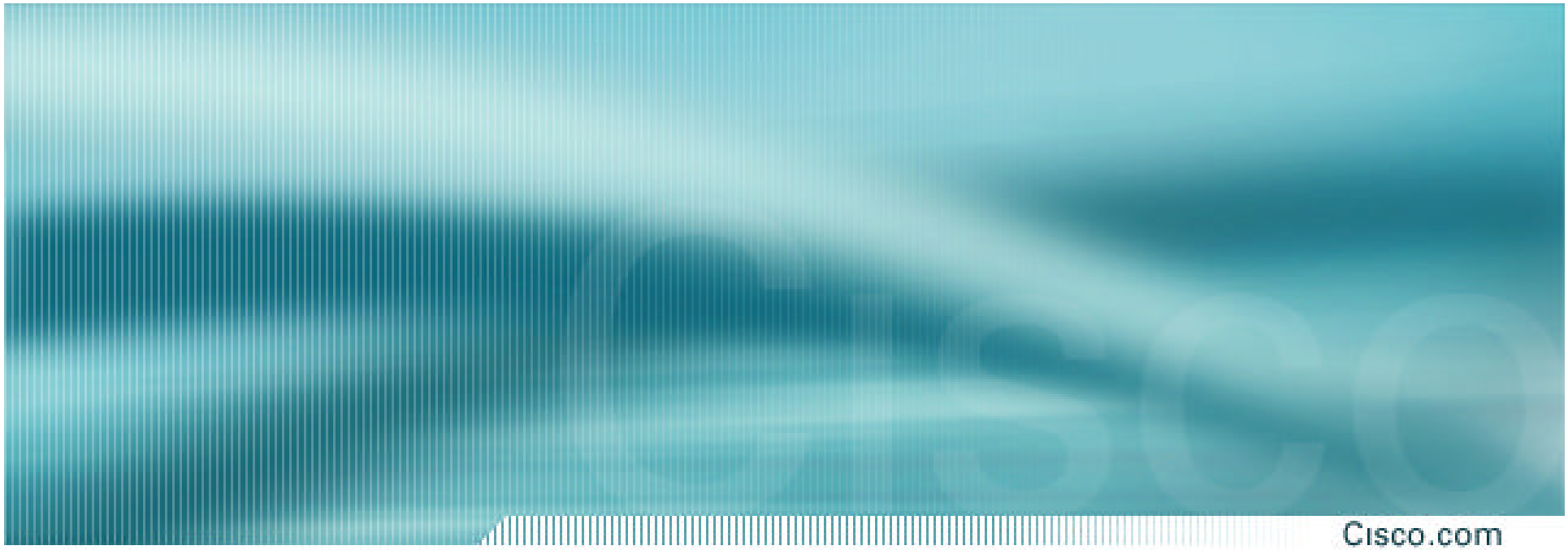
Router Configuration: redistribute static

- **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**
  - **Use ip-unnumbered where possible**
  - **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**

Templates

- **Good practice to configure templates for everything**
  - **Vendor defaults tend not to be optimal or even very useful for ISPs**
  - **ISPs create their own defaults by using configuration templates**
  - **Sample iBGP and eBGP templates follow for Cisco IOS**

