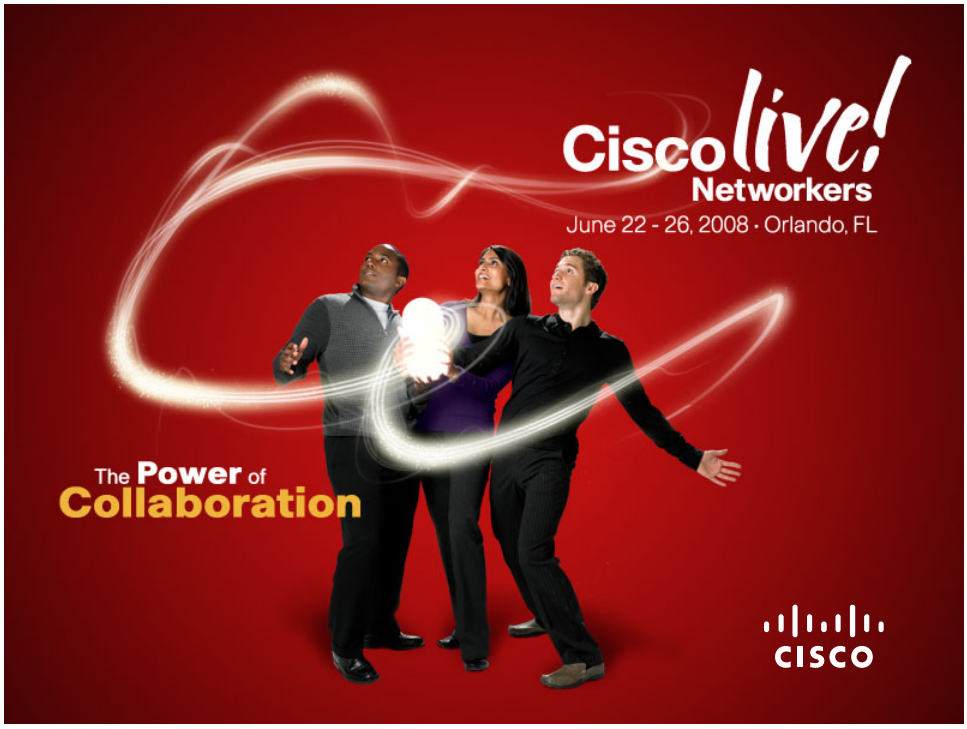
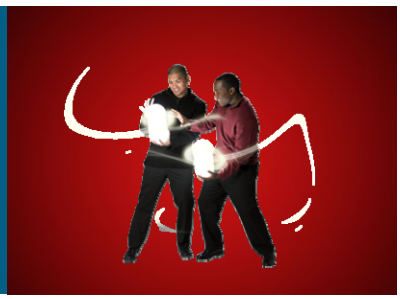


CISCO

Cisco live! Networkers

Scaling BGP



BRKRST-2321

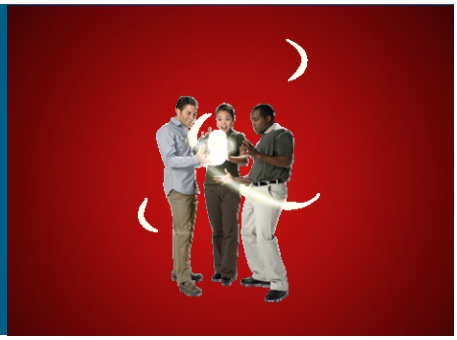
Housekeeping

- We value your feedback—don't forget to complete your online session evaluations after each session and complete the Overall Conference Evaluation which will be available online from Thursday
- Visit the World of Solutions
- Please remember this is a 'non-smoking' venue!
- Please switch off your mobile phones
- Please remember to wear your badge at all times including the Customer Appreciation Event

Scaling BGP in a Service Provider Network

- Full Mesh iBGP
- Route Reflectors
- Confederations
- Scaling BGP Updates

Full Mesh iBGP



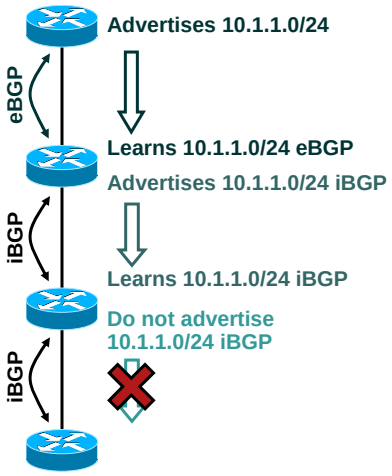
Full Mesh iBGP

- If a particular AS has multiple BGP speakers and is providing transit service for other AS's, then care must be taken to ensure a consistent view of routing within the AS. A consistent view of the interior routes of the AS is provided by the IGP used within the AS. For the purpose of this document, it is assumed that a consistent view of the routes exterior to the AS is provided by having all BGP speakers within the AS maintain IBGP sessions with each other.

draft-ietf-idr-bgp4-26.txt

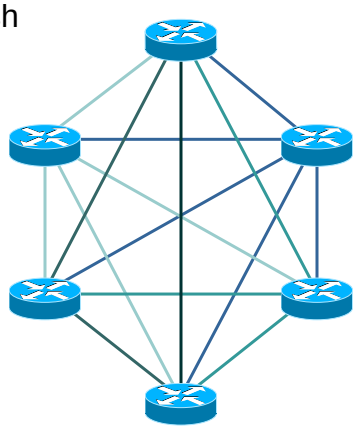
Full Mesh iBGP

- If a router learns a route from an iBGP peer, it will not re-advertise that route to another iBGP peer
- **Why?**
- Because BGP relies on the AS Path to prevent loops
- iBGP peers are in the same AS, so they do not add anything to the AS Path
- There's no way to tell if a route advertised through several iBGP speakers is a loop!



Full Mesh iBGP

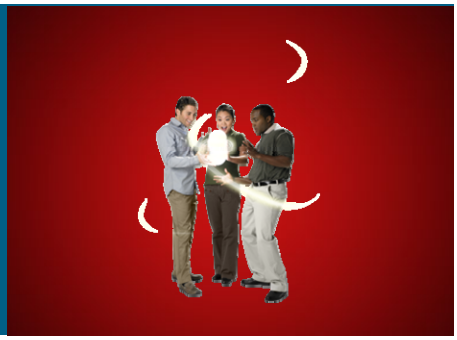
- How scalable is using a full mesh of iBGP speakers?
  - 2 speakers == 1 peer
  - 3 speakers == 3 peers
  - 4 speakers == 6 peers
  - 5 speakers == 10 peers
  - 6 speakers == 15 peers
- n(n-1)/2
- Not very scalable!



Route Reflector Basics

- If full mesh peering doesn't scale, what are our options?
- Route Reflectors
- Confederations

BGP Route Reflectors

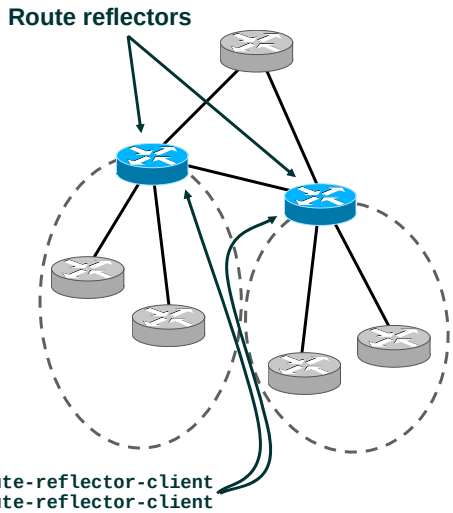


BGP Route Reflectors

- Route Reflector Basics
- Hierarchical Route Reflectors
- Deploying Route Reflectors
- Route Reflector Redundancy

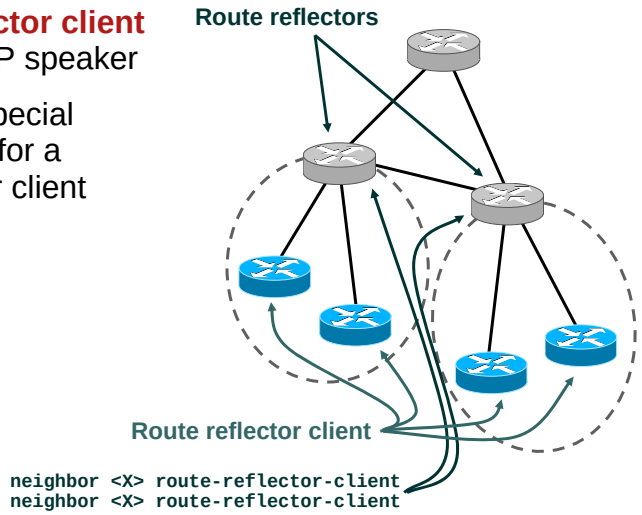
Route Reflector Basics

- A **route reflector** is an iBGP speaker that reflects routes learned from iBGP peers to other iBGP peers
- Route reflectors are designated by configuring some of their iBGP peers as route reflector clients



