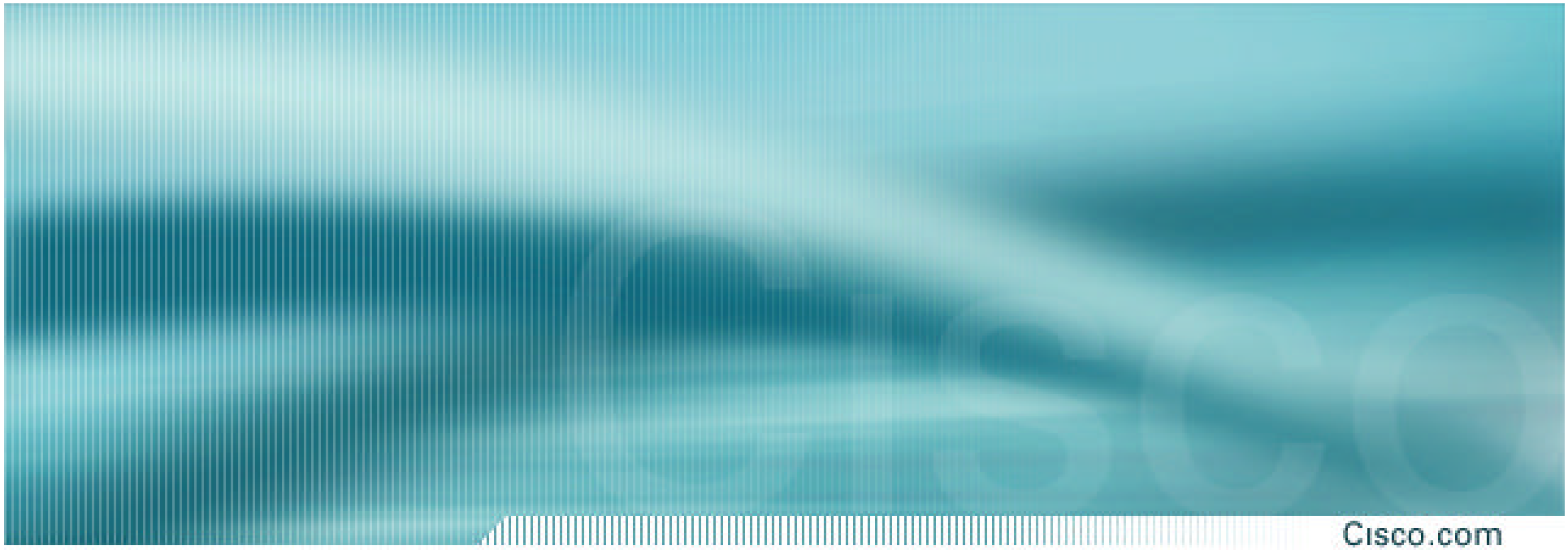


BGP for Internet Service Providers

Philip Smith <pfs@cisco.com>

KIOW2002 Seoul

BGP current status

- RFC1771 is quite old, and no longer reflects current operational practice nor vendor implementations
- Work in progress to update:

