iBGP scalability

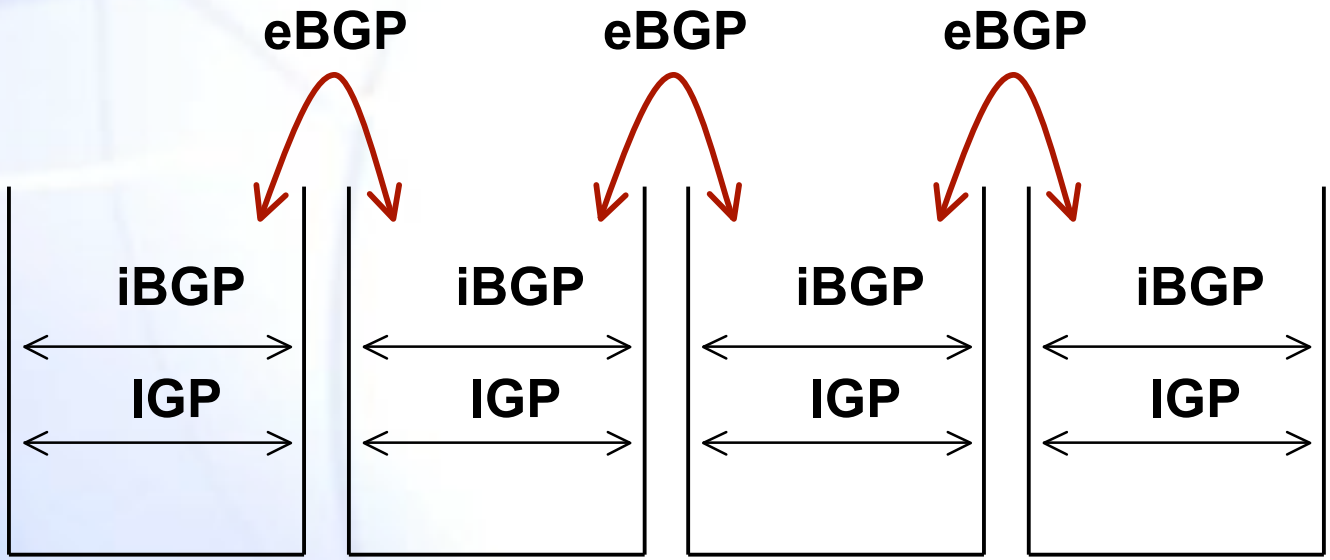
Eduardo Grampín
Universidad Carlos III de Madrid

Outline

- IGP-BGP interaction
- iBGP scaling architectures
  - Route reflectors
  - Confederations
- Known issues

BGP-IGP Interaction

- ASes exchange reachability information using (external) BGP
- Intradomain routing: IGP
- Propagation of BGP information intradomain: (internal) BGP



BGP-IGP interaction alternatives

- **Propagation of BGP Information via the IGP, multicast or other efficient flooding mechanism**
  - Mentioned in rfcs 1772 & 1773, implementation?
    - ✓ BGP Scalable Transport, t.b.d. in next sessions
- **Redistribution/tagged IGP**
  - Specified in rfcs 1403 & 1745 (moved to historical status)
  - Route tagging implemented in cisco & juniper routers
- **Encapsulation**
  - MPLS tunnels among eBGP speakers
- **Pervasive BGP**

Pervasive Internal BGP (iBGP)

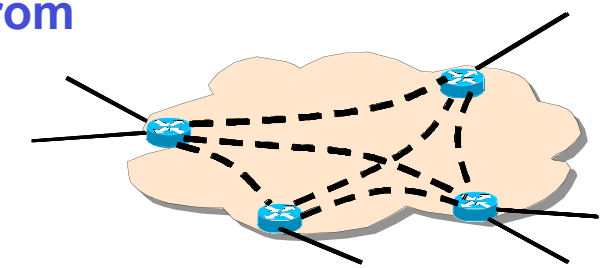
- All routers in an AS are iBGP speakers
- IGP is only used for routing within the AS
  - No BGP routes are imported into the IGP
- Routing table recursive lookup
  - First lookup determine BGP next hop (exit router)
  - Second lookup determine the IGP path to the exit router
- Need to make sure that
  - Internal transport of BGP info is loop-free (just BGP info!)
  - Internal routing is coherent (now, loop-freeness for data plane forwarding)

Internal BGP

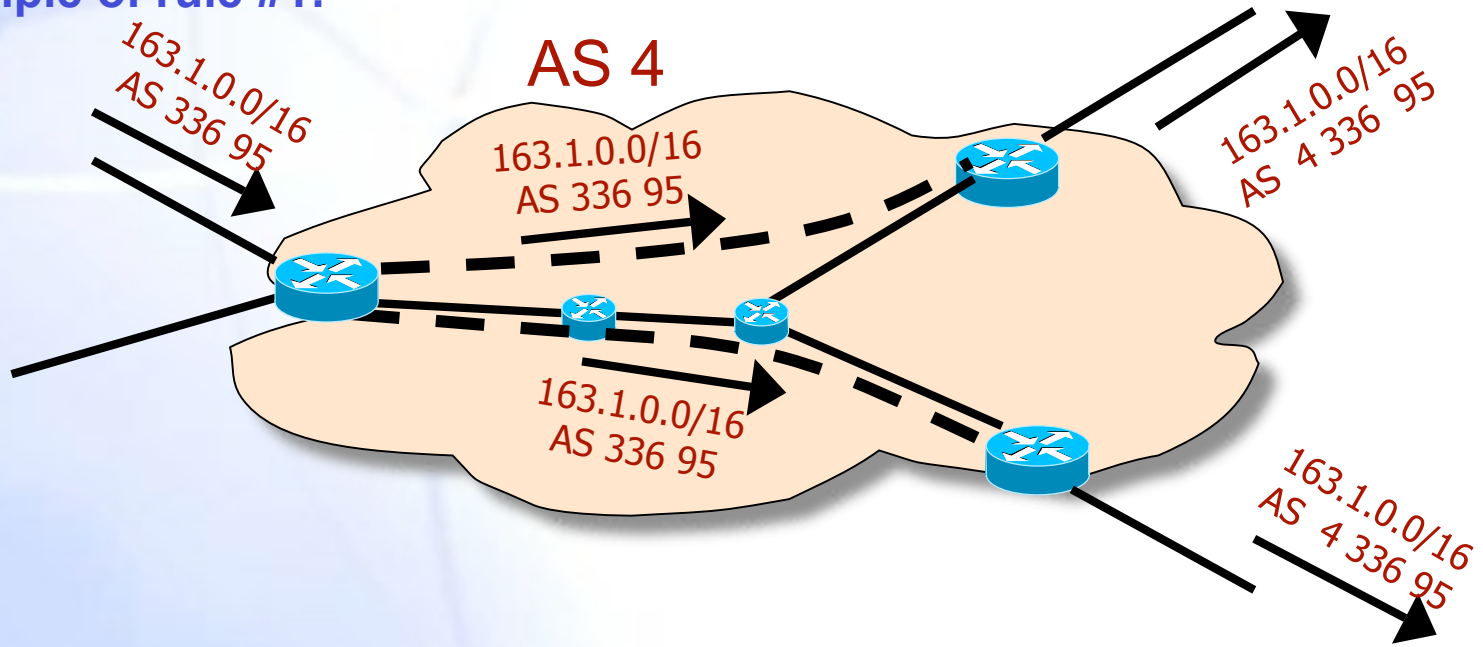
- iBGP and eBGP are same protocol in that
  - same message types used
  - same attributes used
  - same state machine
  - BUT use different rules for readvertising prefixes
- Rules for iBGP
  - #1: prefixes learned from an eBGP neighbor *can* be readvertised to an iBGP neighbor, and vice versa
  - #2: prefixes learned from an iBGP neighbor *cannot* be readvertised to another iBGP neighbor