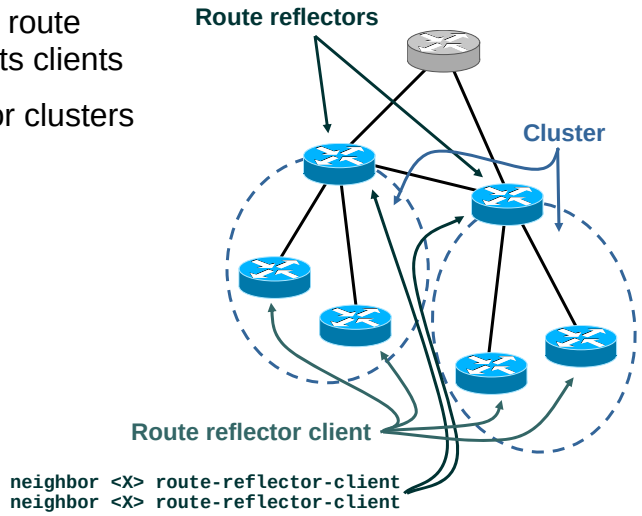
Route Reflector Basics

- A **route reflector client** is just an iBGP speaker
- There is no special configuration for a route reflector client



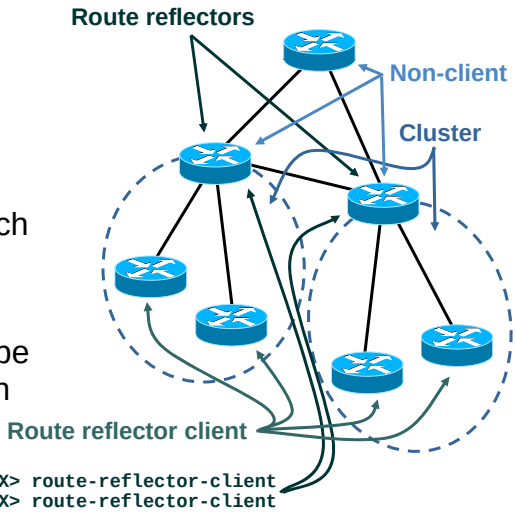
Route Reflector Basics

- A **cluster** is a route reflector and its clients
- Route reflector clusters may overlap



Route Reflector Basics

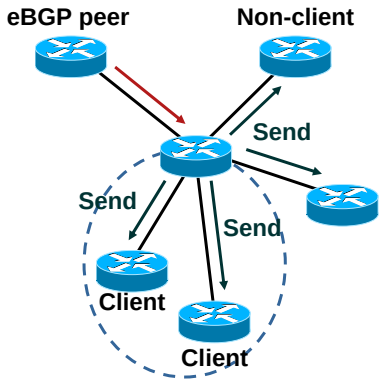
- A non-client is any route reflector iBGP peer that is not a route reflector client
- Each route reflector is also a non-client of each other route reflector in this network
- Route reflectors must be fully iBGP meshed with non-clients



Route Reflector Basics

If a Route Reflector Receives a Route from an eBGP Peer:

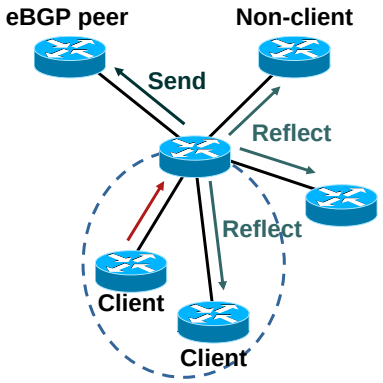
- Send the route to all clients
- Send the route to all non-clients



Route Reflector Basics

If a Route Reflector Receives a Route from a Client:

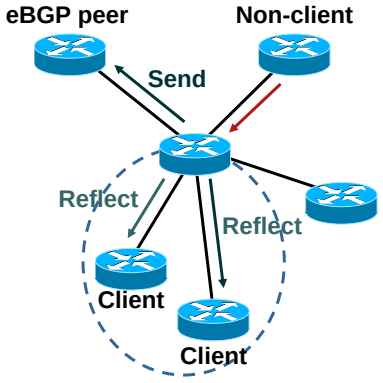
- Reflect the route to all clients
- Reflect the route to all non-clients
- Send the route to all eBGP peers



Route Reflector Basics

If a Route Reflector Receives a Route from a Non-Client:

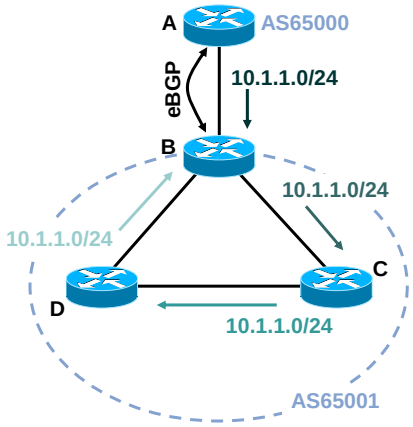
- Reflect the route to all clients
- Send the route to all eBGP peers



Route Reflector Basics

We Know That iBGP Doesn't Guarantee Loop Free Routing Through an AS

- B receives 10.1.1.0/24 with an AS Path of {65000}
- C receives 10.1.1.0/24 with an AS Path of {65000}
- D receives 10.1.1.0/24 with an AS Path of {65000}
- B receives the same route with the same attributes, setting up a loop!



Route Reflector Basics

- What we need is a mechanism to prevent loops within the AS!
- RFC2796, BGP Route Reflection, defines two BGP attributes to provide loop detection within an AS
- Originator ID

  Set to the router ID of the router injecting the route into the AS
- Cluster List

  Each route reflector the route passes through adds their cluster ID to this list. Cluster-id = Router ID by default

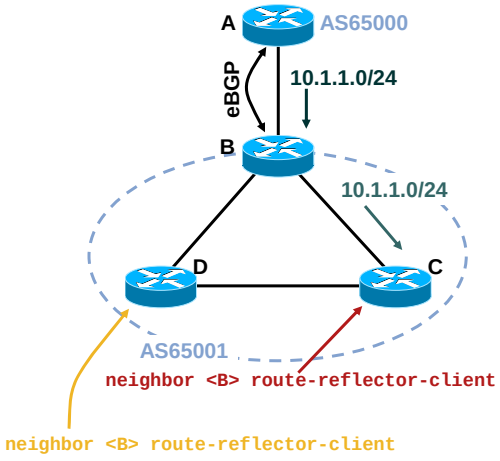
"bgp cluster-id A.B.C.D" command is not needed

Route Reflector Basics

- When reflecting a route, a route reflector always:
  - Creates a Cluster List if one doesn't exist
  - Adds its router ID (or the configured cluster ID) to the Cluster List
  - Adds the router ID of the peer it received the route from as the Originator ID
- When sending a route, a route reflector always follows normal BGP processing rules

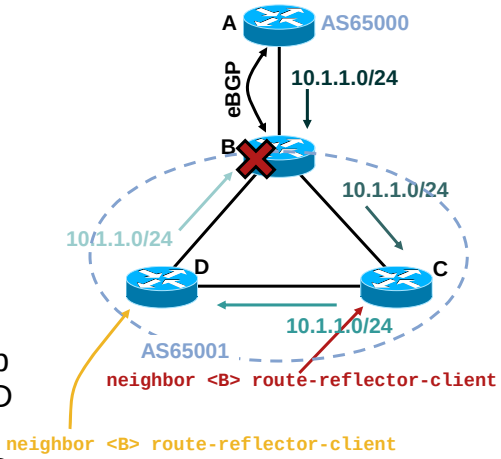
Route Reflector Basics

- B receives 10.1.1.0/24 with an AS Path of {65000}
- C receives 10.1.1.0/24 with an AS Path of {65000}, but adds B's Router ID as the Originator ID
- C also starts a Cluster List, and adds its own local Router ID into the list