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

- BGP has been extended to support capability negotation

  Now allows multiprotocol support for BGP, amongst many other new developments

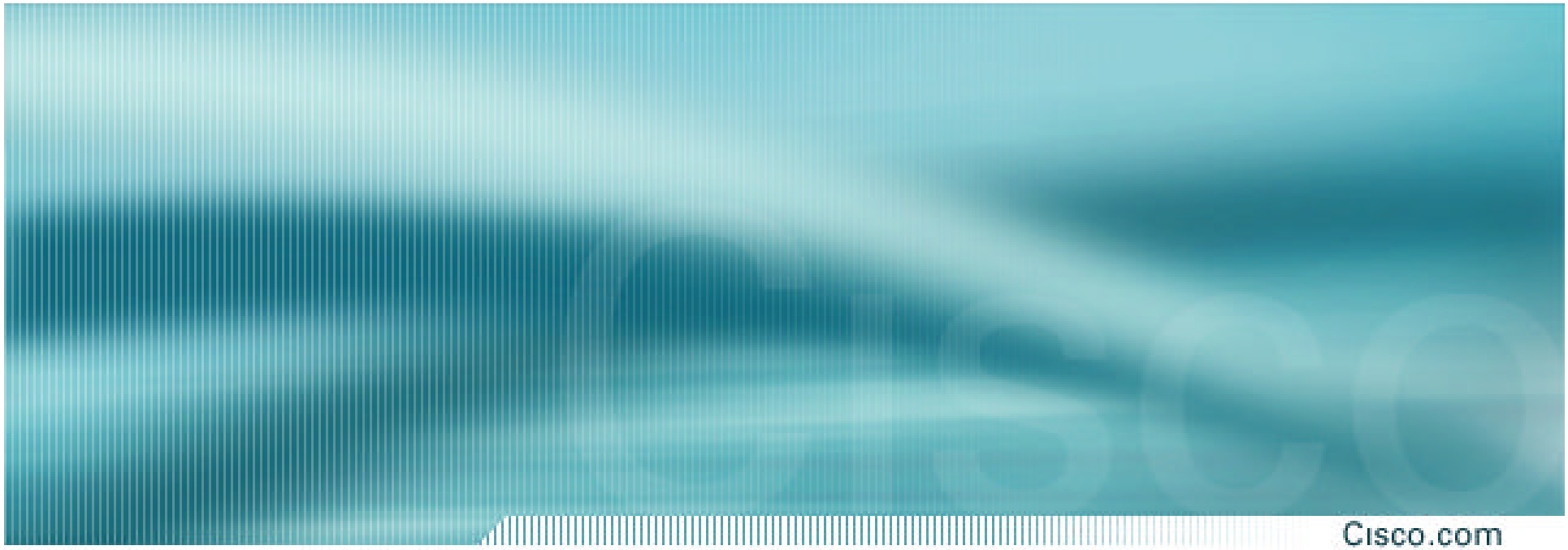
BGP Capabilities

- Documented in RFC3392
- Capabilities parameters passed in BGP open message
- Unknown or unsupported capabilities will result in NOTIFICATION message
- Current capabilities are:

```
  0  Reserved                                        [RFC3392]
  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 for Internet Service Providers

- **Scaling BGP**
- **Best Current Practices**
- **Configuration Tips**

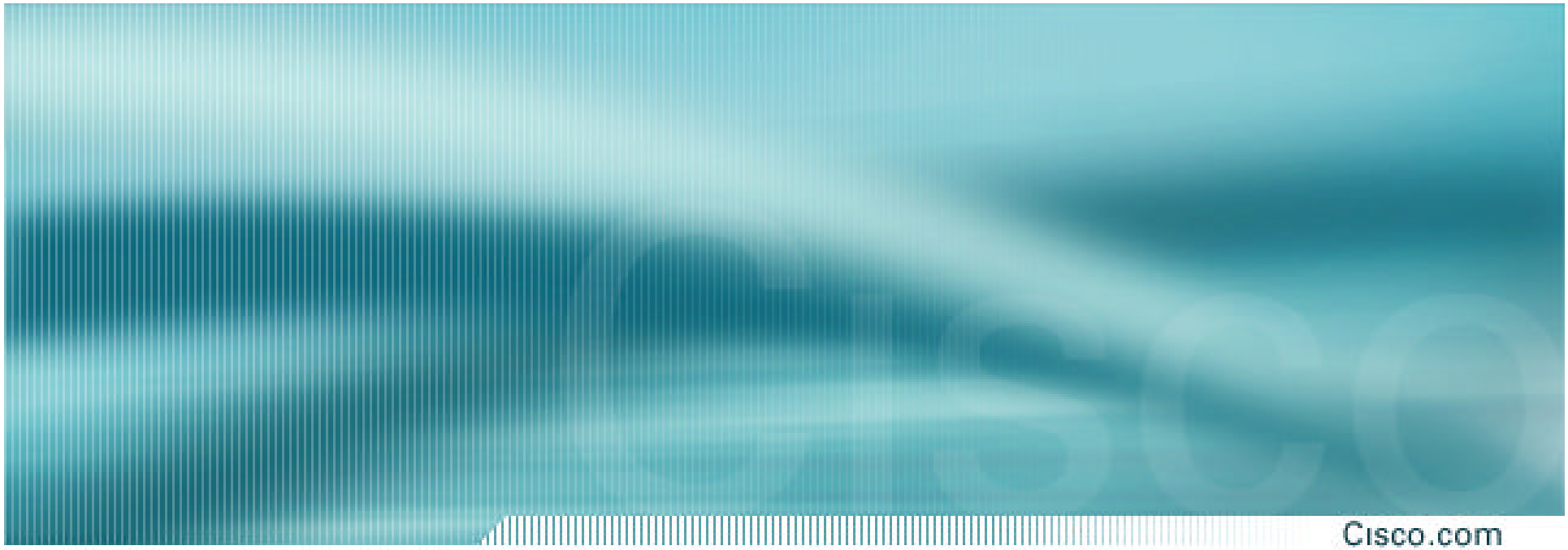


Scaling BGP

Designing in Scalability

BGP Scaling Techniques

- **When ISPs deploy BGP they have to consider the following:**
  - **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?**



Route Refresh

Dynamic Policy Changes for BGP

Route Refresh

Problem:

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

Solution:

- **Route Refresh**

Route Refresh Capability

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

Dynamic Reconfiguration

- **Use Route Refresh capability if supported**
  - **find out from “show ip bgp neighbor”**
  - **Non-disruptive, “Good For the Internet”**
- **Otherwise use Soft Reconfiguration 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 only stores prefixes which have been received from peer *after* policy application**