Loop-freeness of BGP info in iBGP

- Why rule #2? To prevent *BGP announcements* from looping
  - eBGP detect loops via AS-PATH
  - AS-PATH not changed in iBGP
- Implication of rule: a full mesh of iBGP sessions between each pair of routers in an AS is required



- Example of rule #1:

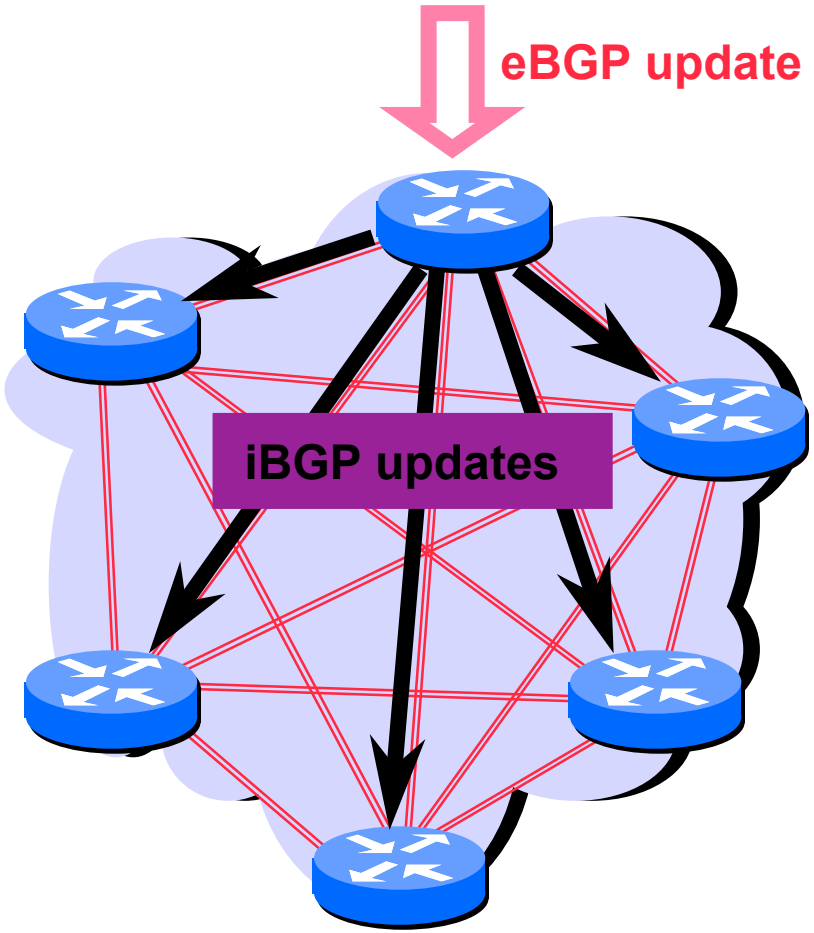


iBGP full-mesh scalability

- n*(n - 1)/2 iBGP sessions
- Configuration management
  - Each router must have n-1 iBGP sessions configured
  - The addition a single iBGP speaker requires configuration changes to all other iBGP speakers
  - E.g. if we have 200 routers in our network that would give us 19900 BGP sessions!

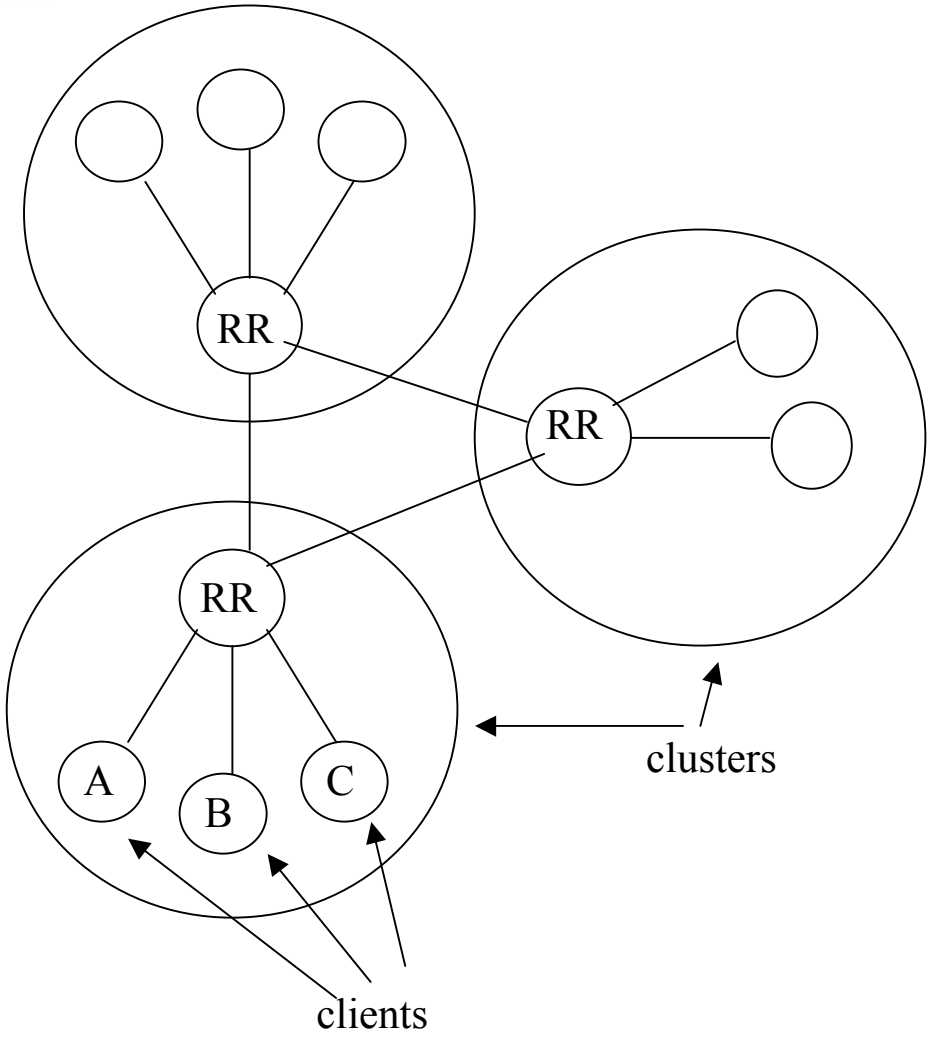
iBGP full-mesh scalability

- **Routing state**
  - Many Adj-RIBs : most routes are not used
  - Size of iBGP routing table can be order n larger than number of best routes (remember alternate routes!)
  - Each router has to listen to update noise from each neighbor
  - CPU and memory resources -> large routers needed!
- **Solutions**
  - Route Reflectors
  - Confederations