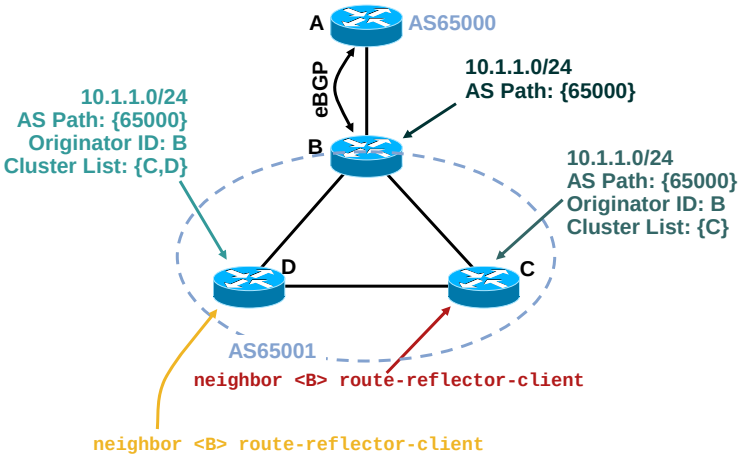


Route Reflector Basics

- Router D receives 10.1.1.0/24 with an AS Path of {65000} and an Originator ID of Router B
- Router D adds its own router ID to the Cluster list
- Router D sends the update to Router B. Router B detects the loop because the Originator ID in the update matches the Router ID of Router B



Route Reflector Basics



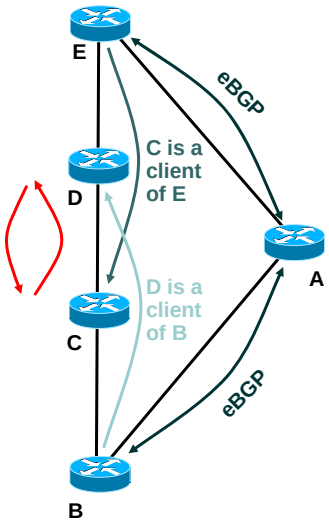
Route Reflector Basics

Router E:
BGP Next-Hop: Router A
Local Next-Hop: Router A
Set: Next-Hop-Self

Router D:
BGP Next-Hop: Router B
Local Next-Hop: Router C

Router C:
BGP Next-Hop: Router E
Local Next-Hop: Router D

Router B:
BGP Next-Hop: Router A
Local Next-Hop: Router A
Set: Next-Hop-Self



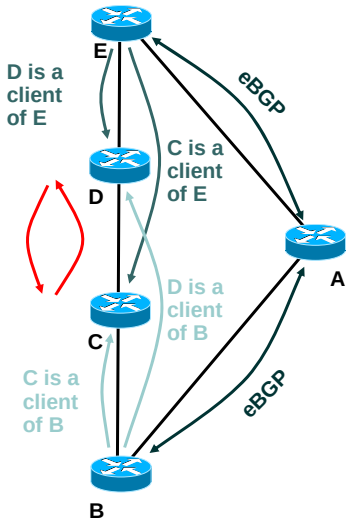
Route Reflector Basics

Router E:
BGP Next-Hop: Router A
Local Next-Hop: Router A
Set: Next-Hop-Self

Router D:
BGP Next-Hop: Router B
Local Next-Hop: Router E

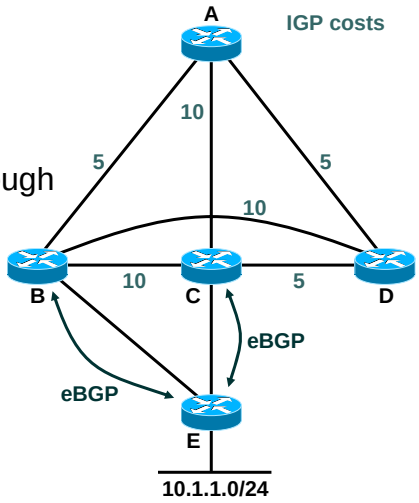
Router C:
BGP Next-Hop: Router B
Local Next-Hop: Router D

Router B:
BGP Next-Hop: Router A
Local Next-Hop: Router A
Set: Next-Hop-Self



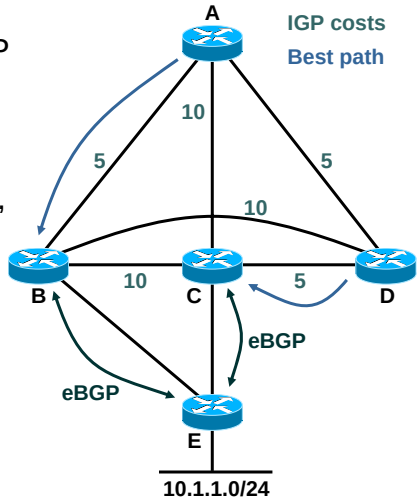
Route Reflector Basics

- Route reflectors can also cause routing to be different (or suboptimal) compared to full mesh iBGP
- E advertises 10.1.1.0/24 through eBGP to both B and C
- The local preference, MED, AS Path length, and all other attributes are the same for 10.1.1.0/24 at both B and C



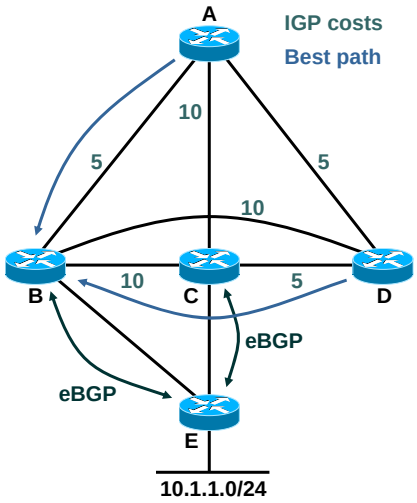
Route Reflector Basics

- Assume A, B, C, and D are configured for full mesh iBGP
- A chooses B as its exit point because of the IGP cost
- D chooses C as its exit point, because of the IGP cost



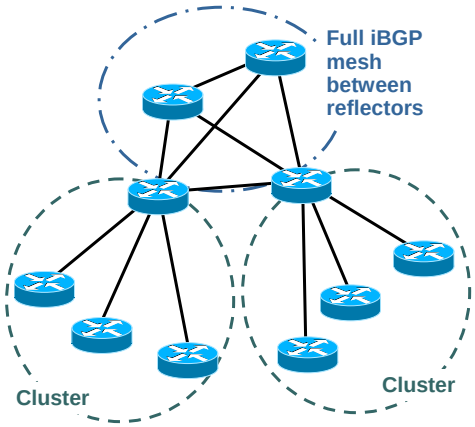
Route Reflector Basics

- Assume B, C and, D are configured as route reflector clients of A
- A chooses B as its best path because of the IGP cost
- A reflects this choice to C, but C chooses its locally learned eBGP route over the internal through B
- A reflects this choice to D, and D chooses the path through B, even though the path through C is shorter



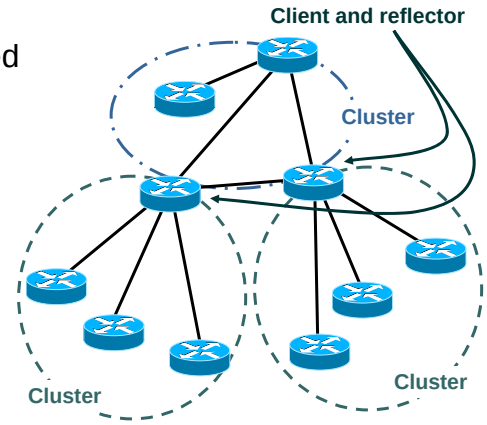
Hierarchal Route Reflectors

- All of the route reflectors will need to be fully meshed
  - Reflectors still follow the normal rules of iBGP route propagation between themselves
- This full iBGP mesh between reflectors can still contain so many routers that it presents a scaling problem



Hierarchal Route Reflectors

- To resolve this, route reflectors can be deployed in a hierarchy
- A single router can be a reflector client and a reflector



Hierarchal Route Reflectors

- An unlimited number of tiers can be used
- The edges of route reflector tiers are a natural place to reduce the amount of routing information being carried in the lower tiers
- The same topology rule applies: The reflector topology must follow the physical topology to prevent loops and black holes
- Suboptimal routing can actually be worse, and harder to figure out

Deploying Route Reflectors

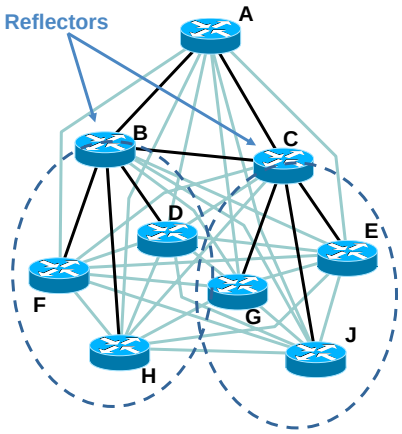
- Use the divide and conquer approach to convert from a full iBGP mesh to route reflectors
- Divide network into multiple clusters, using the physical topology as a guide to the logical divisions
- Pick out one router to act as the reflector in each cluster, making certain reflection follows the physical topology
- Remove redundant iBGP sessions as you configure reflectors in each cluster