  Enabling soft-reconfiguration means router also stores prefixes/attributes received prior to any policy application

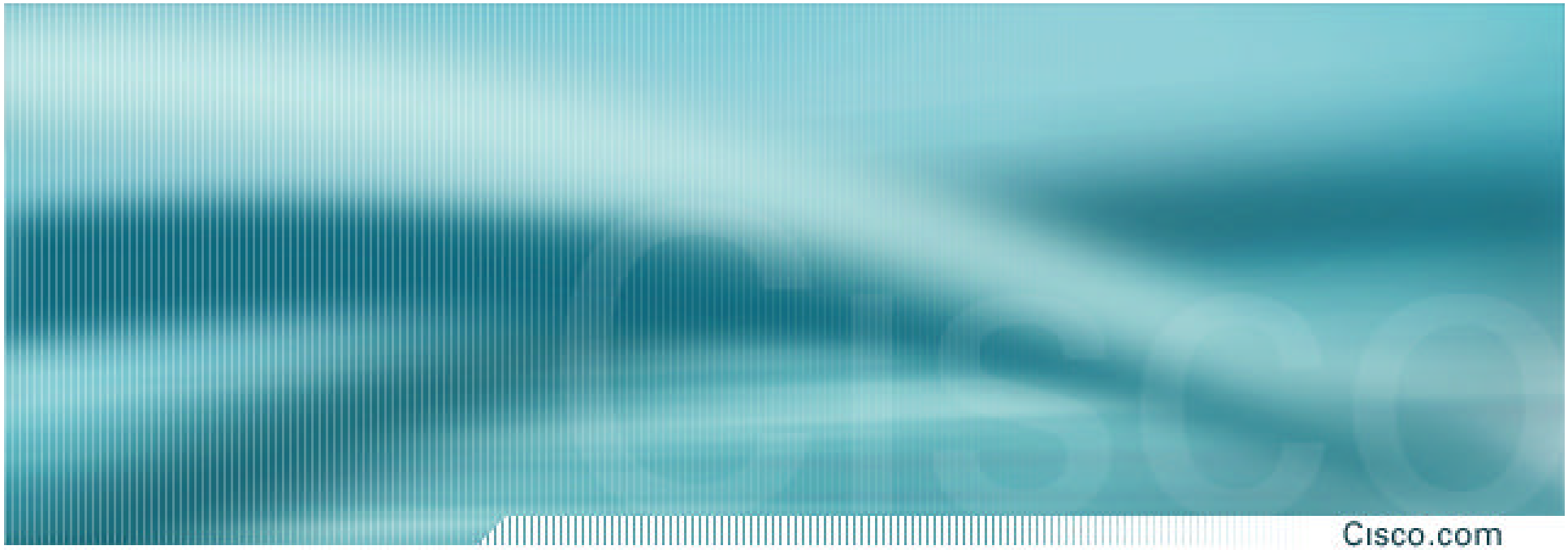
- **New policies can be activated without tearing down and restarting the peering session**
- **Configured on a per-neighbour basis**

Soft Reconfiguration

- Caveat: Uses more memory to keep prefixes whose attributes have been changed or have not been accepted
- Soft Reconfiguration is only used when:

  BGP neighbour does not support Route Refresh BGP Capability

  Local BGP speaker wants to find out what neighbour sent prior to local inbound policy being applied – useful for troubleshooting



Peer Groups

Saving Time

Peer Groups

In an iBGP full mesh:

- **iBGP neighbours receive same update**
- **Large iBGP mesh builds slowly**
- **Router CPU wasted on repeat calculations**

Solution – peer groups!

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

Peer Groups – Advantages

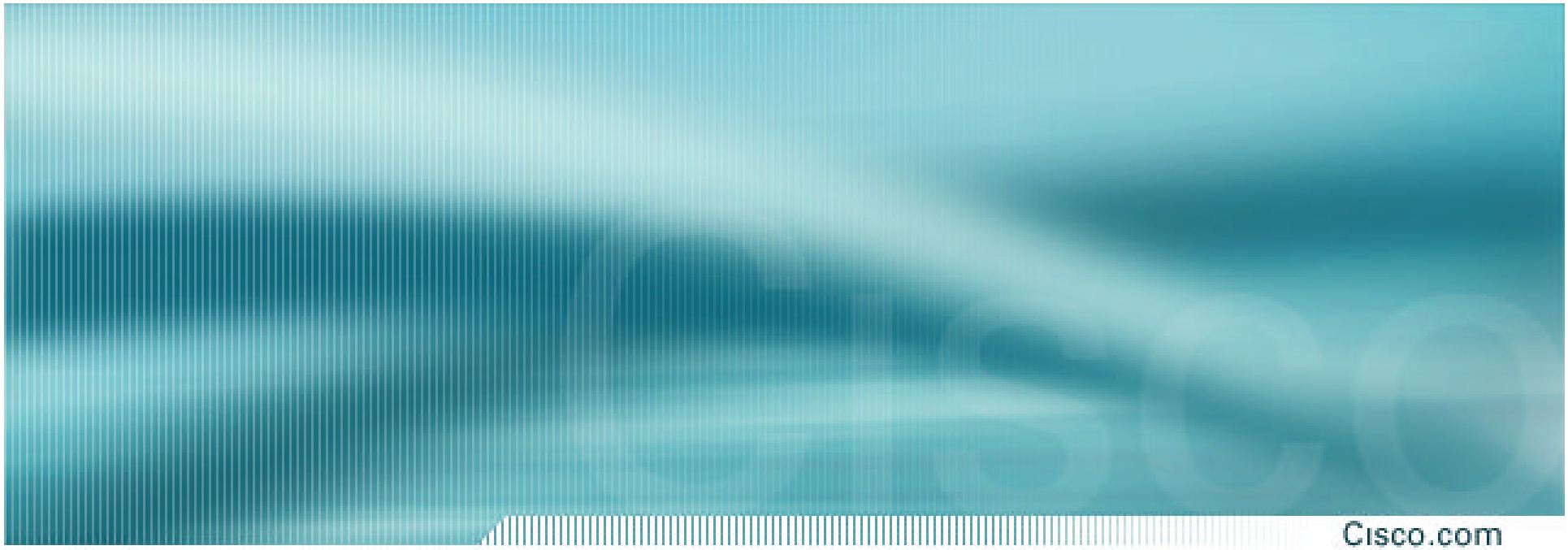
- Makes configuration easier
- Makes configuration less prone to error
- Makes configuration more readable
- Lower router CPU load
- iBGP mesh builds more quickly
- Members can have different inbound policy
- Can be used for eBGP neighbours too!

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* !



