IBGP scaling: Route reflectors and confederations

Olof Hagsand KTH CSC

Literature

- Route Reflectors
  - Practical BGP pages 135 – 153
  - RFC 4456
- Confederations
  - Practical BGP pages 153 – 160
  - RFC 5065

Motivation

- Scalability problems with iBGP full mesh
- n*(n-1)/2 where n = the number of iBGP speaking routers

200 routers in a network results in 19900 iBGP sessions!

This leads to waste of resources:

- Many Adj-RIBs : most routes are not used
- High memory consumption
- Many TCP connections
- High bandwidth usage – same information sent over many TCP connections on same links
- High CPU usage

Solutions

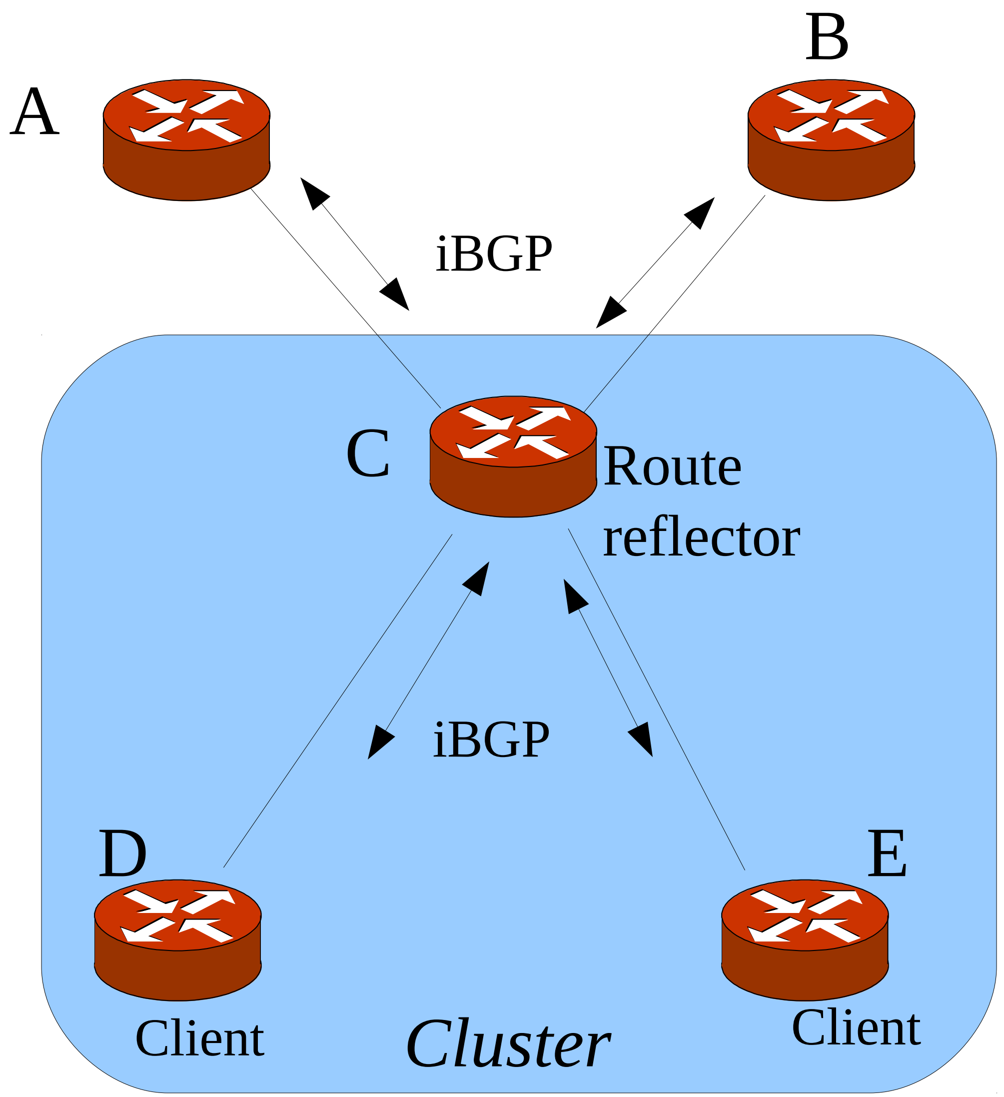
- Introduce a hierarchy of routers in an AS: *Route reflectors*
- Partition the AS into sub-AS:s : *Confederations*
- BGP-free core - Do not run BGP in core (internal) routers: MPLS

Route reflectors, clients and clusters

- Route reflection is concerned with *distributing* routes within an AS, not the actual routing.
- The route reflectors (RRs) have to be configured to reflect routes to router reflector clients.
- The clients do not know they are clients and are configured as normal iBGP peers.
- A set of RRs and clients is referred to as a *cluster*.
- Only the best route to a destination is sent from a RR to a client
  - A reflector makes the route decision for its clients
- To avoid loops, A RR setup should always follow the physical topology