Route Reflectors

- Avoiding the virtual full mesh of iBGP sessions:
  - group routers into clusters
  - Assign a leader to each cluster, called a route reflector (RR)
  - Members of a cluster are called clients of the RR
- The clients do not know they are clients and are configured as normal iBGP peers
- Only the best route to a destination is sent from a RR to a client



Route Reflectors: announcements

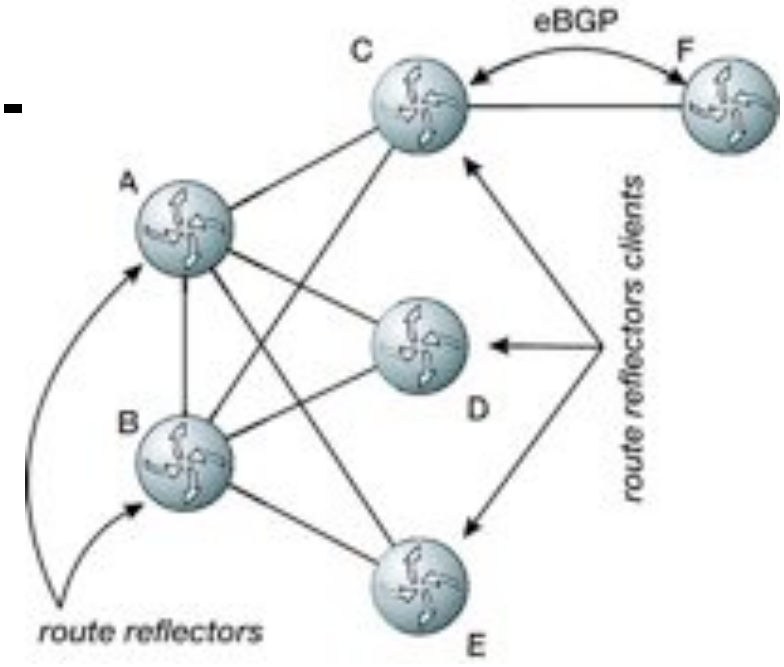
- If received from RR, reflect to clients
- If received from a client, reflect to RRs and clients
- If received from eBGP, reflect to all: RRs and clients
- RRs reflect only the best route to a given prefix, not all announcements they receive
  - helps size of routing table
  - sometimes clients don't need to carry full table
- RR should not change the attributes
  - NEXT_HOP
  - AS_PATH
  - LOCAL_PREF
  - MED

Avoiding Loops with Route Reflectors

- Loops cannot be detected by traditional approach using AS_PATH because AS_PATH not modified within an AS
- Announcements could leave a cluster and re-enter it
- Two new attributes added by RR *if a route is reflected*
  - ORIGINATOR_ID: Router ID of route's originator in AS
    rule: announcement discarded if returns to originator
  - CLUSTER_LIST: a sequence of Cluster Ids, set by RRs
    rule: if an RR receives an update and the cluster list contains its Cluster ID, then update is discarded
- Both are optional, nontransitive (dont propagate to eBGP)

Multiple route reflectors

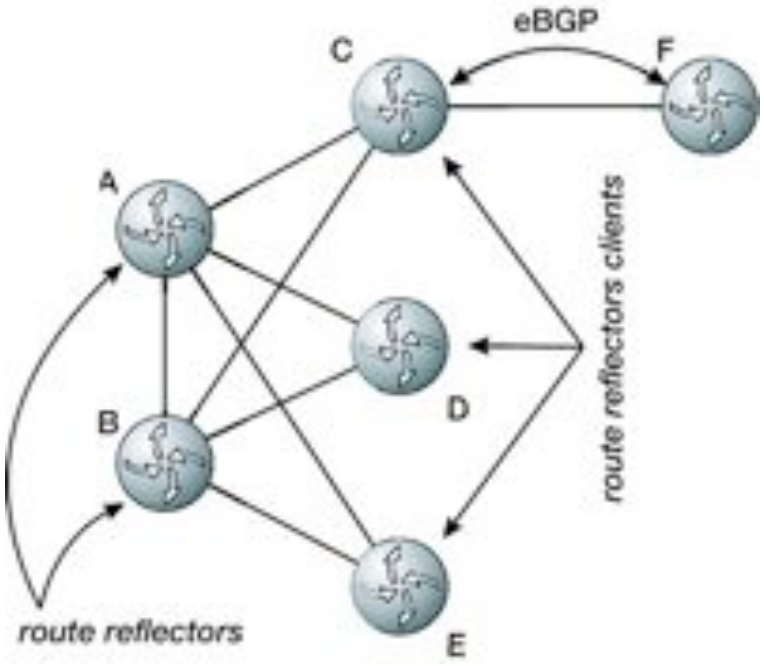
- For redundancy, is possible to have more than one route reflector in a cluster
  - Otherwise, the RR is a 'single-point-of-failure'
- RRs in a cluster may have the same cluster ID
  - …or different cluster IDs
  - *Let's discuss alternatives over this sample topology*



Source: Practical BGP

Multiple route reflectors

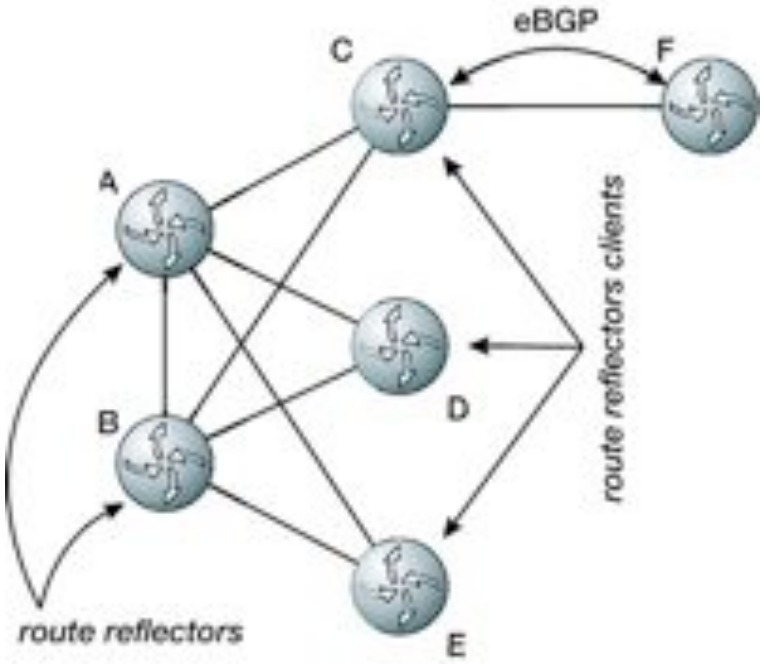
- **Different Cluster IDs:**
  - C receives some route *p* from F. It advertises *p* to A and B
  - A creates a new Cluster List attribute and inserts its router ID, and sets the originator ID to C. A advertises *p* to B, D, and E
  - B receives this update, and discovers that its cluster ID is not in the Cluster List. It accepts the advertisement and prepends its cluster ID to the Cluster List
  - B also receives the update from C, creates a new Cluster List attribute, and inserts its cluster ID. Also sets the originator ID to C. B then runs the best path algorithm and selects the direct path via C and advertises this update to D, E and A
  - Since A and B are using different Cluster IDs, D and E each receive two copies of the update