Route Flap Damping

Stabilising the Network

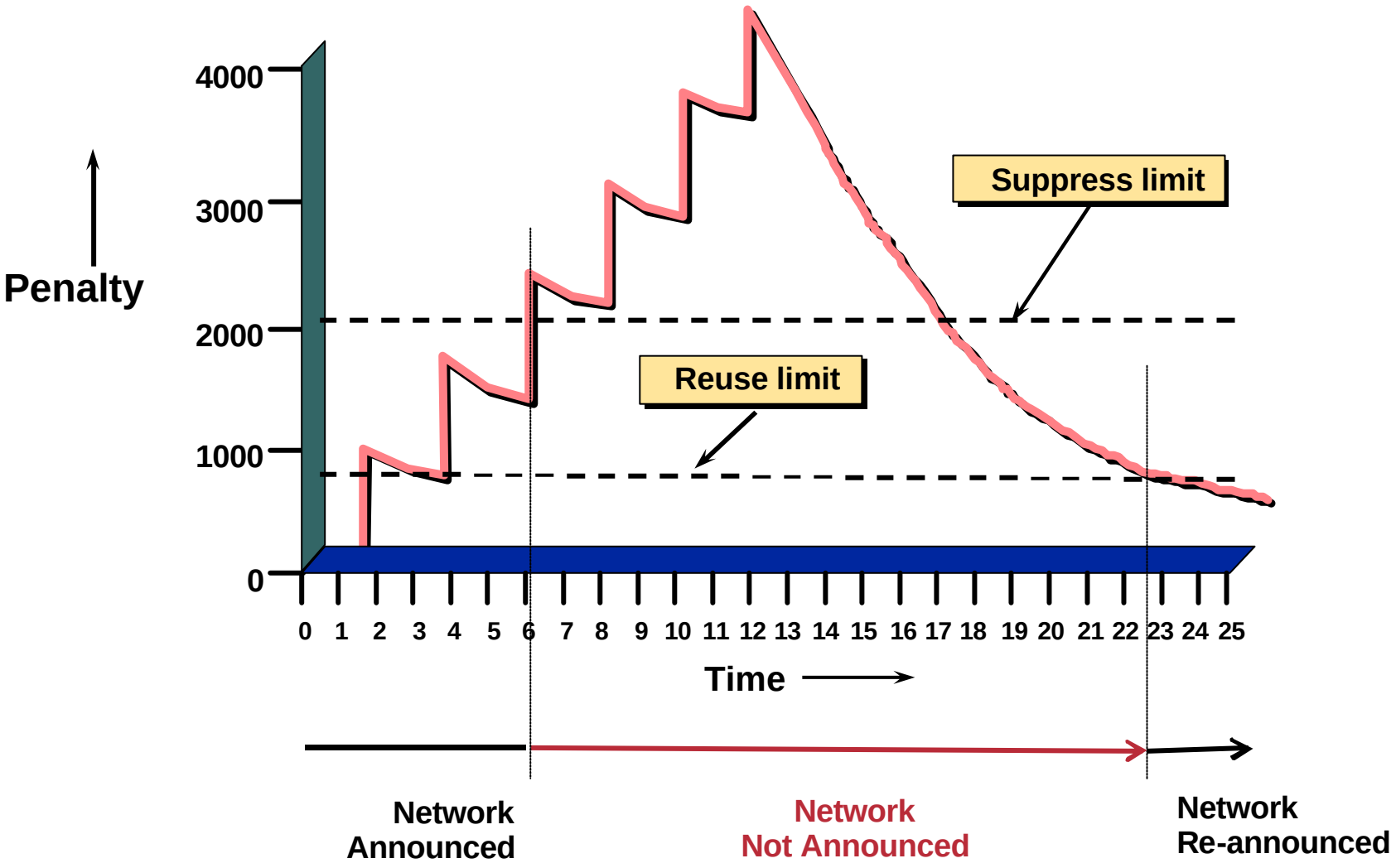
Route Flap Damping

- Route flap
  - Going up and down of path or change in attribute
    - BGP WITHDRAW followed by UPDATE = 1 flap
    - Change in BGP attribute = 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
- Documented in RFC2439

Route Flap Damping (continued)

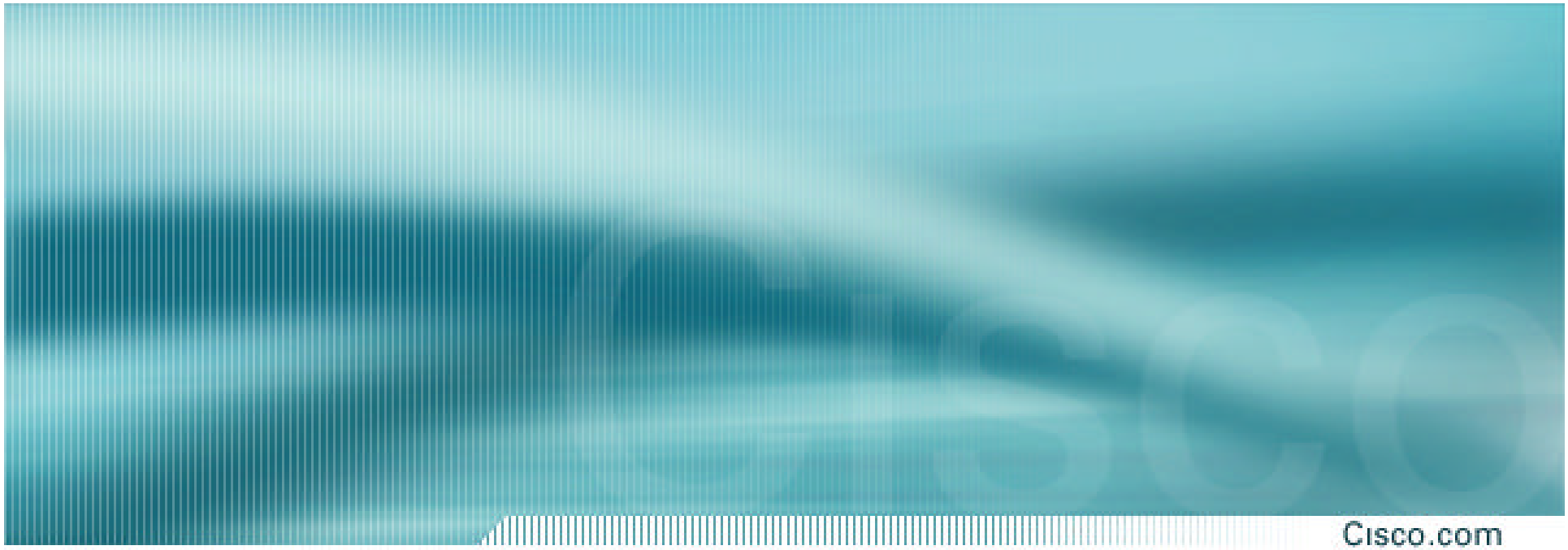
- **Requirements**
  - **Fast convergence for normal route changes**
  - **History predicts future behaviour**
  - **Suppress oscillating routes but advertise stable routes**
- **Operation**
  - **Add penalty (1000) for each flap**
    - **Change in attribute gets penalty of 500**
  - **Exponentially decay penalty (determined by half-life)**
  - **Penalty above suppress-limit ® route not advertised**
  - **Penalty decayed below reuse-limit ® route re-advertised**
    - **penalty reset to zero when it is half of reuse-limit**

Operation



Operation & Configuration

- 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)
- Recommendations for ISPs
  - http://www.ripe.net/docs/ripe-229.html



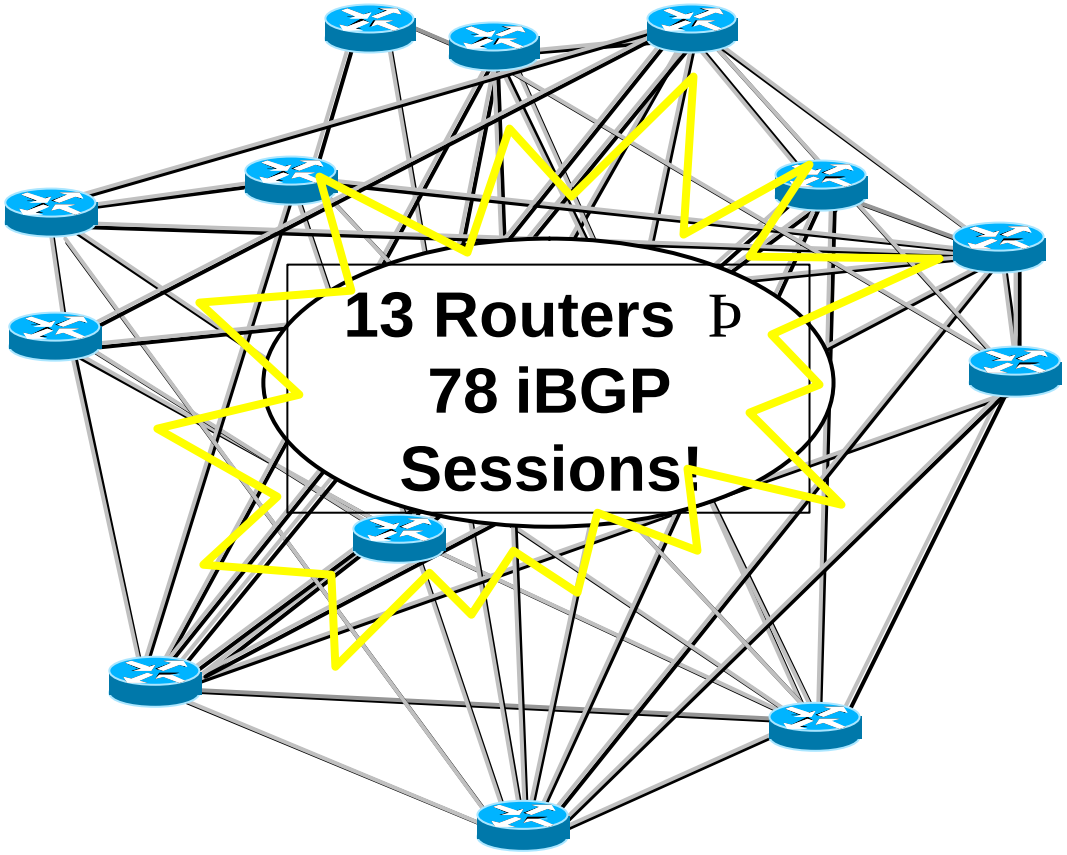
Route Reflectors

Scaling the iBGP mesh

Scaling the iBGP mesh

ISPs have to avoid ½n(n-1) iBGP mesh

n=1000 Þ nearly half a million ibgp sessions!

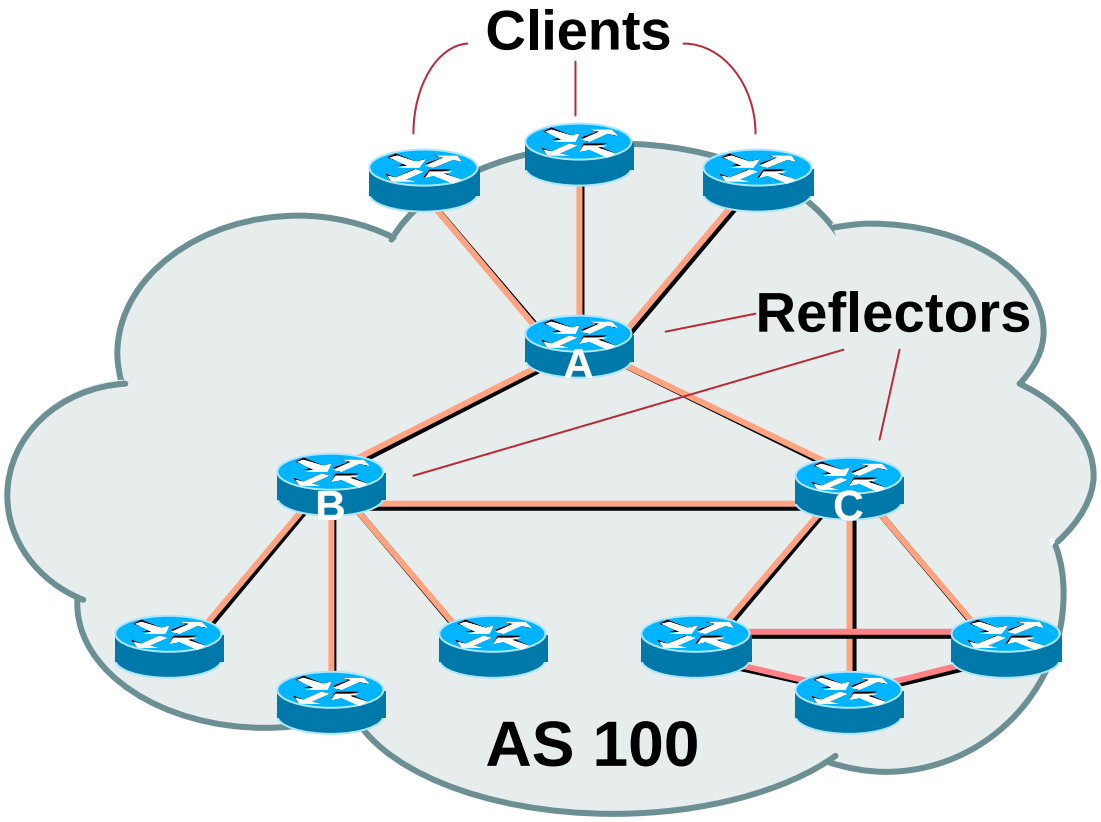


Two solutions

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