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
```

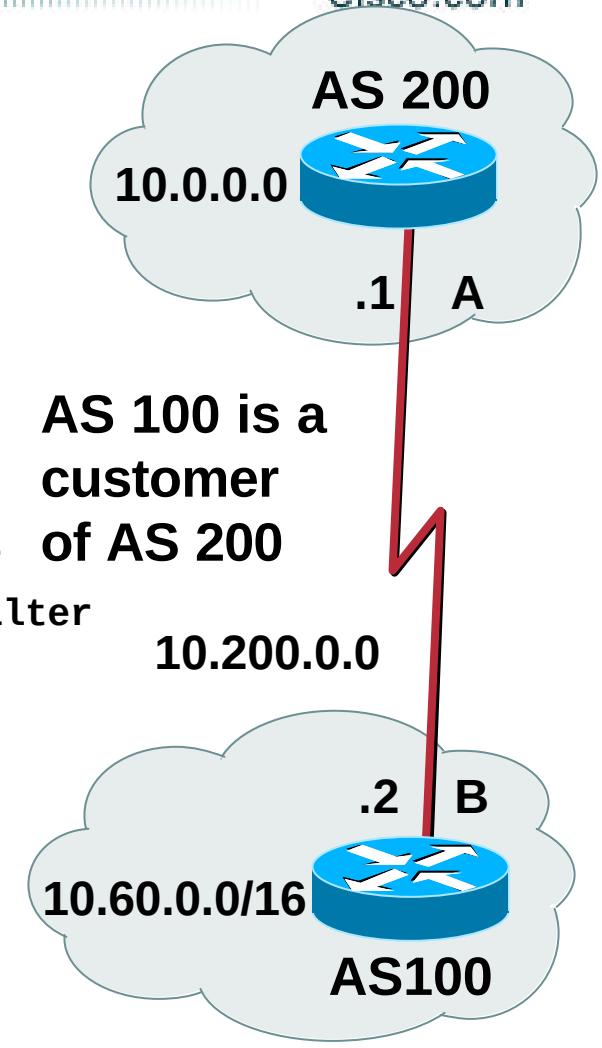
BGP Template – iBGP peers

Cisco.com

- 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 RIPE229-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 ! “real” filter
neighbor external filter-list 1 out      ! “accident” filter
neighbor external route-map ispout out
neighbor external prefix-list ispin in
neighbor external filter-list 2 in
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
```



BGP Template – eBGP peers

- BGP damping – use RIPE-229 parameters
- Remove private ASes from announcements
  - Common omission today
- Use extensive filters, with “backup”
  - Use as-path filters to backup prefix-lists
  - Use route-maps for policy
- 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 desired