Source: Practical BGP

Multiple route reflectors

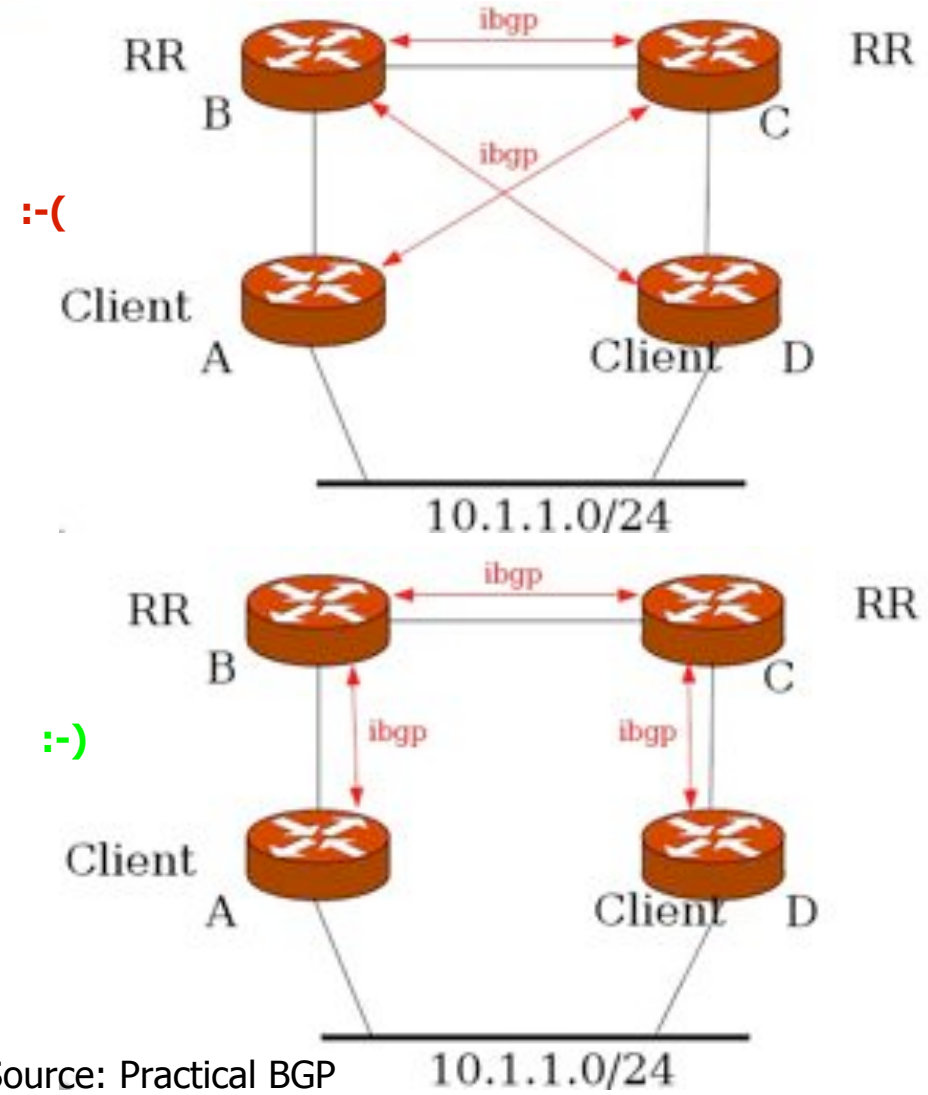
- **Same cluster IDs:**
  - C receives some route *p* from F. It advertises *p* to A and B
  - A creates a new Cluster List attribute, inserts its cluster ID (the same as B), and sets the originator ID to C. Router A then advertises *p* to B, D, and E.
  - B receives this update and discards it, because its locally defined cluster ID is already in the Cluster List
  - B also receives the update from C, adds a Cluster List attribute, inserts its cluster ID, and sets the originator ID to C. B advertises this update to D, E and A
  - When A receives the update from B, it discards it because the cluster list contains the locally configured cluster ID
  - Now routers A and B are only required to store one copy of the route



Source: Practical BGP

Multiple route reflectors

- A and D receive an advertisement for 10.1.1.0/24 from an external BGP peer
- B will prefer D and C will prefer A => routing loop!