Deploying Route Reflectors

- If you're going to use hierarchal route reflectors, do the outer edge first, leave the core full mesh iBGP until the outer edge is done
- Continue using a single IGP—the next-hop is unmodified by reflectors unless set via an explicit route-map

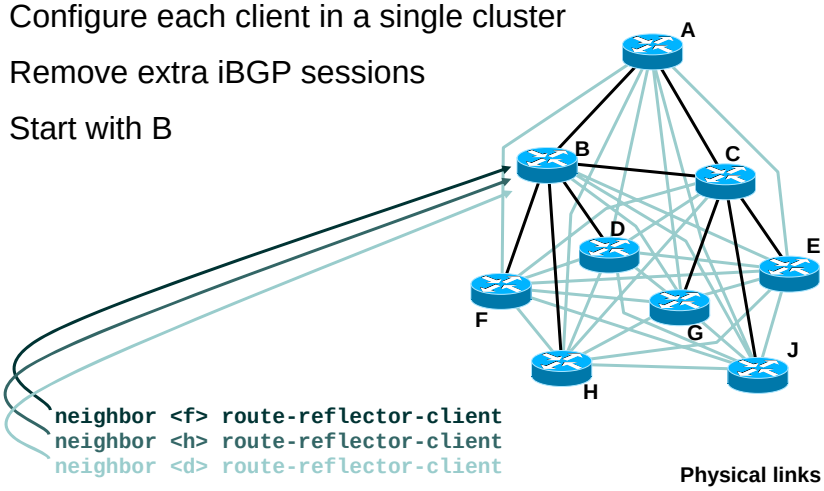
Deploying Route Reflectors

- This small network has nine routers, and 36 iBGP sessions
- First, choose clusters using the physical topology as a guide
- Next choose reflectors based on the physical topology



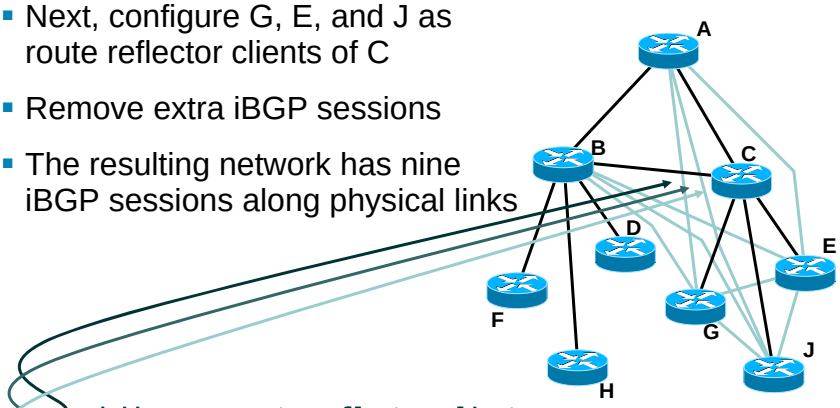
Deploying Route Reflectors

- Configure each client in a single cluster
- Remove extra iBGP sessions
- Start with B



Deploying Route Reflectors

- Next, configure G, E, and J as route reflector clients of C
- Remove extra iBGP sessions
- The resulting network has nine iBGP sessions along physical links

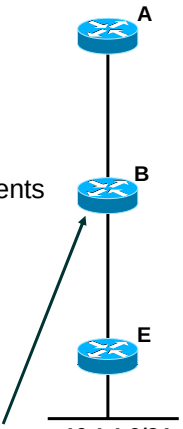


Route Reflector Redundancy

- A client may peer with more than one reflector, in different clusters
  - A client that peers to only one reflector has a single point of failure
  - Clients should peer to at least two reflectors to provide redundancy
- How many reflectors should a single client be peered to?
- Should redundant reflectors be in the same cluster or should they be in separate clusters?

Route Reflector Redundancy

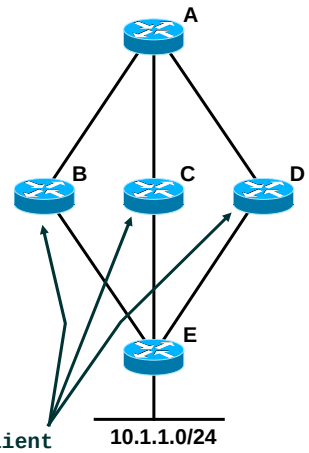
- How many route reflectors should a single client be peered to?
- Two considerations are important:
  - Network configuration and management
  - Router memory and processing requirements
- If A is the client of only one reflector, it only receives one copy of the route to 10.1.1.0/24



`neighbor <a> route-reflector-client`

Route Reflector Redundancy

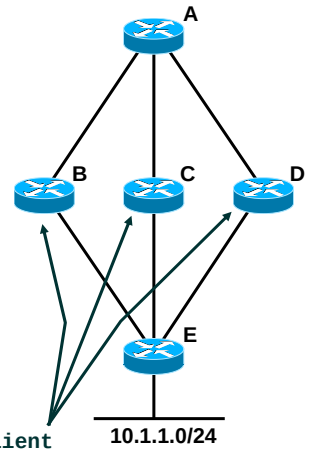
- Each new route reflector A becomes a client of adds more configuration and management
- Each new route reflector A becomes a client of adds another path to 10.1.1.0/24
- This increases the amount of memory A requires to operate, and also increases A's processing requirements



`neighbor <a> route-reflector-client`

Route Reflector Redundancy

- Each new client B, C, and D are peered to also increased their processing requirements
- At some point, the additional reflectors will stop adding to the resilience of the network, and make management and memory requirements similar to a full iBGP mesh



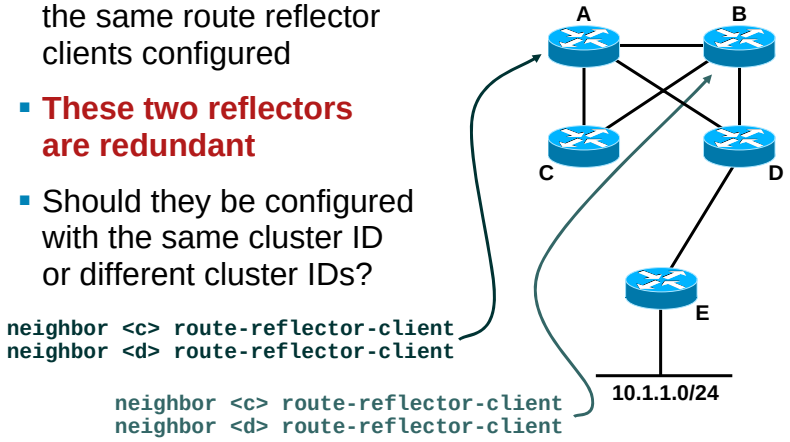

```
neighbor <a> route-reflector-client
```


Route Reflector Redundancy

- Some redundancy is needed
- Too much burns memory on RRCs because the client learns the same information from each RR
- Also burns memory on the RRs because they learn multiple paths for each route introduced by a RRC
- **Two or three route reflectors per client should be plenty**

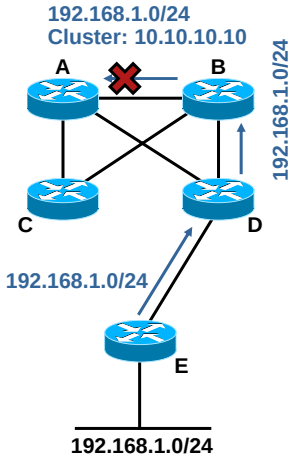
Route Reflector Redundancy

- Assume A and B have the same route reflector clients configured
- **These two reflectors are redundant**
- Should they be configured with the same cluster ID or different cluster IDs?