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

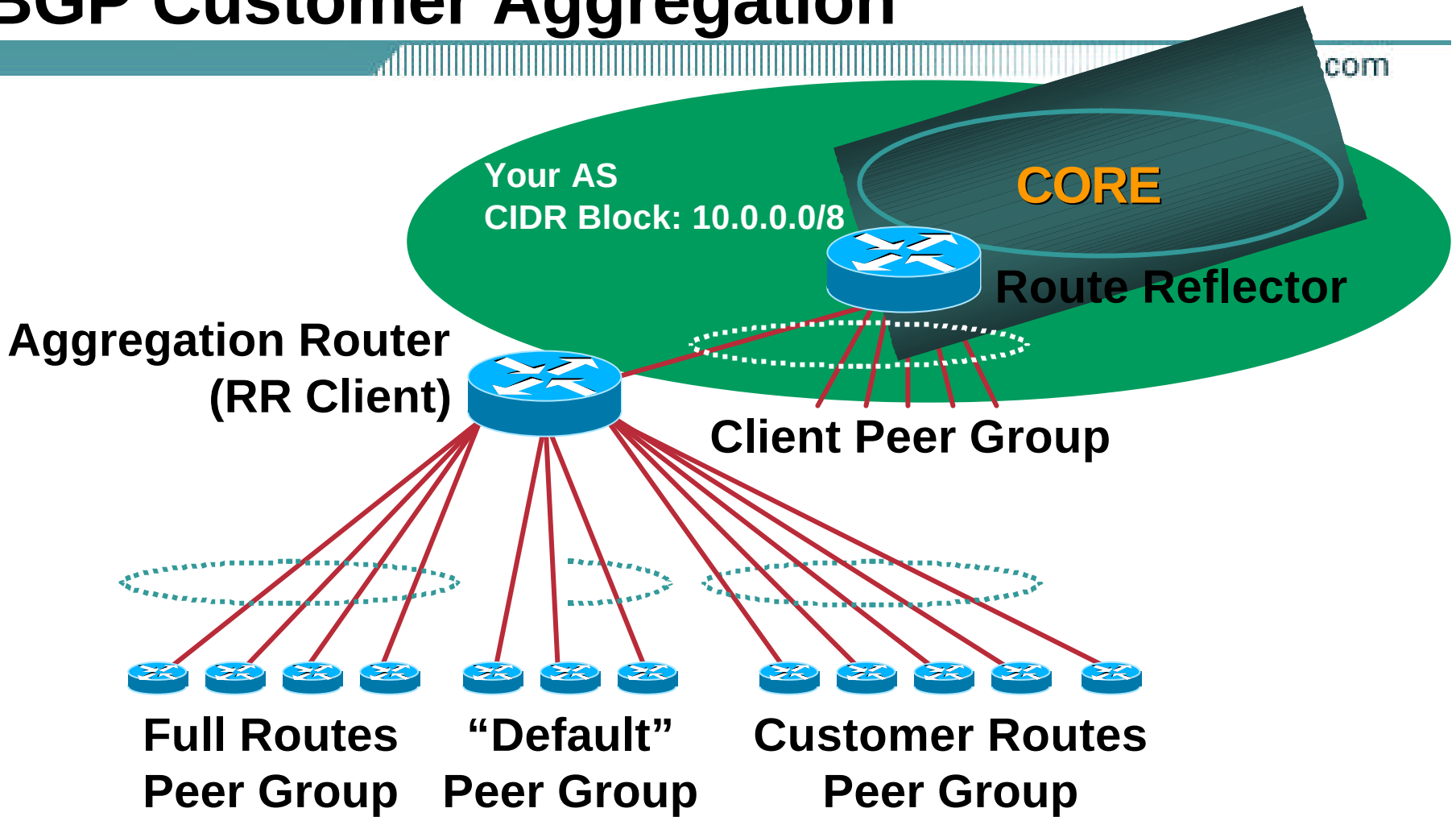
Customer Aggregation

- **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

  100:4100 = “specials” from my blocks

  100:4200 = customers from my blocks

  100:4300 = customers outside my blocks

  Helps with aggregation, iBGP, filtering

- BGP network statements on aggregation routers set correct community

Sample core configuration

Cisco.com

- **eBGP peers and upstreams**

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

- **iBGP full routes**

  Send everything (only to network core)

- **iBGP partial routes**

  Send communities 100:4000, 100:4100, 100:4200, 100:4300 and 100:4500 (to 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?