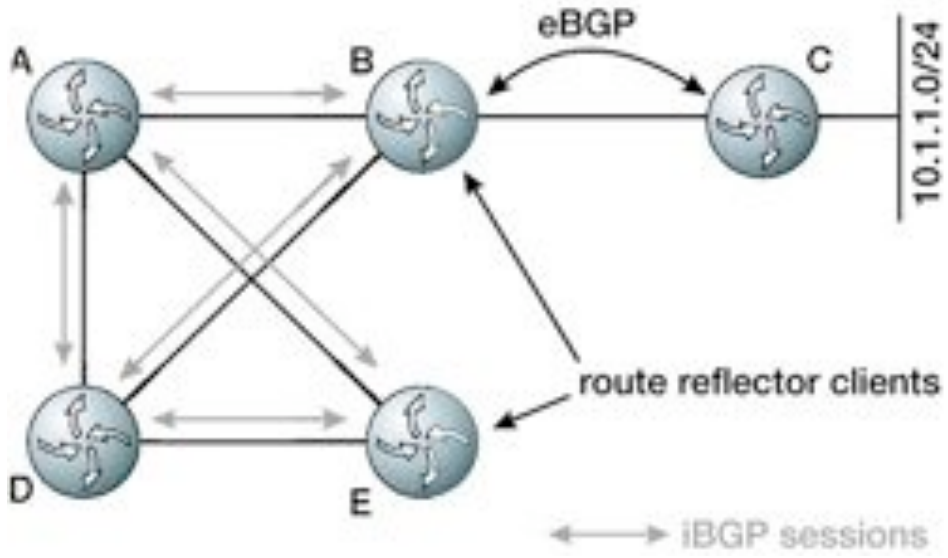


Route reflectors: follow the physical topology

Source: Practical BGP

Multiple route reflectors

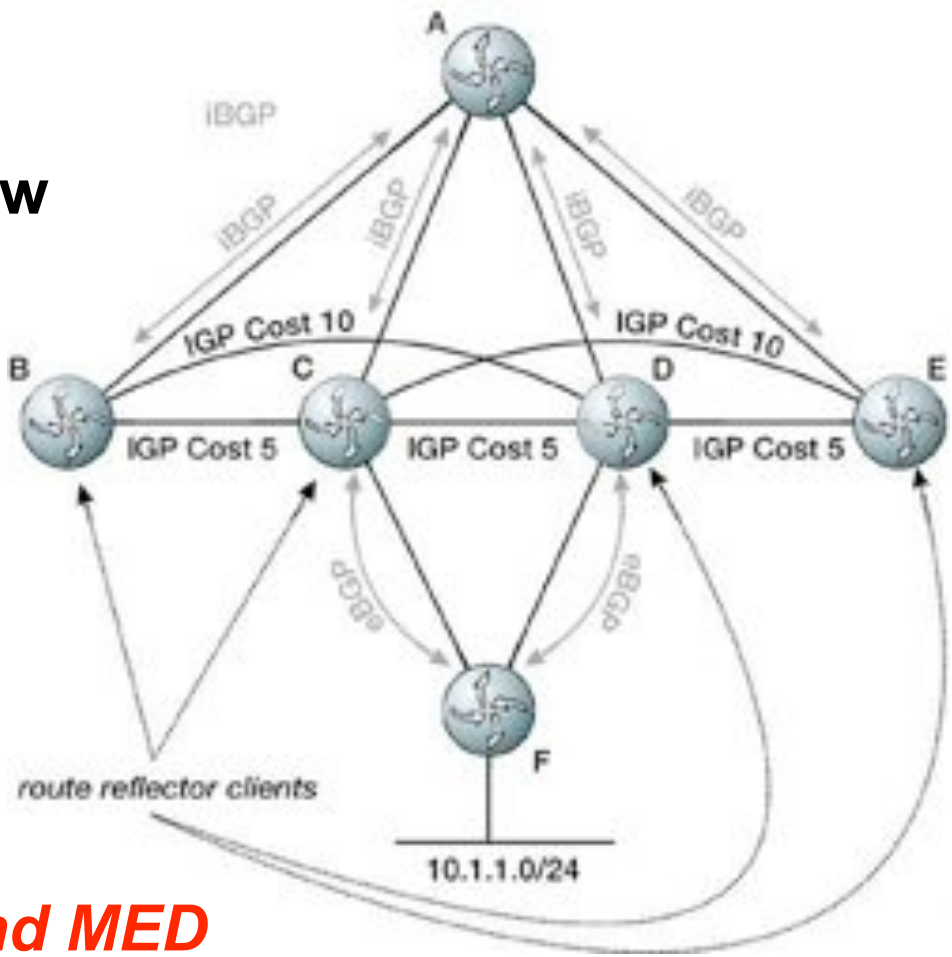
- Routers A and B shall be configured as clients of both A and D
  - To avoid single point of failures
- Alternative: B and E iBGP peers
  - Problem?



Source: Practical BGP

Route reflectors: suboptimal route selection

- First design: full mesh iBGP
  - Next hop selection follow IGP
- Using A as RR
  - Supose A chooses C
  - What happens?

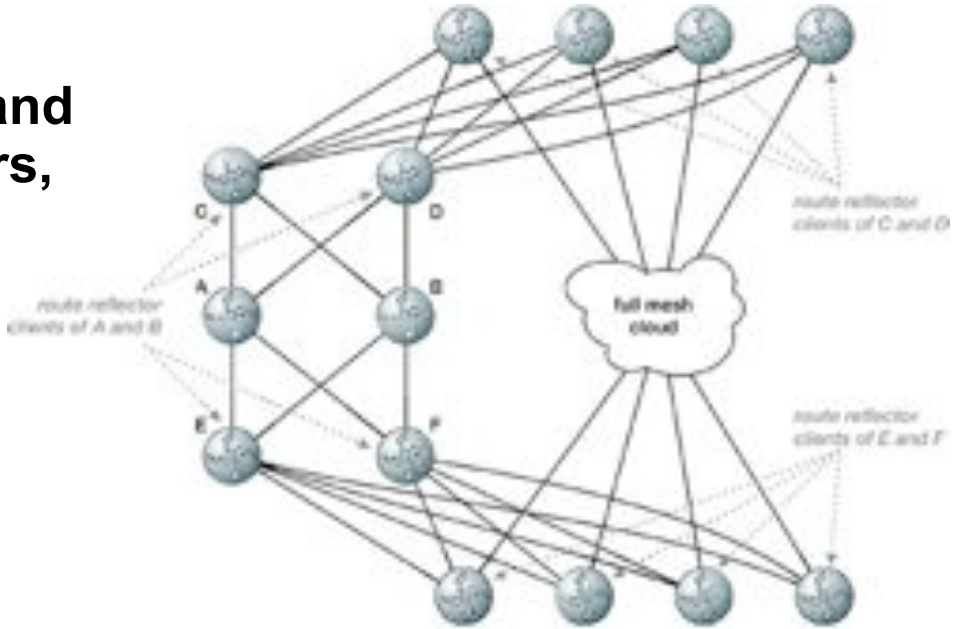


Obs: ignore LOCAL_PREF and MED

Source: Practical BGP

Hierarchical Route Reflectors

- Example with three route reflection clusters
  - Root cluster: A and B are route reflectors, while D, C, E, and F are clients.
  - Two lower-level clusters: C and D, E and F are route reflectors, respectively.
- Remember: RR topology should follow the physical topology of the interconnected iBGP speakers.
  - Avoiding forwarding loops



Source: Practical BGP

Confederations

- Another way of solving iBGP full mesh
- The idea behind confederations is to take one large AS and divide it into several smaller ones
- Non-members of the confederation see one AS, members of the confederation are divided into sub-ASes
- One IGP must usually be run in the whole confederation to support connectivity
- LOCAL_PREF and NEXT_HOP is preserved through the confederation

Confederations

From the outside, this looks like AS 1

Confederations: mechanism

- Need to prevent loops within the confederation
- Two new segments of the AS_PATH are added (apart from AS_SEQUENCE and AS_SET):
  - AS_CONFED_SET
  - AS_CONFED_SEQUENCE
- BGP speakers add sub-AS numbers to these within the confederation
- These are stripped when announced over eBGP

Confederations: Sub AS numbers

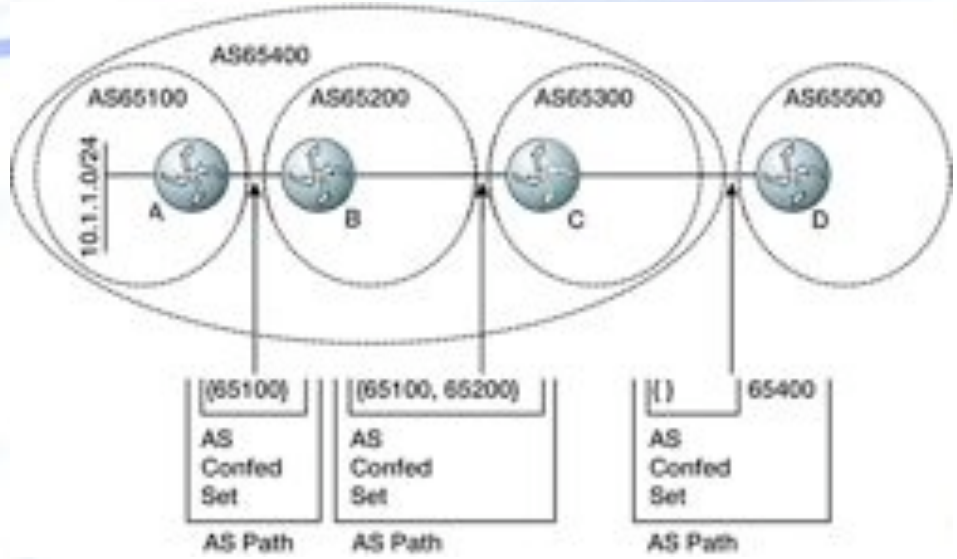
- AS confederation identifier = the external AS number
- AS member number = the confederation sub-AS number
- Design considerations: when configuring confederations use private AS numbers (64512 – 65535)
  - Some implementations of confederations have been known to leak the member sub-AS numbers to its eBGP peers
  - What happens if you use public AS numbers that belonged to someone else?