Route Reflector

- 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

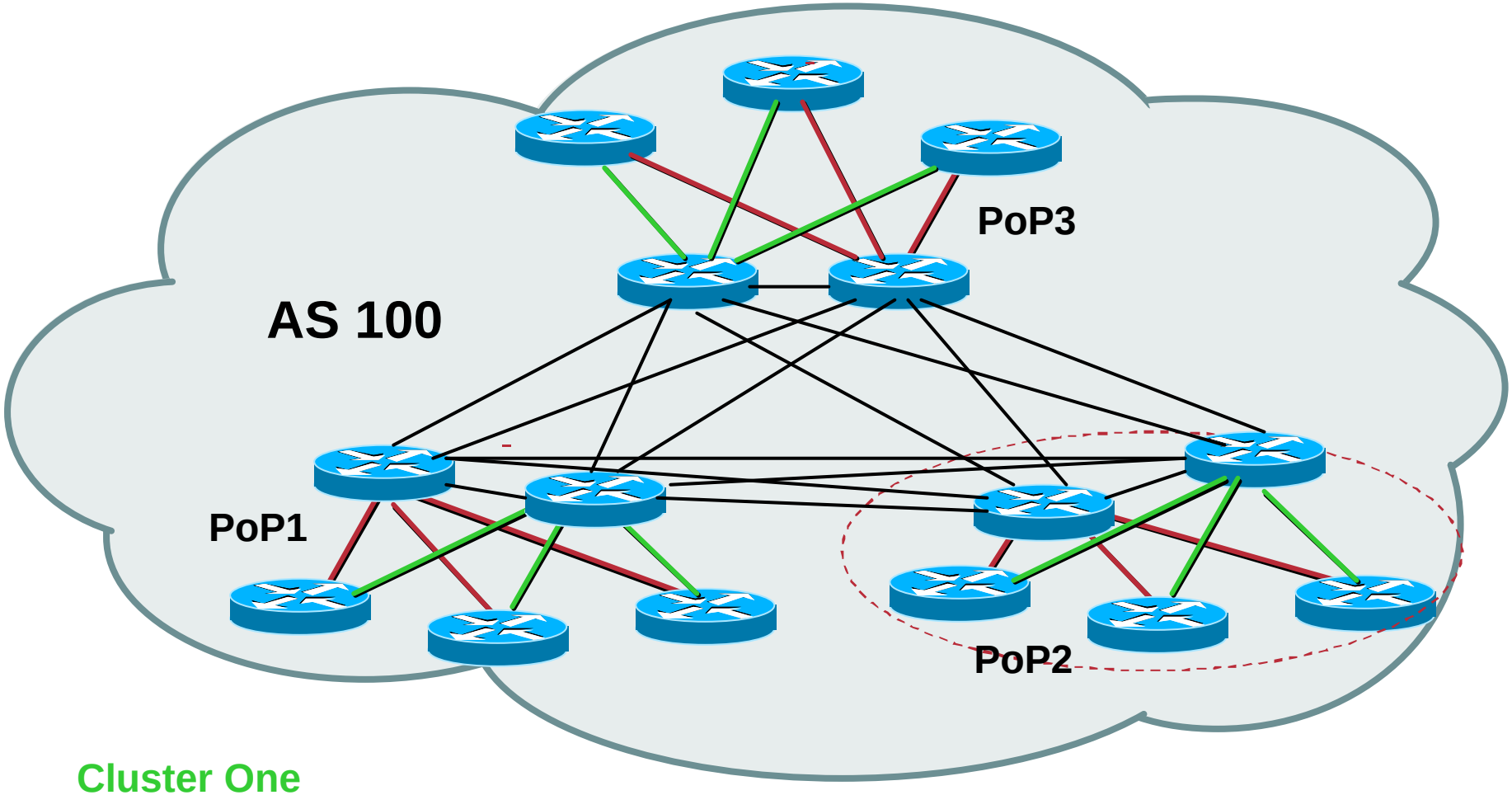
  Cluster-id is automatically set from router-id (address of loopback)

  Do NOT use *bgp cluster-id x.x.x.x*

Route Reflectors: Redundancy

- **Multiple RRs can be configured in the same cluster – but 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

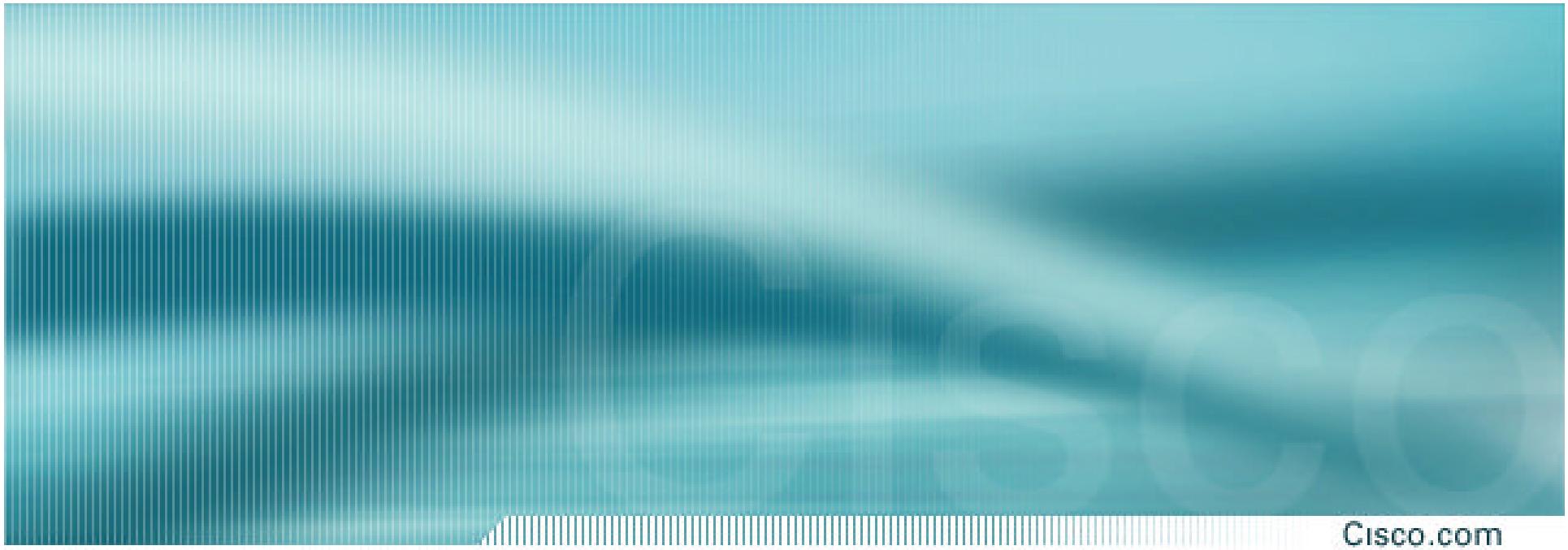
- **Typical ISP network:**
  - **Core routers have fully meshed iBGP**
  - **Create further hierarchy if core mesh too big**
    - **Split backbone into regions**
- **Configure one cluster pair at a time**
  - **Eliminate redundant iBGP sessions**
  - **Use only one RR per cluster**
  - **Use at least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

BGP Scaling Techniques

- **These 4 techniques are necessary requirements in all ISP networks**
  - **Route Refresh**
  - **Peer groups**
  - **Route flap damping**
  - **Route reflectors**
- **All new ISP networks should implement these techniques from DAY ONE**
- **All operational ISP networks should consider migrating to support these 4 techniques**

BGP for Internet Service Providers

- Scaling BGP
- Best Current Practices
- Configuration Tips



Best Current Practices

Being a Good Internet Citizen

Deploying BGP in an ISP network

BGP versus OSPF/ISIS

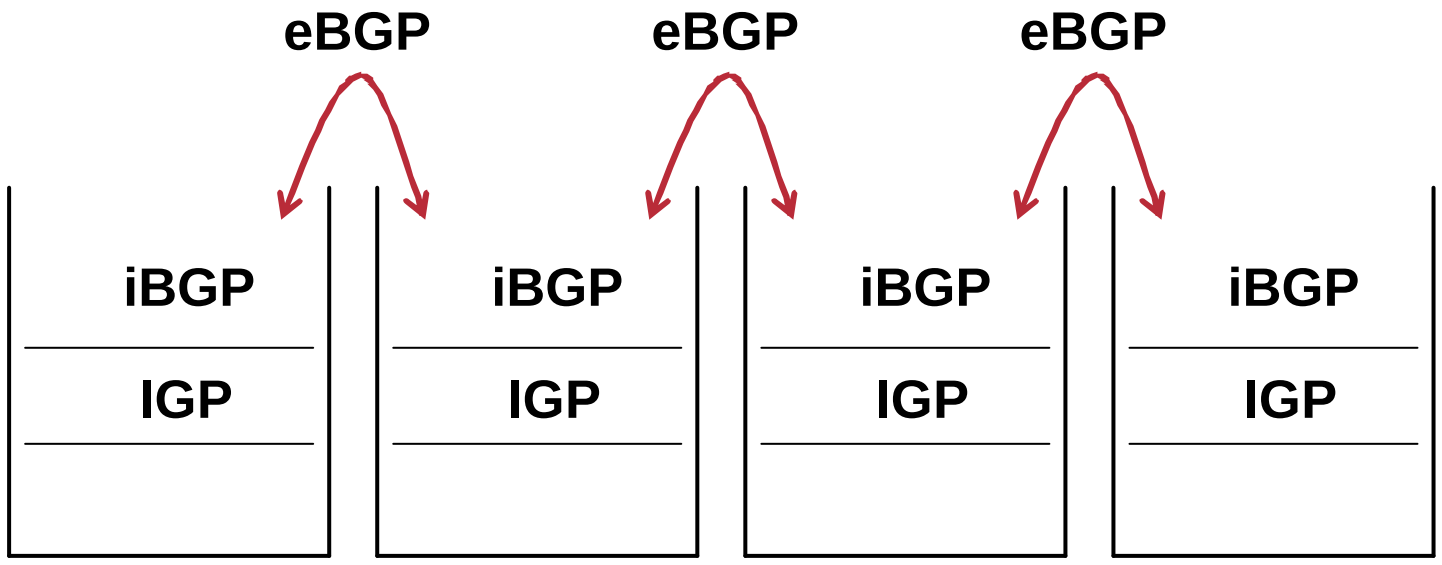
- **Internal Routing Protocols (IGPs)**