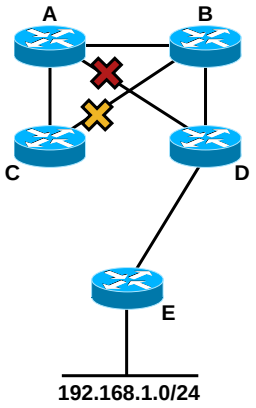
Route Reflector Redundancy

- **Assume A and B are using the same cluster ID, 10.10.10.10**
- E advertises 192.168.1.0/24 to D
- D sends this route to its reflector, B
- B adds a Cluster List and the Originator ID, and reflects the route to A and C
- When A receives this route, it notes its local cluster ID is already in the Cluster List (since A and B have the same cluster ID), and rejects the route



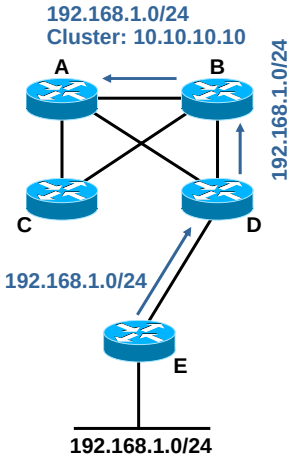
Route Reflector Redundancy

- If the A to D link fails, A won't have any path to 192.168. 1.0/24, since it is rejecting the route from B
- If the B to C link fails, C won't have any path to 192.168.1.0/24, since A is rejecting the route from B, and won't reflect it to C



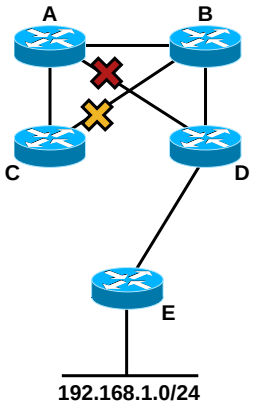
Route Reflector Redundancy

- Cisco recommendation is to configure A and B with unique cluster IDs
- Now, when A receives B's reflected route, it will keep the route, since the cluster ID in the Cluster List doesn't match its own cluster ID
- A will run the BGP bestpath algorithm, and advertise its path through B or its path through D to C



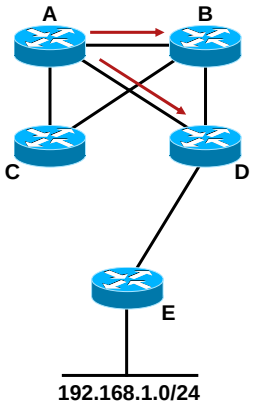
Route Reflector Redundancy

- If the A to D link fails, A will still have the path through B to reach 192.168.1.0/24
- If the B to C link fails, C will still have the path through A to reach 192.168.1.0/24
- **This provides full redundancy but at a cost ☺**

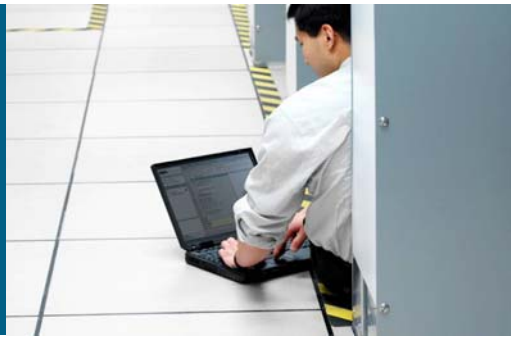


Route Reflector Redundancy

- A now has two routes to 192.168.1.0/24, one through D, and one through B
- Each additional path A must hold and process adds additional memory and processor overhead



BGP Confederations



BGP Confederations

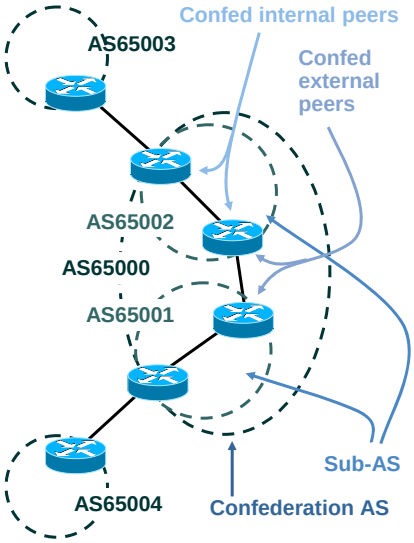
- Confederation Basics
- Deploying Confederations
- Comparing Confederations with Route Reflectors

Confederation Basics

- Confederations provide another way of scaling BGP within the AS
- Rather than adding new attributes to prevent loops within an AS, Confederations add more information to the AS Path
- RFC3065 defines Confederations

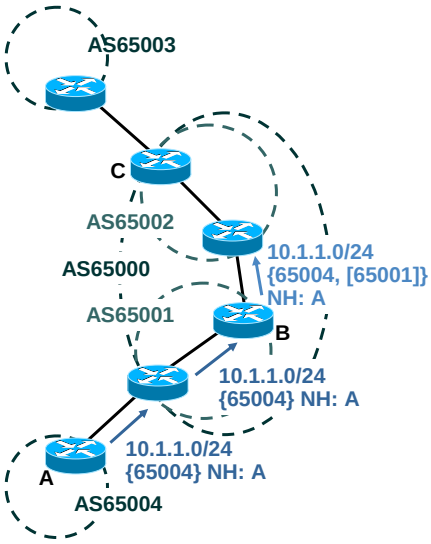
Confederation Basics

- In a confederation, the AS is split into multiple AS'
  - The "outer" AS is called the **Confederation AS**
  - And "inner" AS is a sub-AS
  - Each **sub-AS** has its own AS number
- Peers in the same sub-AS are **Confed Internal Peers**
- Peers in different sub-AS' are **Confed External Peers**



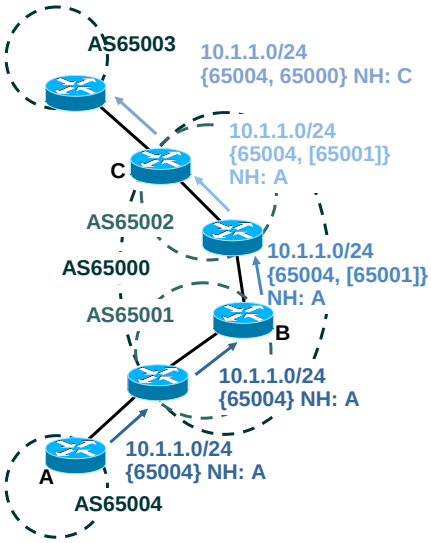
Confederation Basics

- When A sends the route, AS65004 is included in the AS Path, and the next hop is set to A
- Within sub-AS 65001, the attributes remain the same
- B creates an **AS Confederation Sequence,** and adds its local sub-AS, 65001, to the AS Path



Confederation Basics

- Within sub-AS 65002, the route's attributes remain the same
- The next hop remains A throughout the confederation, even though the route is passing through multiple sub-AS'
- At C, the AS Confederation Path is stripped off, and the Confederation AS is added to the AS Path

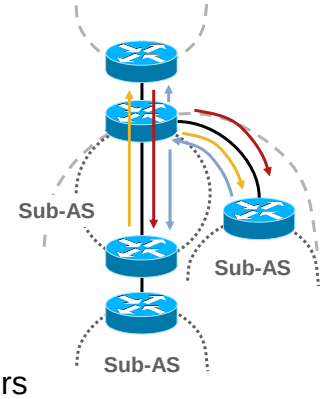


Confederation Basics

- At confed external peers, three attributes are preserved:
  - Next Hop
  - Local Preference
  - MED
- This should produce consistent routing throughout the confederation AS
- Since the Next Hop isn't changed, a single confederation AS should run a single IGP

Confederation Basics

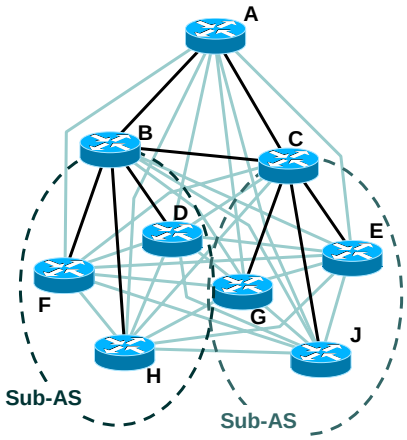
- A route learned from an eBGP peer is advertised to all confed external and internal peers
- A route learned from a confed internal peer is advertised to all confed external and eBGP peers
- A route learned from a confed external peer is advertised to all confed internal and eBGP peers



eBGP learned route
Confed internal learned route
Confed external learned route

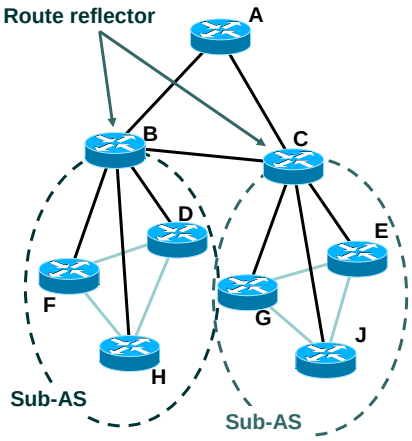
Confederations Basics

- All of the BGP speakers within a confederation must still maintain a full mesh of iBGP connectivity
- Confederations reduce the iBGP mesh, since routers in different sub-AS' usually don't peer with each other (unless they are at the sub-AS border