Confederations: announcing rules

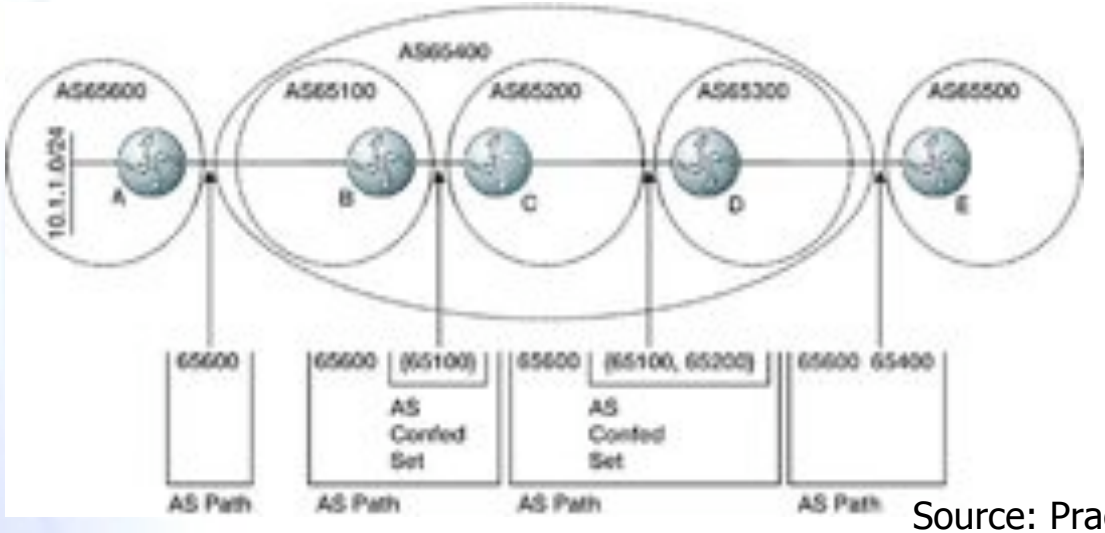
- IBGP (within a sub-AS) behaves as normal
- BGP peering between sub-ASs (sometimes called eiBGP):
  - Prepend the sub-AS (AS member #) to the AS_PATH using the AS_CONFED_SEQ
- When a BGP update is leaving the confederation
  - Remove the prepended sub-AS information from the AS_PATH
- Differences between eBGP and eiBGP
  - LOCAL_PREF is preserved through the confederation
  - NEXT_HOP is also preserved
- Speak eiBGP or eBGP to your neighbor?
  - Share AS confederation identifier? -> eiBGP

Confederation: examples

Tracking an update originated in a confederation



A route propagated through a confederation



Source: Practical BGP

Confederations: sub-hierarchies

- Is not possible to build sub-hierarchies using confederations
- Is possible to use route reflection within a sub-AS
  - and even sub-route reflector hierarchies....

Deploying Confederations

- **Why?**
  - **You might wish to use BGP confederations to scope BGP policy and administrative controls based on geographic or political boundaries**
  - **You're planning to run an independent interior gateway protocol within each sub-AS and are bounded by the existing IGP topologies or scalability thresholds**
    - ✓ **Alternative to IGP areas**
  - **Transoceanic links and other physical topology characteristics of the network dictate constraints**
  - **You've acquired other networks and are going to employ BGP confederations as a first phase of full integration**

MED oscillation

- RFC 3345
  - Bad news:
    - ✓ "In certain topologies involving either route reflectors or confederations, the partial visibility of the available exit points into a neighboring AS may result in an inconsistent best path selection decision as the routers don't have all the relevant information. If the inconsistencies span more than one peering router, they may result in a persistent route oscillation"
  - (Relative) good news
    - ✓ "The persistent route oscillation behavior is deterministic and can be avoided by employing some rudimentary BGP network design principles until protocol enhancements resolve the problem"

MED oscillation

- RR/confederation hides some information
  - RR/confederation sends best path only
  - not all routers know all best paths
- MED (Multi Exit Discriminator) vs IGP cost to the neighbor…
- Seen on a RR/confederation border:

```
#show ip bgp 10.0.0.0 | include best #
    Paths: (3 available, best #3)
#show ip bgp 10.0.0.0 | include best #
    Paths: (3 available, best #2)
#show ip bgp 10.0.0.0 | include best #
    Paths: (3 available, best #3)
                    ...
```