  - **examples are ISIS and OSPF**
  - **used for carrying infrastructure addresses**
  - **NOT used for carrying Internet prefixes or customer prefixes**
  - **design goal is to minimise number of prefixes in IGP to aid scalability and rapid convergence**

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

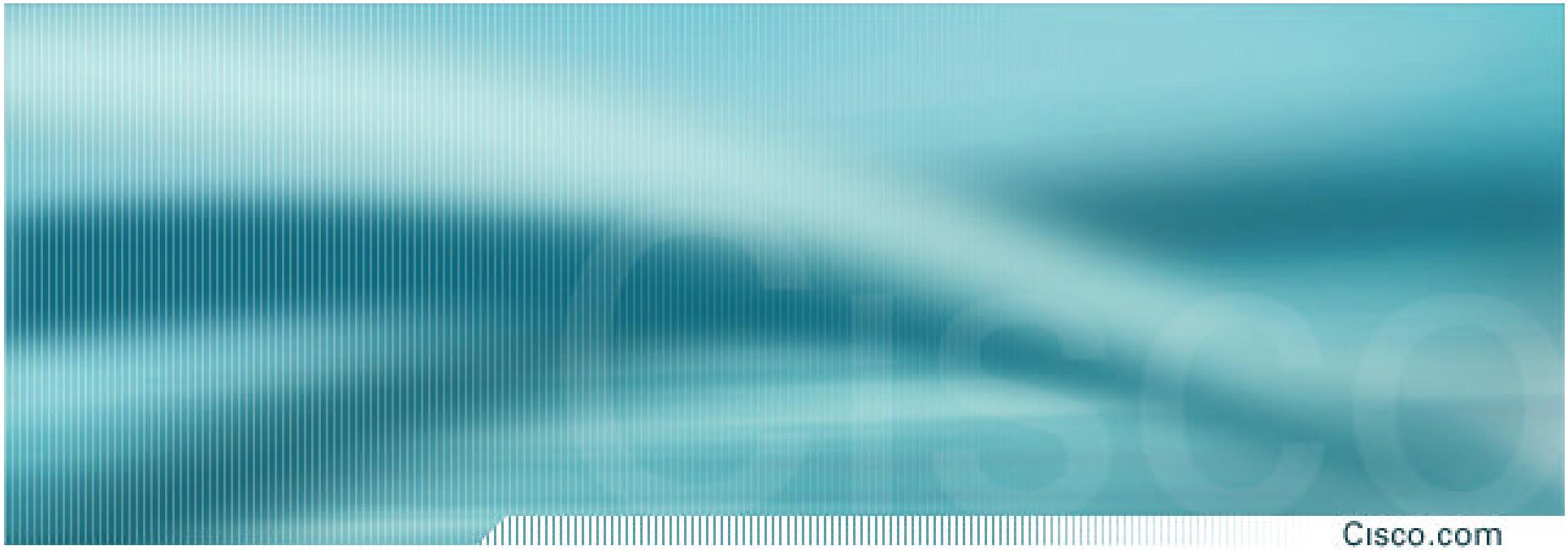
BGP/IGP model used in ISP networks

- **Model representation**



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

- 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**

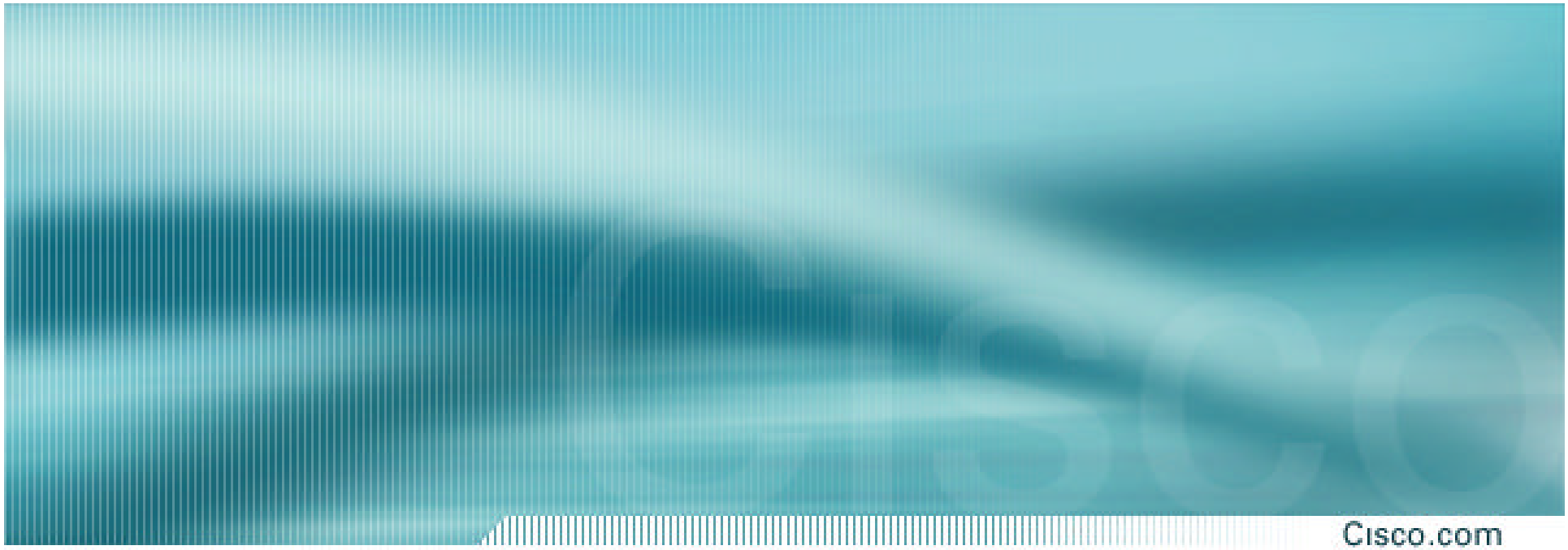
  no real reason to see subprefixes of allocated blocks in the Internet

  BUT there are currently >65000 /24s!

The Internet Today

- **Current Internet Routing Table Statistics**

BGP Routing Table Entries	117341
Prefixes after maximum aggregation	75164
Unique prefixes in Internet	56249
Prefixes larger than registry alloc	49462
/24s announced	65033
only 5612 /24s are from 192.0.0.0/8	
ASes in use	14056



Receiving Prefixes

Receiving Prefixes: From Downstreams

- **ISPs should:**
  - **accept only prefixes which have been assigned or allocated to their downstream customer**
  - **validate assignment/allocation in RIR databases**
- **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**

  e.g. multihoming, traffic engineering

- **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
```