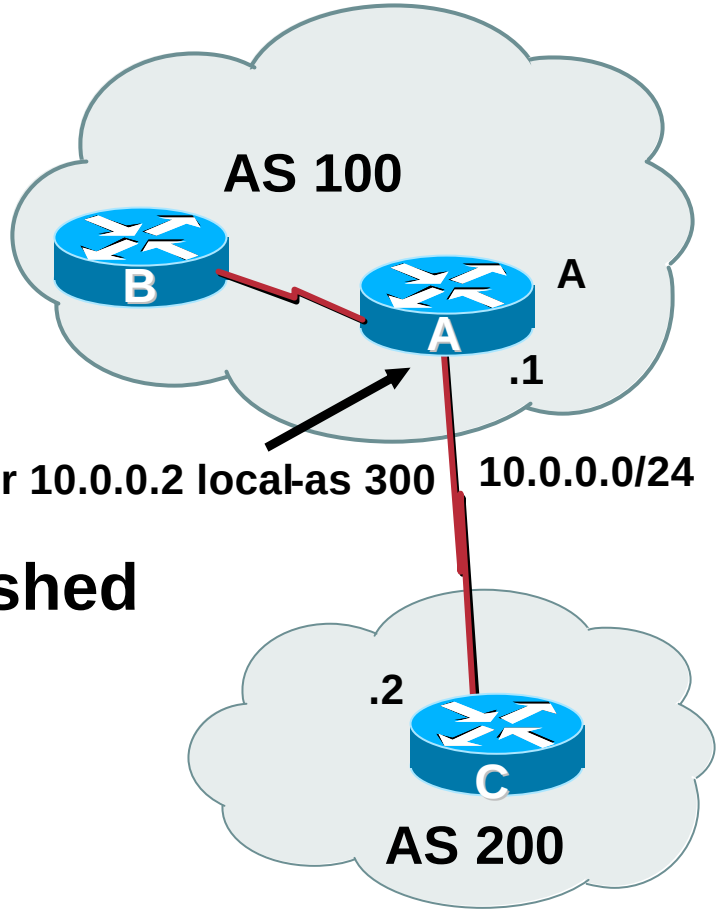
- **Options:**

  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

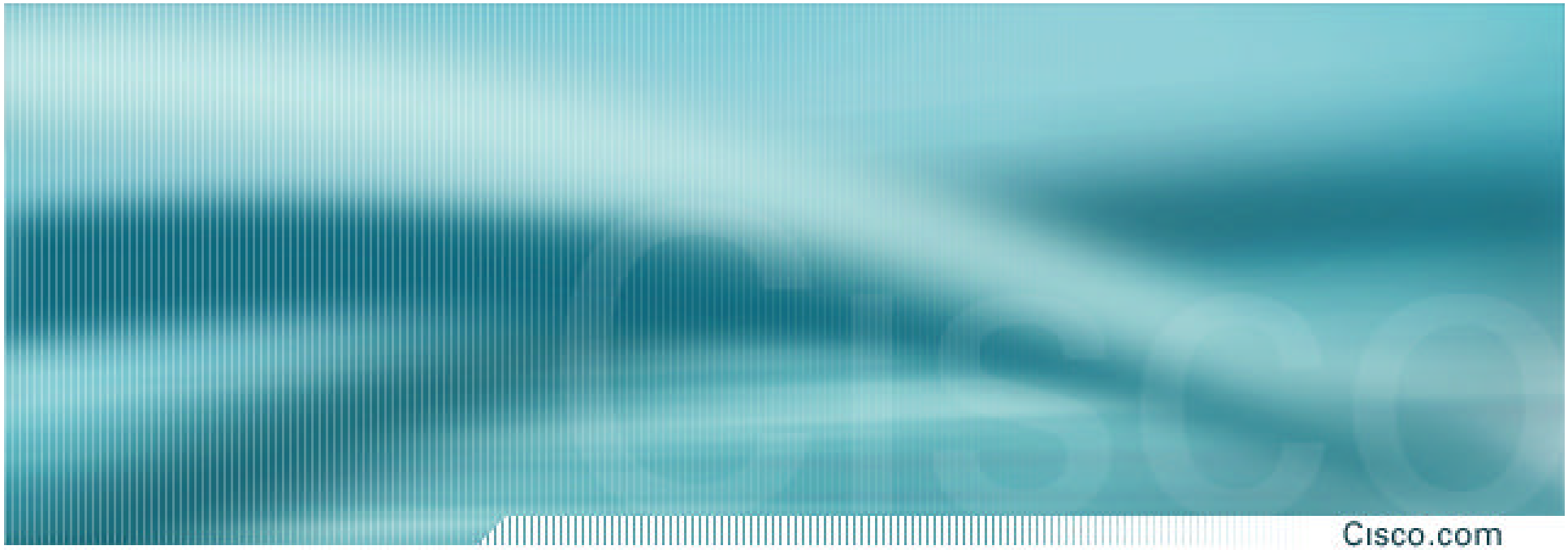
  routes originated in AS100 = 300 100

  routes received from AS200 = 300 200

BGP for Internet Service Providers

Cisco.com

- BGP Basics (quick recap)
- Scaling BGP
- Using Communities
- Deploying BGP in an ISP network



BGP for Internet Service Providers

End of Tutorial