MED oscillation

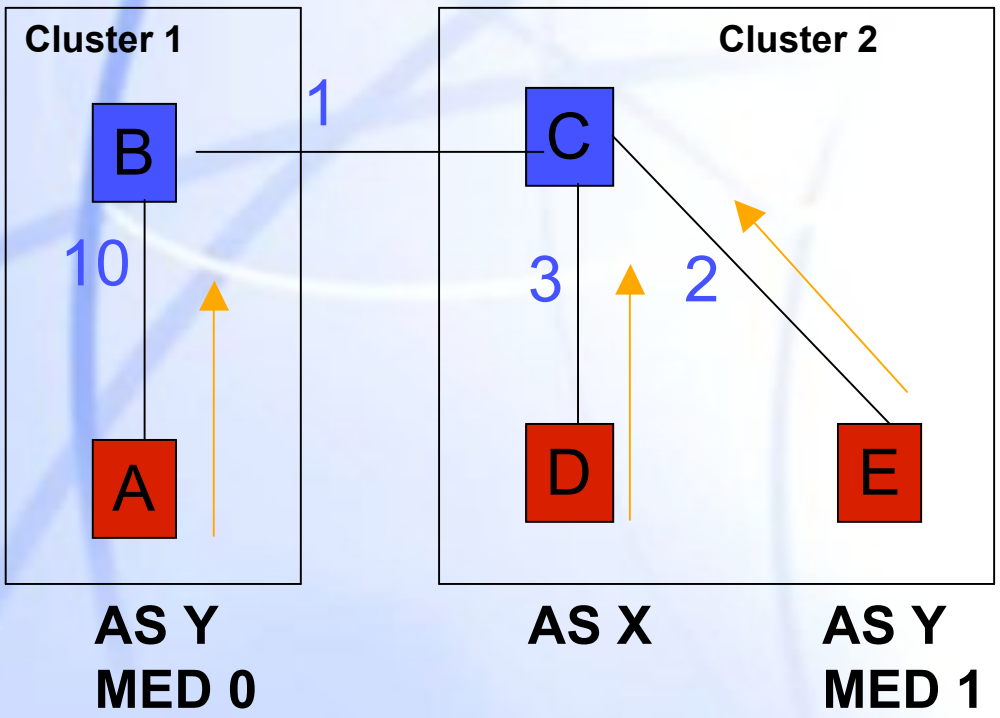
= Route Reflector

= Client

= Advertisement

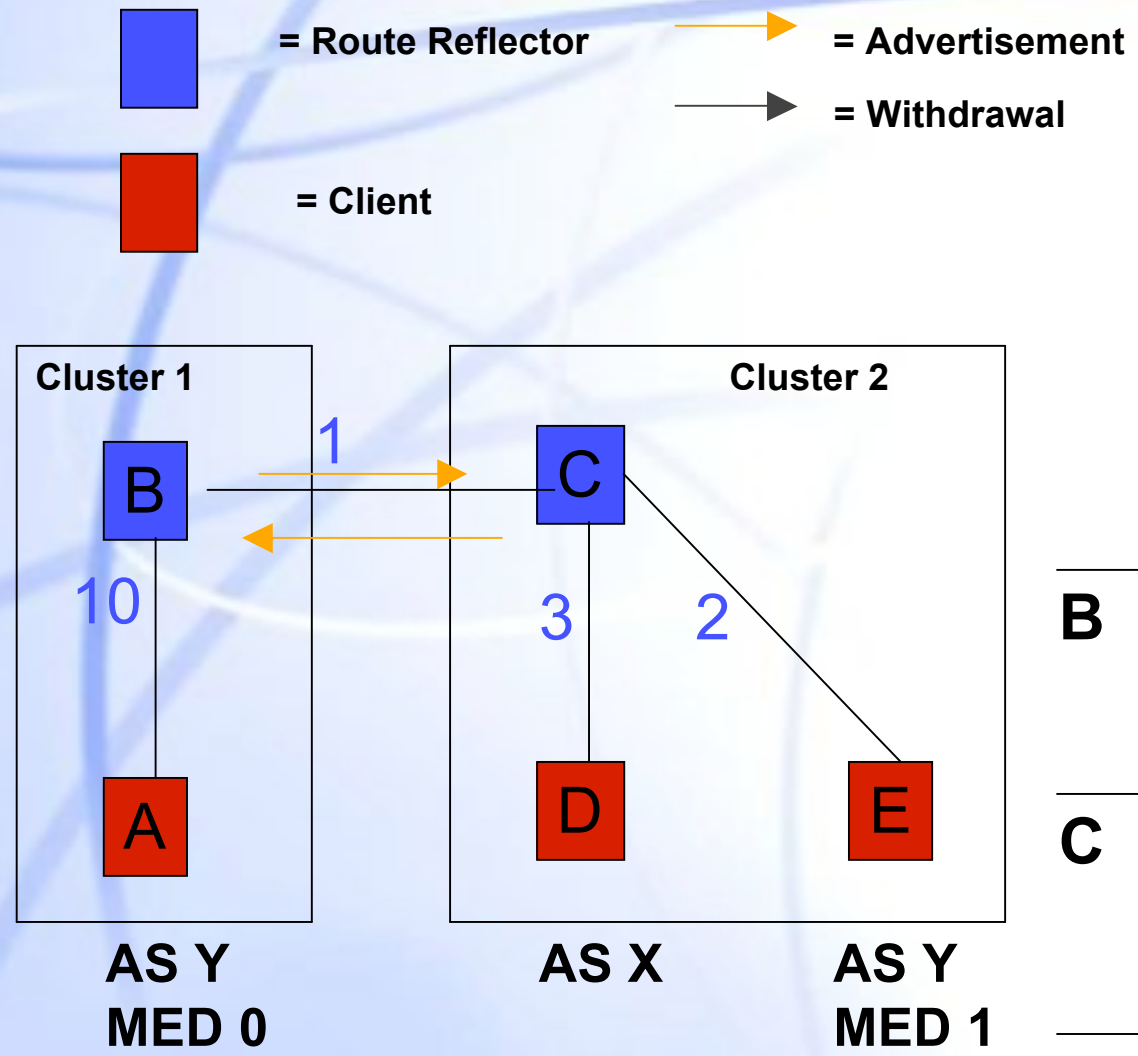
= Withdrawal

Step 1
- B selects Y0
- C selects Y1



	AS_PATH	MED	IGP
B	* Y	0	10
C	X		3
	* Y	1	2

MED oscillation



Step 2
- **C selects X**

	AS_PATH	MED	IGP
B	Y	1	3
	* Y	0	10
C	* X		3
	Y	1	2
	Y	0	11

MED oscillation

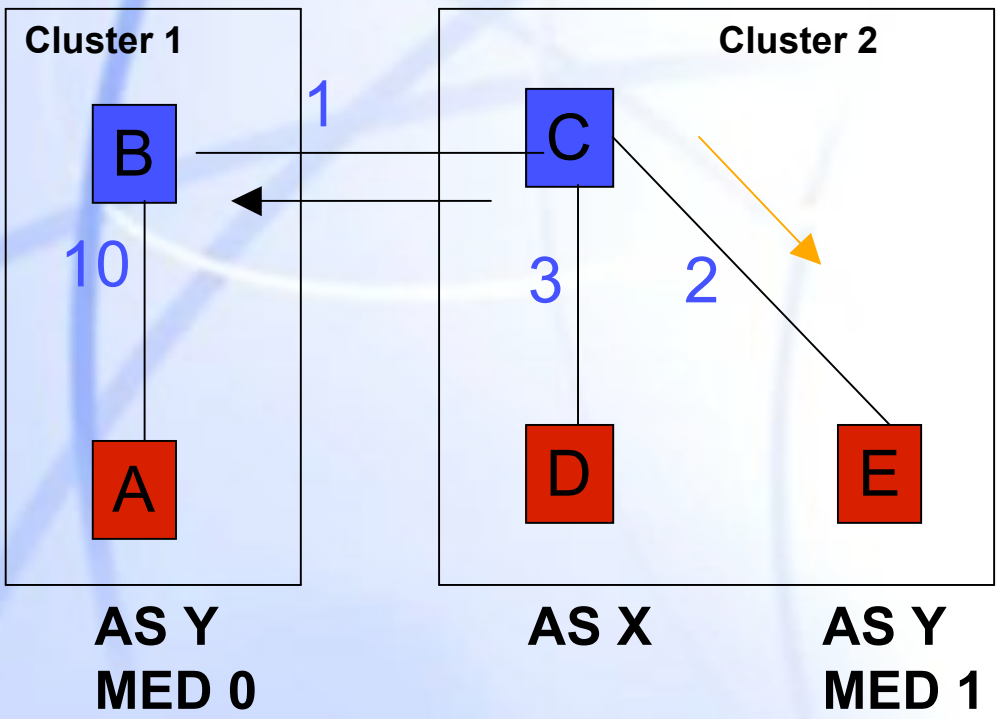
= Route Reflector

= Client

= Advertisement

= Withdrawal

Step 3
- **B selects X**



	AS_PATH	MED	IGP
B	* X		4
	Y	0	10
C	* X		3
	Y	1	2
	Y	0	11

MED oscillation

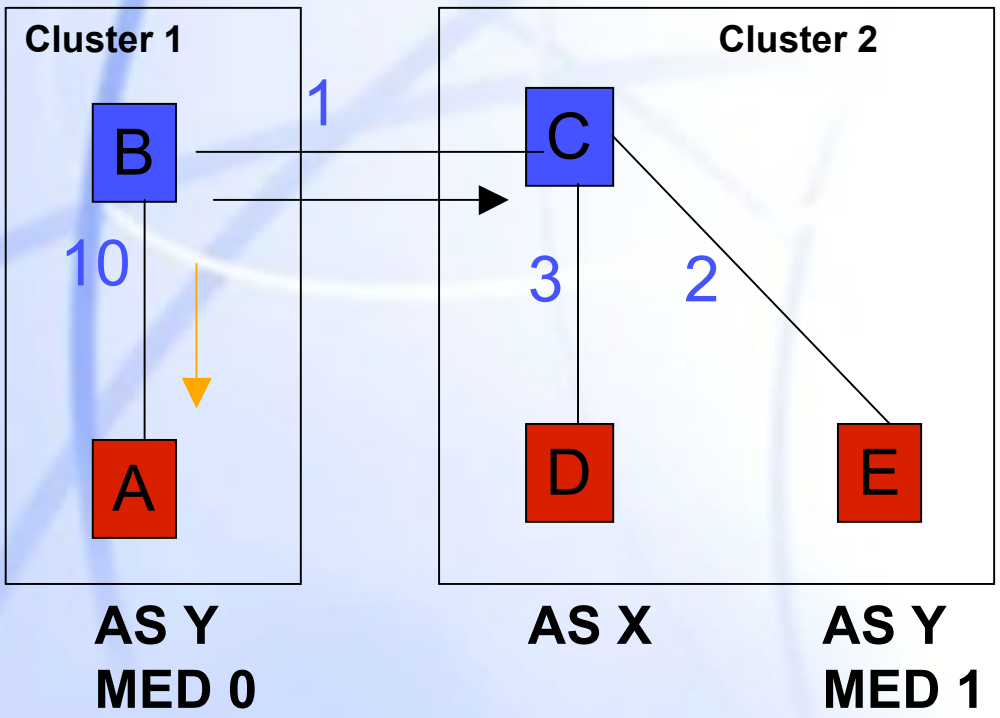
= Route Reflector

= Client

= Advertisement

= Withdrawal

Step 4
- **C selects Y1**



	AS_PATH	MED	IGP
B	* X		4
	Y	0	10
C	X		3
	* Y	1	2

MED oscillation

= Route Reflector

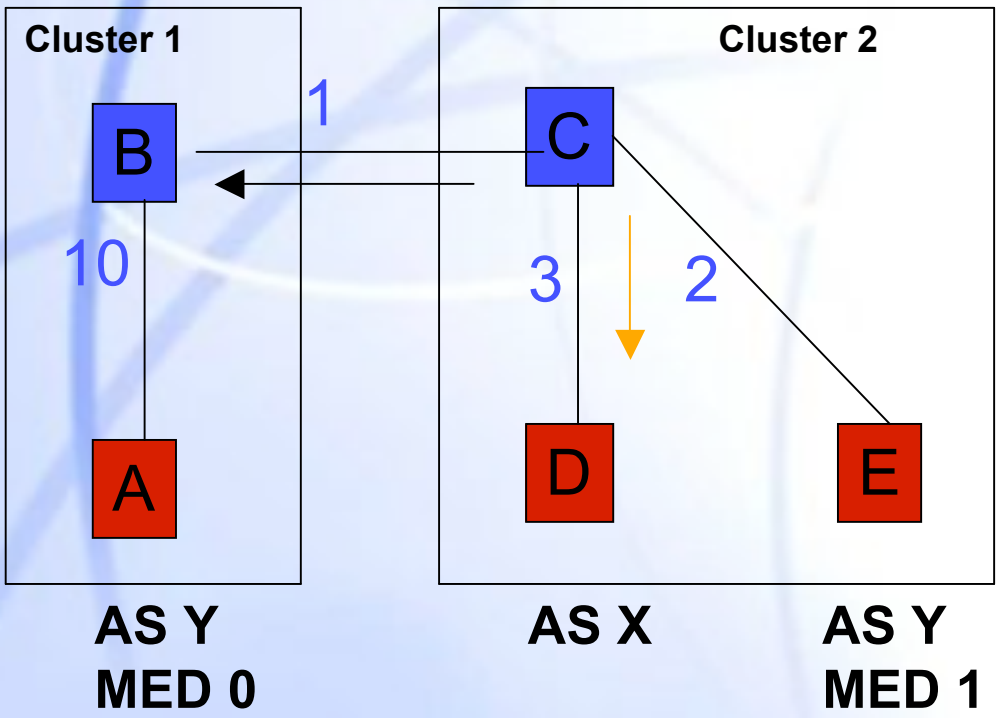
= Client

= Advertisement

= Withdrawal

Step 5
– **B selects Y0**

= Step 1!!



	AS_PATH	MED	IGP
B	* Y	0	10
C	X		3
	* Y	1	2

MED oscillation: workarounds

- Use full mesh iBGP
  - Scalability…
- Do not listen to the MED (or only with stub-AS)
  - `set metric 0` on all prefixes
- `bgp always-compare-med`
  - Inconsistent! -> remember MEDs is per-AS attribute
- Use local-pref to force decision
  - exit no longer chosen by peer = more work :(