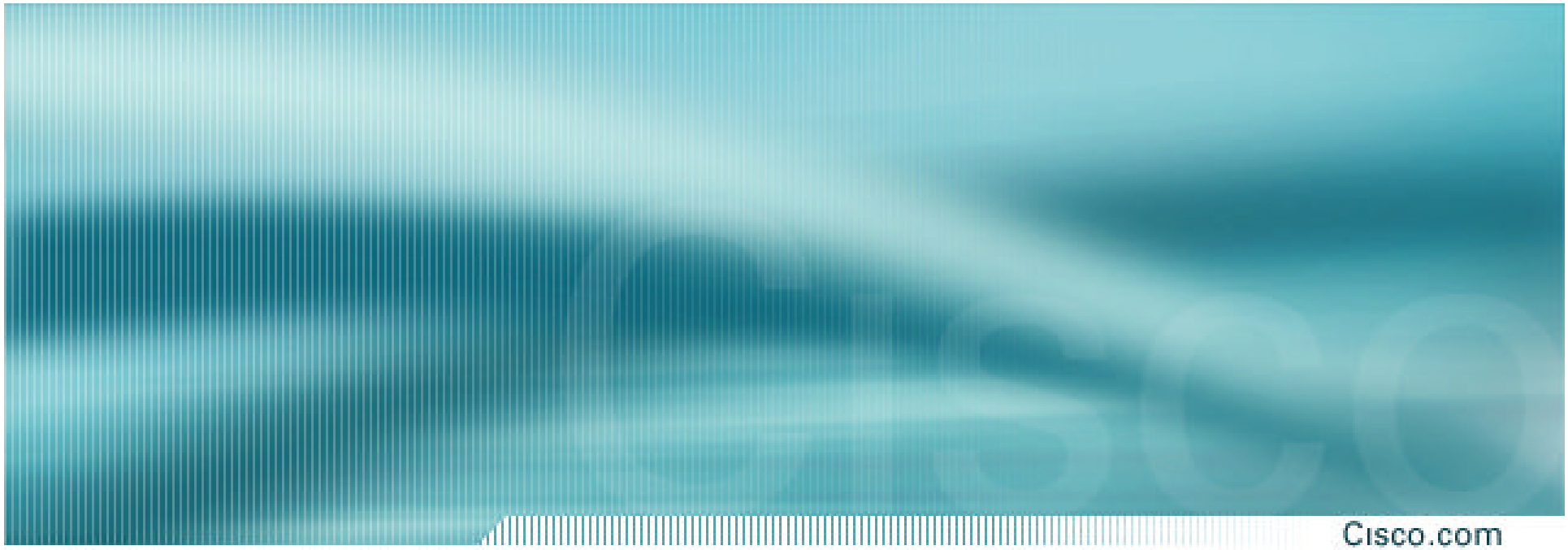
Receiving Prefixes: From Peers and Upstreams