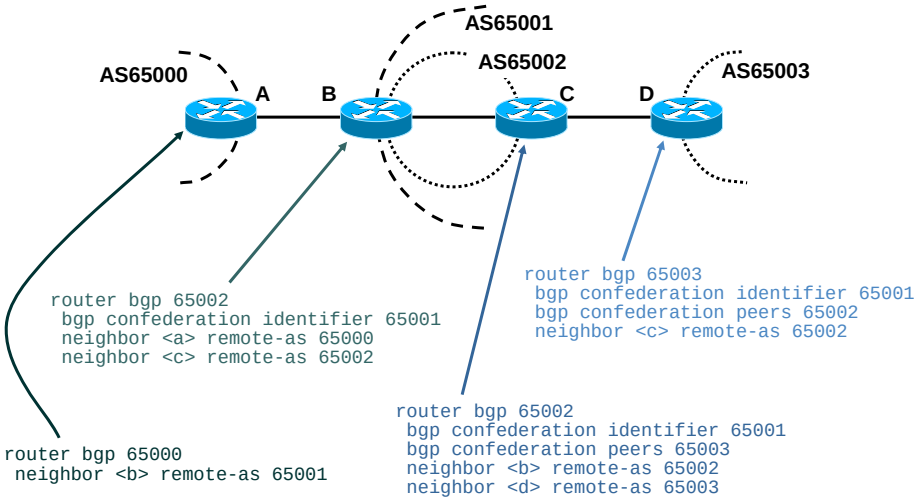


Confederations Basics

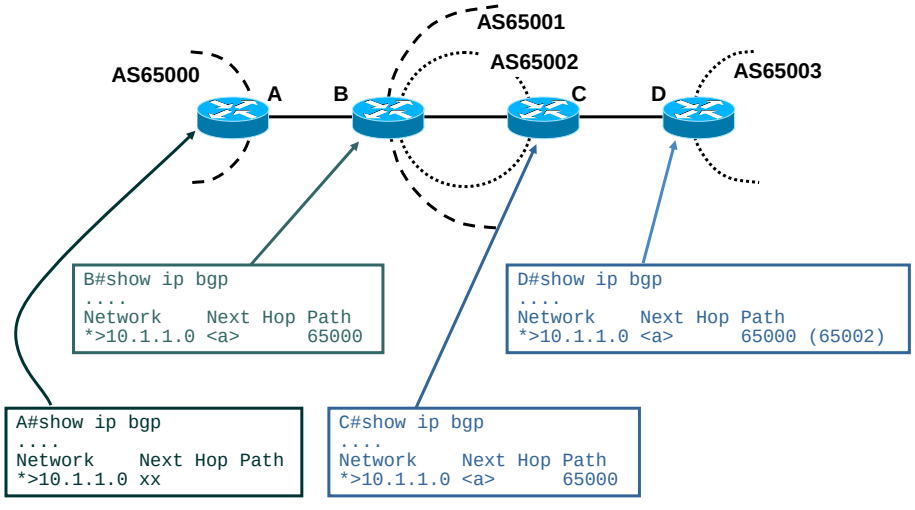
- Route reflectors can be used within confederations to further reduce network complexity
- This is becoming more common as service provider networks grow in size



Confederation Basics

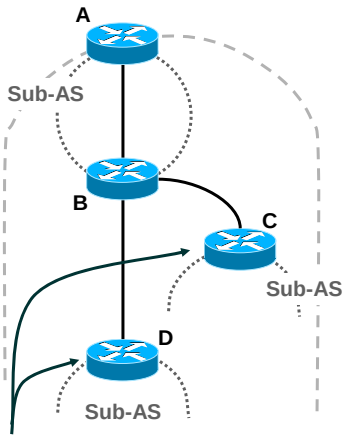


Confederation Basics



Deploying Confederations

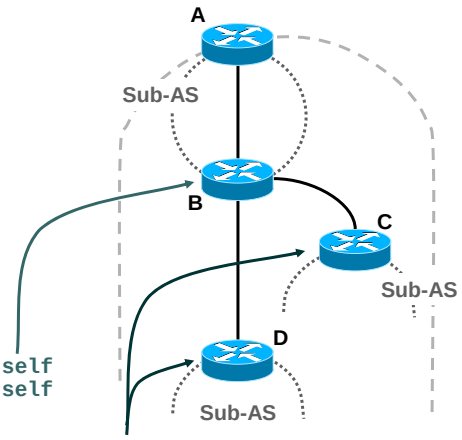
- Normally the Next Hop isn't changed when a route is advertised between sub-AS'
- This allows (or requires) the entire confederation to run one IGP



Must be able to reach A

Deploying Confederations

- It's possible to configure B so it sets the Next Hop to itself when advertising routes to C and D
- **Why would you want to do this?**

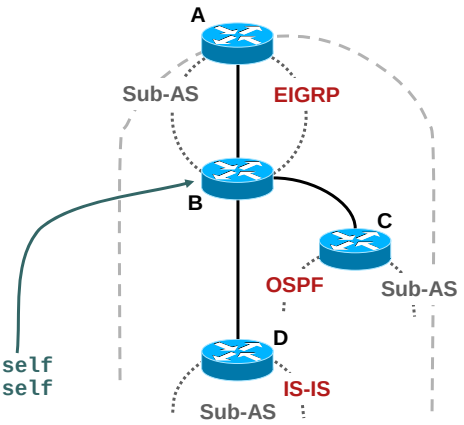



```
neighbor <c> next-hop self
neighbor <d> next-hop self
```

Must be able to reach B

Deploying Confederations

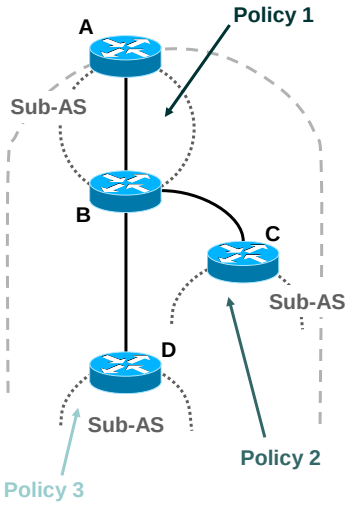
- So each sub-AS can run a different IGP
- This breaks the confederation up from an IGP perspective, as well as a BGP perspective, increasing the scaling properties of the network




```
neighbor <c> next-hop self
neighbor <d> next-hop self
```


Deploying Confederations

- Each sub-AS can also have its own policies
  - MED acceptance or stripping
  - Local Preference settings
  - Route Dampening
  - Etc.
- Policies can also be applied between sub-AS'
- This is very difficult to do with route reflectors (if not impossible)

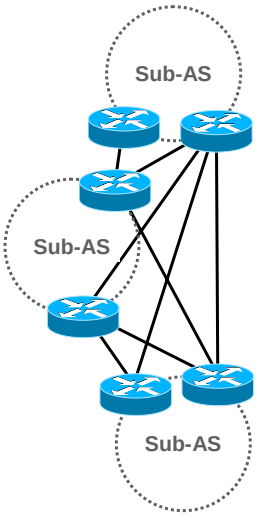


Deploying Confederations

- What does having separate policies and IGPs per Sub-AS buy you?
- Ease of transition when purchasing another company!!
- Simply add the new network as another Sub-AS
- Acquired company can keep the same IGP
- Acquired company can keep old policies with their customers without a policy change in your entire network

Deploying Confederations

- To provide redundancy within a confederation, provide multiple links between the sub-AS'
- Each link added, however, increases the amount of memory and processing power required on each sub-AS border router
- Two to three connections between sub-AS' is normally enough to provide the required redundancy



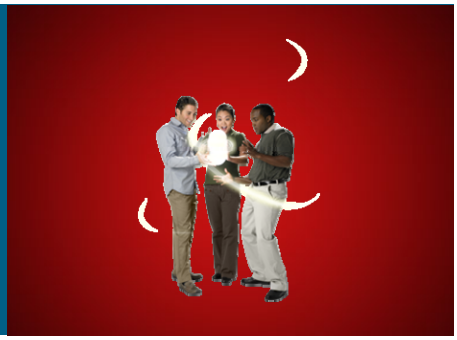
Confederations—Deployment

- Unlike Route Reflectors, there is no graceful way to migrate a network from a full mesh to confederations
- Not a problem if building a network from scratch, otherwise downtime needs to be scheduled
- Good solution if:
  - Multiple sets of policies need to be used in a single AS
  - For companies that make a lot of acquisitions

Comparing Confederations

	Confederation	Route Reflector
Loop prevention	AS Confederation Set	Originator/Cluster ID
Break up a single AS	Sub-AS'	Clusters
Redundancy	Multiple connections between sub-AS'	Client connects to several reflectors
External Connections	Anywhere in the network	Anywhere in the network
Multilevel Hierarchy	Reflectors within sub-AS'	Clusters within clusters
Policy Control	Along outside borders and between sub-AS'	Along outside border
Scalability	Medium; still requires full iBGP within each sub-AS	Very high
Migration	Very difficult (impossible in some situations)	Moderately easy