- Allow peer to set local-pref using community

- *Protocol improvement?*

iBGP Scalability: summary

- iBGP is necessary for core routers in a transit network
- BGP loop detection mechanism is based on AS_PATH-> IBGP peering must be fully meshed
- This leads to scaling problems
- Solutions:
  - Route reflectors
  - AS confederations

iBGP Scalability: summary

- Known issues
  - Suboptimal intra-domain routing
  - Divergence and/or non-deterministic update process
    - ✓ MED oscilation
  - Route deflection
    - ✓ Re-route at exit point!

References

1. *Practical BGP*, by Russ White, Danny McPherson, Sangli Srihari
2. *IBGP scaling: Route reflectors and confederations*, Olof Hagsand KTH /CSC
3. *BGP Oscillation*, Fabien Berger
4. RFC 1772, "Application of the Border Gateway Protocol in the Internet", March 1995.
5. T. Bates, E. Chen, R. Chandra, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*. IETF RFC 4456 (Obsoletes: 2796, 1966), April 2006.
6. P. Traina, D. McPherson, J. Scudder, *Autonomous System Confederations for BGP*. IETF RRFC 5065 (Obsoletes: 3065), August 2007.
7. D. McPherson, V. Gill, D. Walton, A. Retana, *Border Gateway Protocol (BGP) Persistent Route Oscillation Condition*. IETF RFC 3345, August 2002.