- **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)**
- **Check Rob Thomas' list of "bogons"**
  - http://www.cymru.org/Documents/bogon-list.html

Receiving Prefixes – IOS Example

Cisco.com

```
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

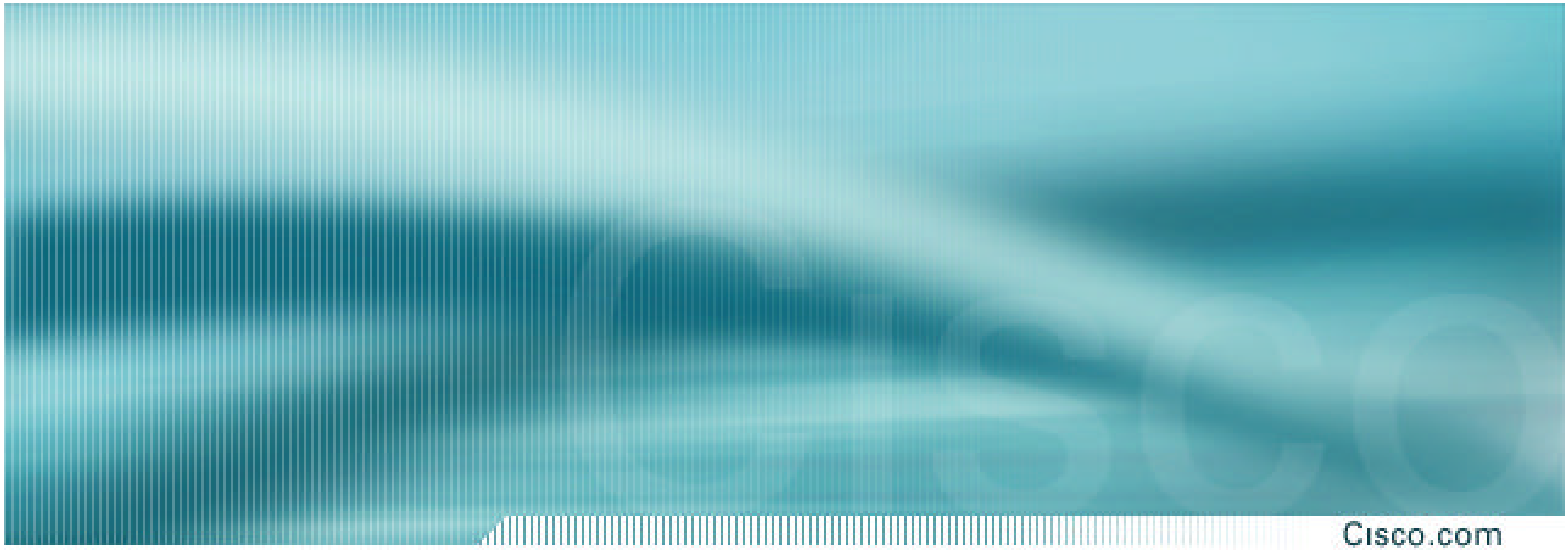
- **Point static route to customer interface**
- **Use BGP network statement**
- **As long as static route exists (interface active), prefix will be included in BGP**

Injecting prefixes into iBGP

- **interface flap will result in prefix withdraw and re-announce**
  - **use “ip route…permanent” if this is a concern**
  - **Static route always exists, even if interface is down ® prefix announced in iBGP**
- **many ISPs redistribute from static into BGP rather than network statement**
  - **Not recommended unless you understand why you need to do this**
  - **Uncontrolled redistribution (deliberate or mistaken) has led to many accidents on the Internet in the past**

BGP for Internet Service Providers

- Scaling BGP
- Best Current Practices
- Configuration Tips



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**

BGP Template – iBGP peers

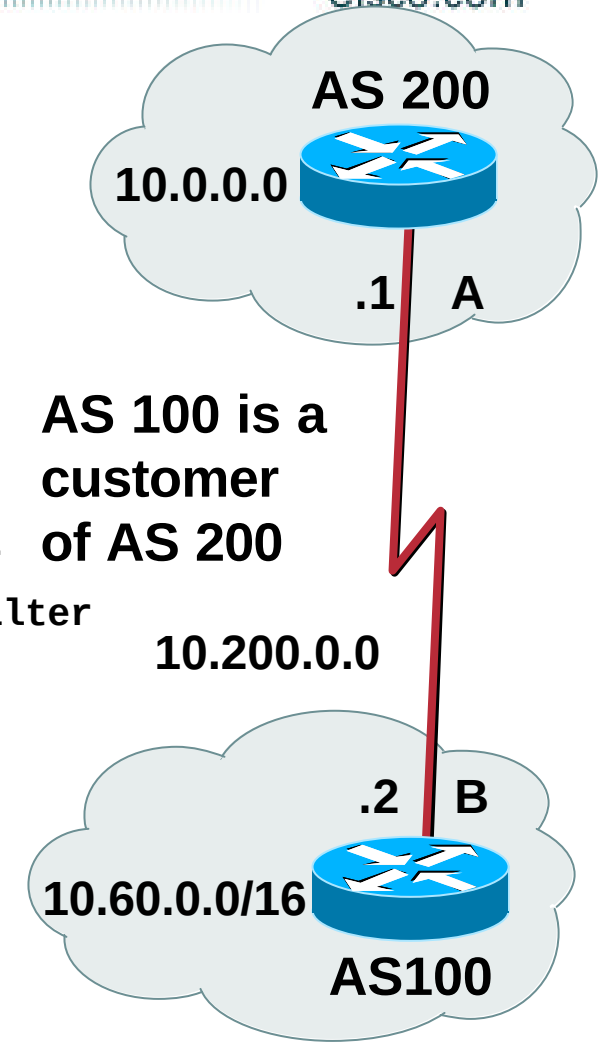

```
router bgp 100
neighbor internal peer-group
neighbor internal description ibgp peers
neighbor internal remote-as 100
neighbor internal update-source Loopback0
neighbor internal next-hop-self
neighbor internal send-community
neighbor internal version 4
neighbor internal password 7 03085A09
neighbor 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 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 130000 [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
  - Essential option for ISPs
- 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**

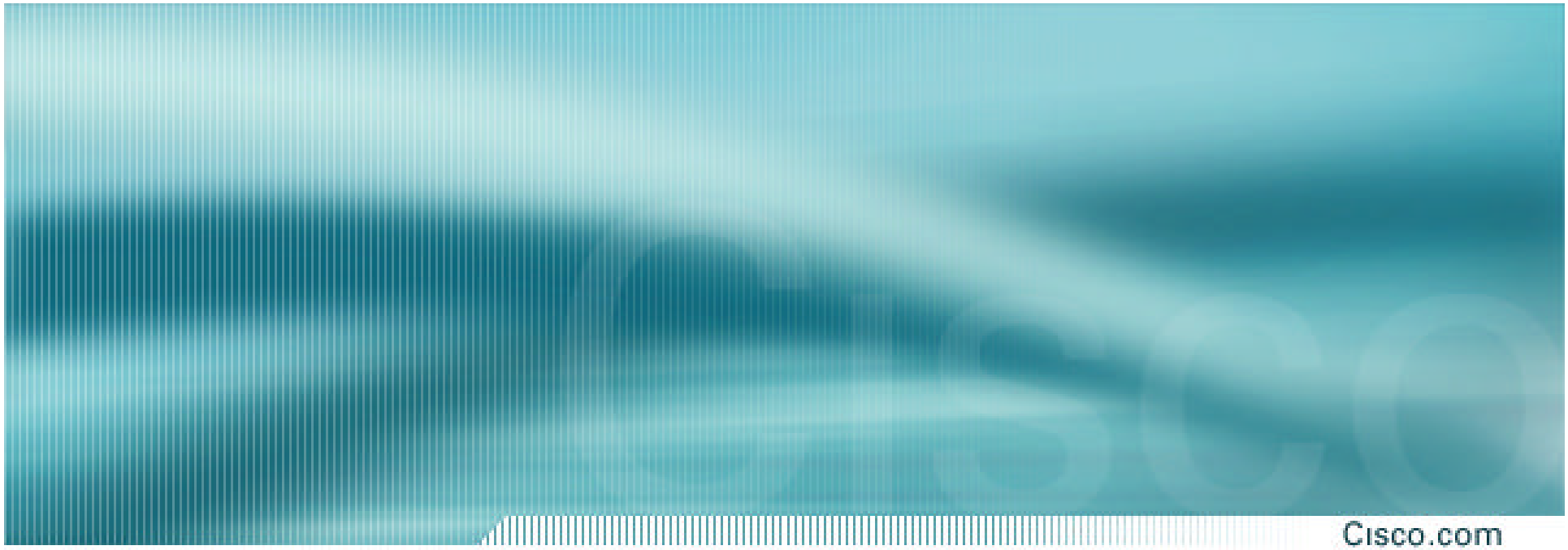
  distance bgp 200 200 200

Customer Aggregation: Guidelines

- **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**

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 for Internet Service Providers

End of Tutorial