Scaling BGP Updates



Scaling BGP Updates

- Aggregation
- Peer Groups
- Input Queue Tuning
- Path MTU Discovery

Aggregation

- **Why aggregate?**
- Reduce number of Internet prefixes

  Advertise only your CIDR block

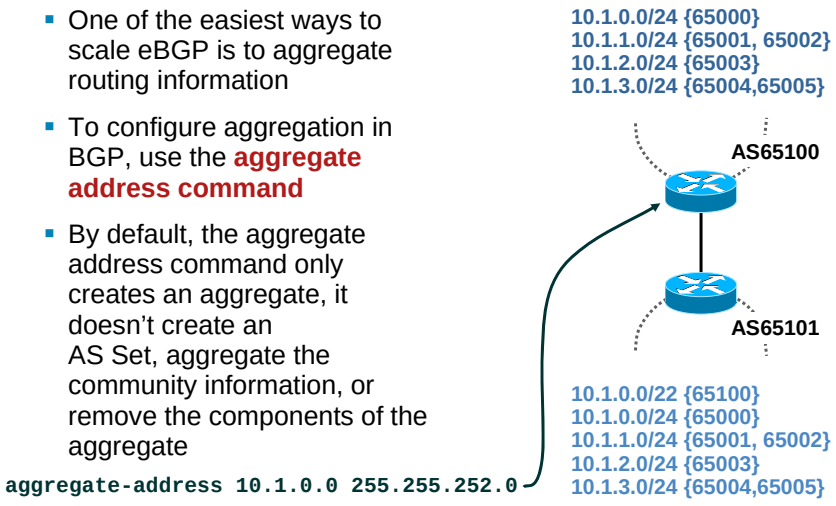
  According to some studies, about 50% of the current Internet routing table represents “leakage past aggregates”
- Increase stability

  If you aggregate properly, the aggregate will remain stable even if specific components of the aggregate come and go

  This can be very important when your neighbors have dampening configured

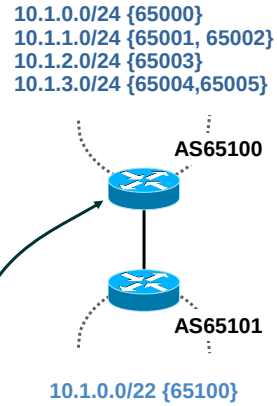
Aggregation

- One of the easiest ways to scale eBGP is to aggregate routing information
- To configure aggregation in BGP, use the **aggregate address command**
- By default, the aggregate address command only creates an aggregate, it doesn’t create an AS Set, aggregate the community information, or remove the components of the aggregate



Aggregation

- Adding the keyword **summary-only** causes BGP to remove the components of the aggregate
- Components are the longer length prefixes that fall within the aggregate's range

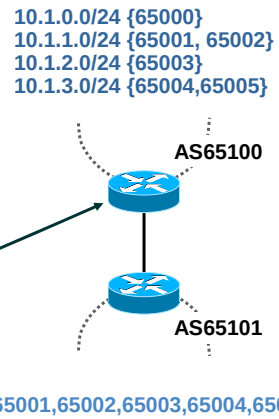



```
aggregate-address 10.1.0.0 255.255.252.0 summary-only
```


Aggregation

Adding the Keyword **as-set** Causes BGP to:

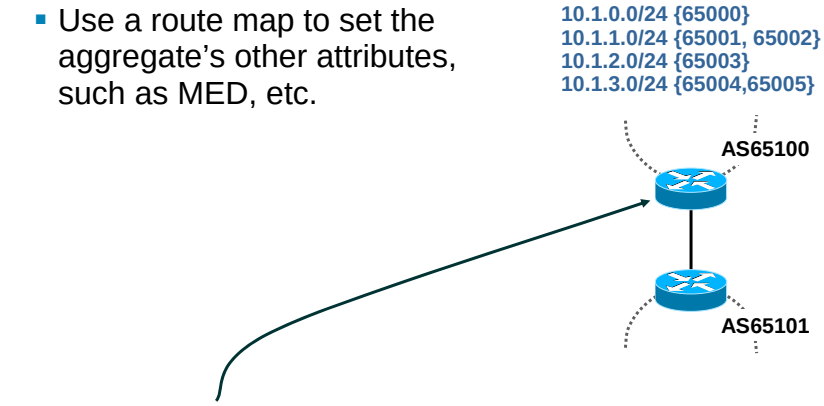
- Create an AS Set containing all the AS Paths within the aggregate's components
- Merge all the communities attached to each aggregate into one set of communities in the aggregate




```
aggregate-address 10.1.0.0 255.255.252.0 summary-only as-set
```

Aggregation

- Use a route map to set the aggregate's other attributes, such as MED, etc.




```
aggregate-address 10.1.0.0 255.255.252.0 summary-only as-set route-map ....
```


Aggregation

- If you configure aggregation without the **as-set** keyword, BGP will mark the route as an **Atomic Aggregate**
- Indicates loss of AS Path information
- Must not be removed once set
- Informational attribute only

  It doesn't really do anything

Aggregation

- BGP always adds the **Aggregator** attribute to any created aggregate route
- AS number and IP address of router generating aggregate
- Useful for troubleshooting

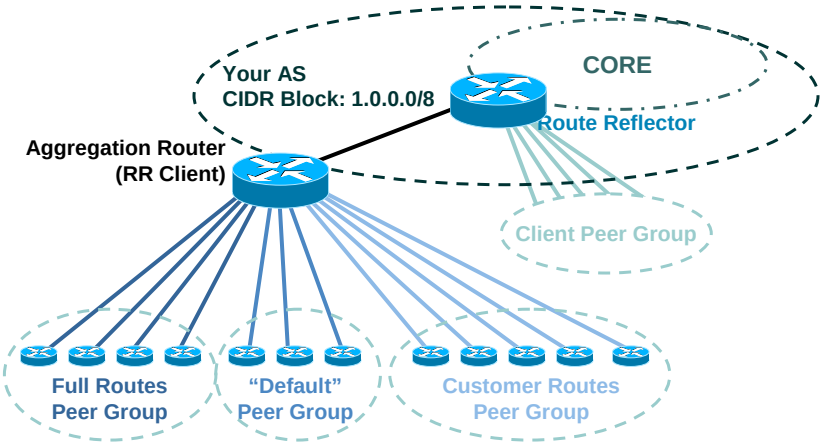
Peer Groups

- Group peers for:
  - Ease of administration
  - Scaling
- For ease of administration, consider:
  - Offering customers a few options in the number of routes they receive, rather than filtering per customer
  - Classifying peering arrangements with other providers so you only manage two or three types of connections, rather than managing each peering point independently

Peer Groups

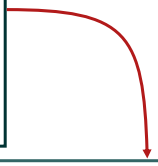
- For customer define three peer groups (or peer templates):
  - cust-default—send default route only
  - cust-cust—send customer routes only
  - cust-full—send full Internet routes
- Identify routes via communities
  - 1:100 = Routes from customers
  - 1:80 = Routes from fellow ISP

Peer Groups



Peer Groups

```
router bgp 65000
 neighbor 10.1.1.1 remote-as 65001
 neighbor 10.1.1.1 route-map cust-receive in
 neighbor 10.1.1.1 route-map cust-default out
 neighbor 10.1.1.1 send-community
 neighbor 10.1.1.2 remote-as 65002
 neighbor 10.1.1.2 route-map cust-receive in
 neighbor 10.1.1.2 route-map cust-default out
 neighbor 10.1.1.2 send-community
 neighbor 10.1.1.3 remote-as 65003
 neighbor 10.1.1.3 route-map cust-receive in
 neighbor 10.1.1.3 route-map cust-default out
 neighbor 10.1.1.3 send-community
```



```
router bgp 65000
 neighbor 10.1.1.1 remote-as 65001
 neighbor 10.1.1.1 peer-group cust-default
 neighbor 10.1.1.2 remote-as 65002
 neighbor 10.1.1.2 peer-group cust-default
 neighbor 10.1.1.3 remote-as 65003
 neighbor 10.1.1.3 peer-group cust-default
 neighbor cust-default route-map cust-receive in
 neighbor cust-default route-map cust-default out
 neighbor cust-default send-community
```

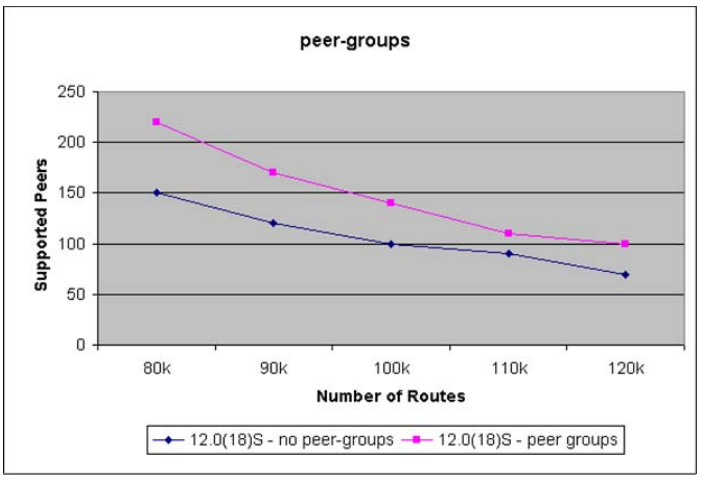

Peer Groups

- Peer groups also improve scaling
- Advertising 100,000+ routes to hundreds of peers is a big challenge from a scalability point of view
  - Each packet to each peer must be individually generated
  - Each packet to each peer must be individually transmitted
- Peer-groups make it easier for BGP to advertise routes to large numbers of peers by addressing these two problems

Peer Groups

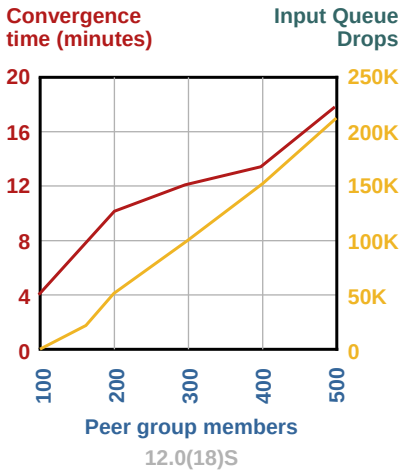
- Update generation without peer groups
  - BGP table is walked for every peer
  - Prefixes are filtered through outbound policies
  - Updates are generated and sent to this one peer
- Update generation with peer groups
  - A peer-group leader is elected for each peer group
  - The BGP table is walked for the leader only, prefixes are filtered through outbound policies
  - Updates are generated, sent to the peer-group leader, and replicated for peer-group members
- Peer Groups perform the grouping dynamically behind the scenes divorcing the update generation from the policy configuration

Peer Groups



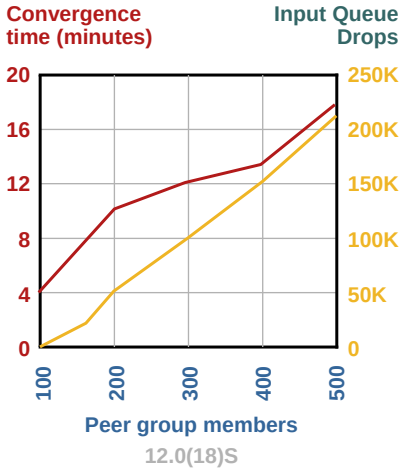
Input Queue Tuning

- Using the default input queues on all interfaces…
- As the number of peers in a single peer group increases, the number of input queue drops on the router's interfaces increases
- This makes BGP converge more slowly, since each drop represents a TCP segment resend and slow start



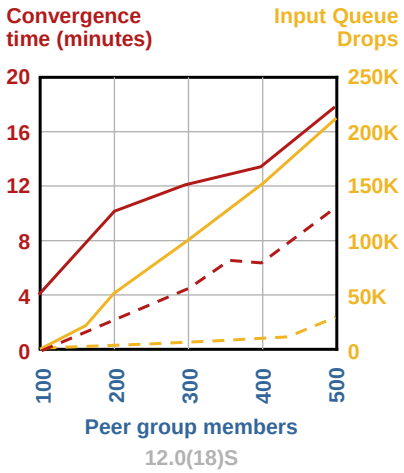
Input Queue Tuning

- Most of the drops we see are TCP acknowledgements
- This indicates the router with all the peer groups can't keep up with the amount of inbound traffic flow, rather than outbound

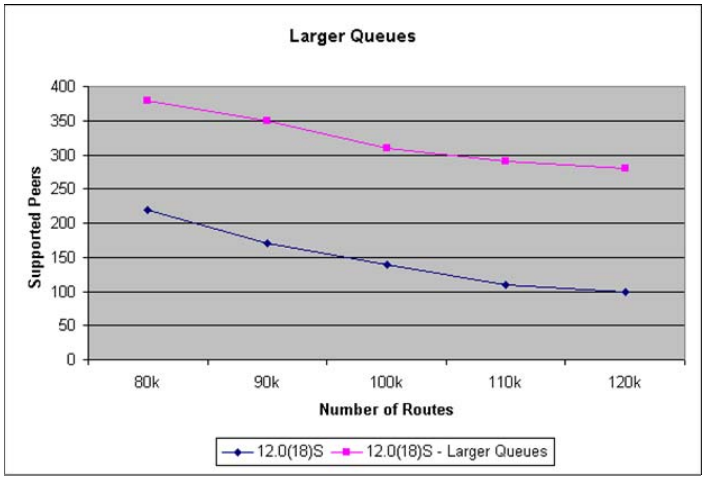


Input Queue Tuning

- Setting the input queues on all interfaces with peers in this peer group improves the situation dramatically
- We can converge the same number of peers in a peer group in about nine minutes, rather than about 18 minutes
- We only see about 20,000 packets dropped, rather than about 220,000 packets

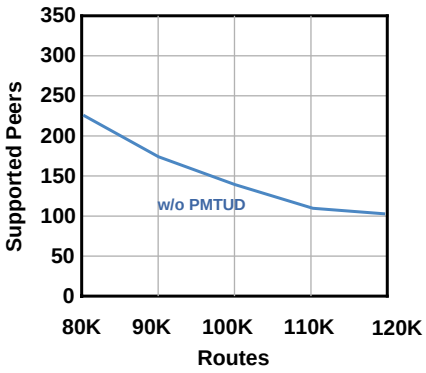


Larger Input Queues



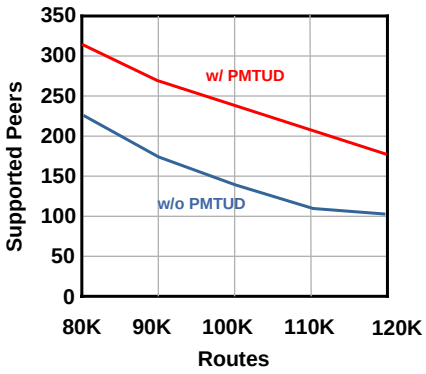
Path MTU Discovery

- The default maximum segment size for TCP in Cisco IOS Software is 536 bytes
  - Older links with a default MTU of 536 bytes are still in use in many places
- A low maximum segment size hampers TCP's performance badly



Path MTU Discovery

- Most links have an MTU larger than 536 bytes
- Configuring path MTU discovery between BGP peers can provide dramatic results in the speed of convergence
- **Both of these tests are on 12.0(18)S, one with path MTU discovery enabled, the other without**



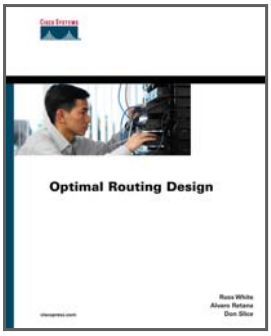
Recommended Reading for BRKIPM-2007

- BGP Design and Implementation
- Internet Routing Architectures
- Cisco BGP-4 Command and Configuration Handbook

Available Onsite at the Cisco Company Store

Further Reading

- Optimal Routing Design



ISBN 1587051877

Meet The Expert

- To make the most of your time at Cisco Networkers 2008, schedule a face-to-face meeting with a top Cisco expert.
- Designed to provide a “big picture” perspective as well as “in-depth” technology discussions, these face-to-face meetings will provide fascinating dialogue and a wealth of valuable insights and ideas.
- Visit the Meeting Centre reception desk in the CCIB Level 1, Room 117.

Q and A

Cisco Networkers Connect

- More questions on this session?
- Like advice on your next session?
- Where to go for dinner this evening?
- Find your colleagues…
- For answers to these questions and more, log-on to:

Cisco Networkers Connect via the Onsite Portal

- Challenge your creativity and record a video! *
- Best video wins an Apple iPhone

Prize Draw on Thursday at 12.00 at the Cisco Networkers Connect Video Station in the CCIB Foyer

*Onsite Recording Facilities at Cisco Networkers Connect Video Station

Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Passport points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

Don't forget to activate your Cisco Live virtual account for access to all session material on-demand and return for our live virtual event in October 2008.

Go to the Collaboration Zone in World of Solutions or visit www.cisco-live.com.