See practical BGP p 135-153

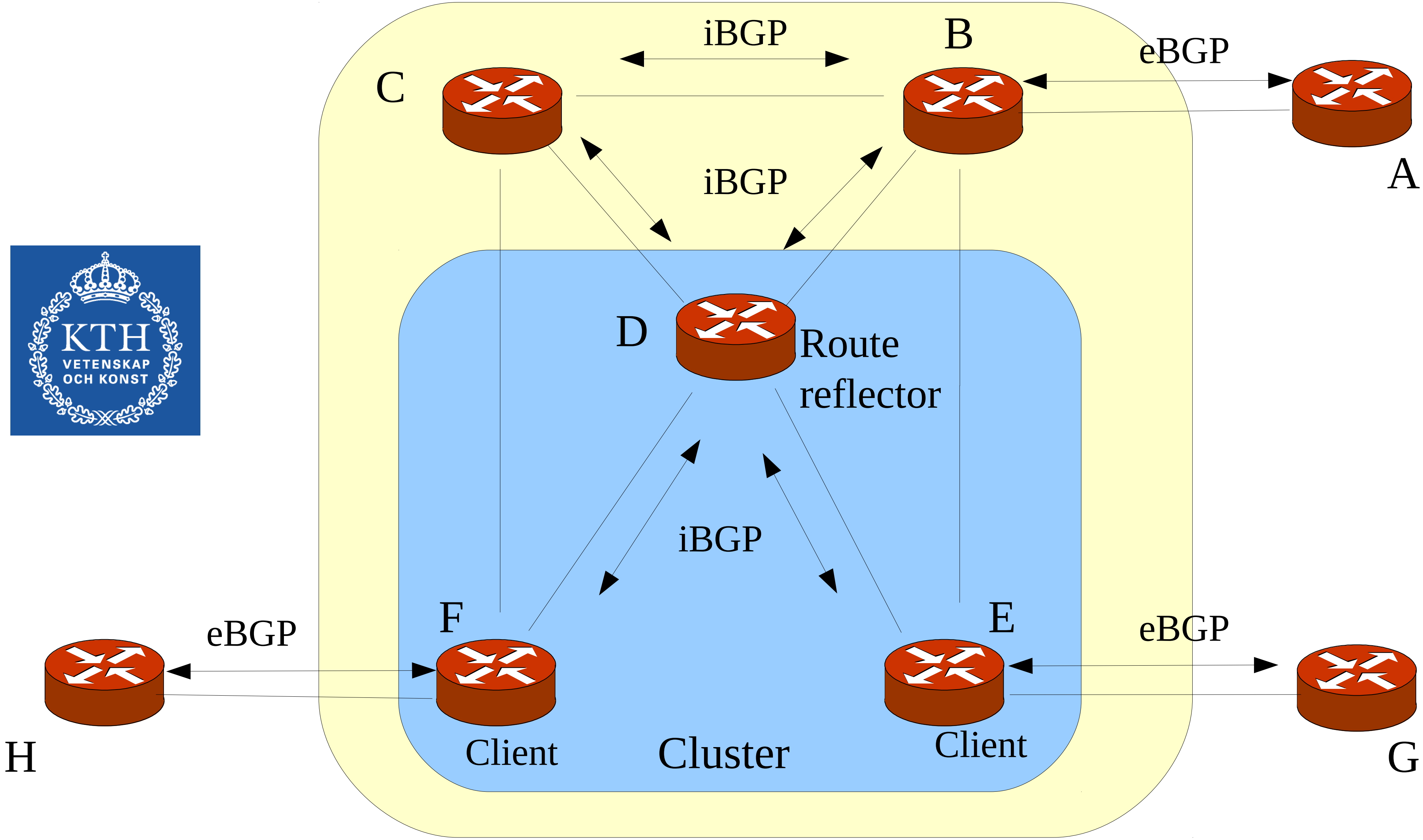
A cluster with one RR and two clients



Route reflectors

- A route reflector reflects iBGP routing information:
- From clients to iBGP peers and other clients
- From iBGP peers to clients
- Never from iBGP peers to iBGP peers (as before)
- Should not change the attributes
    - NEXT_HOP
    - AS_PATH
    - LOCAL_PREF
    - MED

Route reflector example



See practical BGP p 136

Exercise

- Using the previous figure:

1) Trace a route from A, G and H respectively
2) Suppose the same route comes from both G and H, how does it propagate throughout the AS?
3) How does traffic go in the AS. For example transit traffic between A and G?

Path attributes

- Two new attributes added by RR *if a route is reflected*
  - CLUSTER_LIST
    - RR adds a clusterid in the cluster list (like a path)
  - ORIGINATOR_ID
    - First RR add routerid of the peer it heard it from
  - Both are optional, nontransitive (dont propagate to EBGP)
- Cluster ID
  - 32 bit dotted decimal notation in JunOS
  - Does not have to be a routed address
  - Usually the RRs routerid is used, but can be configured to something else (see clusterid with multiple RRs)

Route reflector configuration in JunOS

```
protocols {
    bgp {
        group INTERNAL-RR {
            type internal;
            local-address 192.168.1.1;
            cluster 192.168.1.1;
            neighbor 192.168.1.2;
        }
    }
}
```

Route reflector client configuration

```
protocols {
    bgp {
        group INTERNAL {
            type internal;
            local-address 192.168.1.2;
            neighbor 192.168.1.1;
        }
    }
}
```

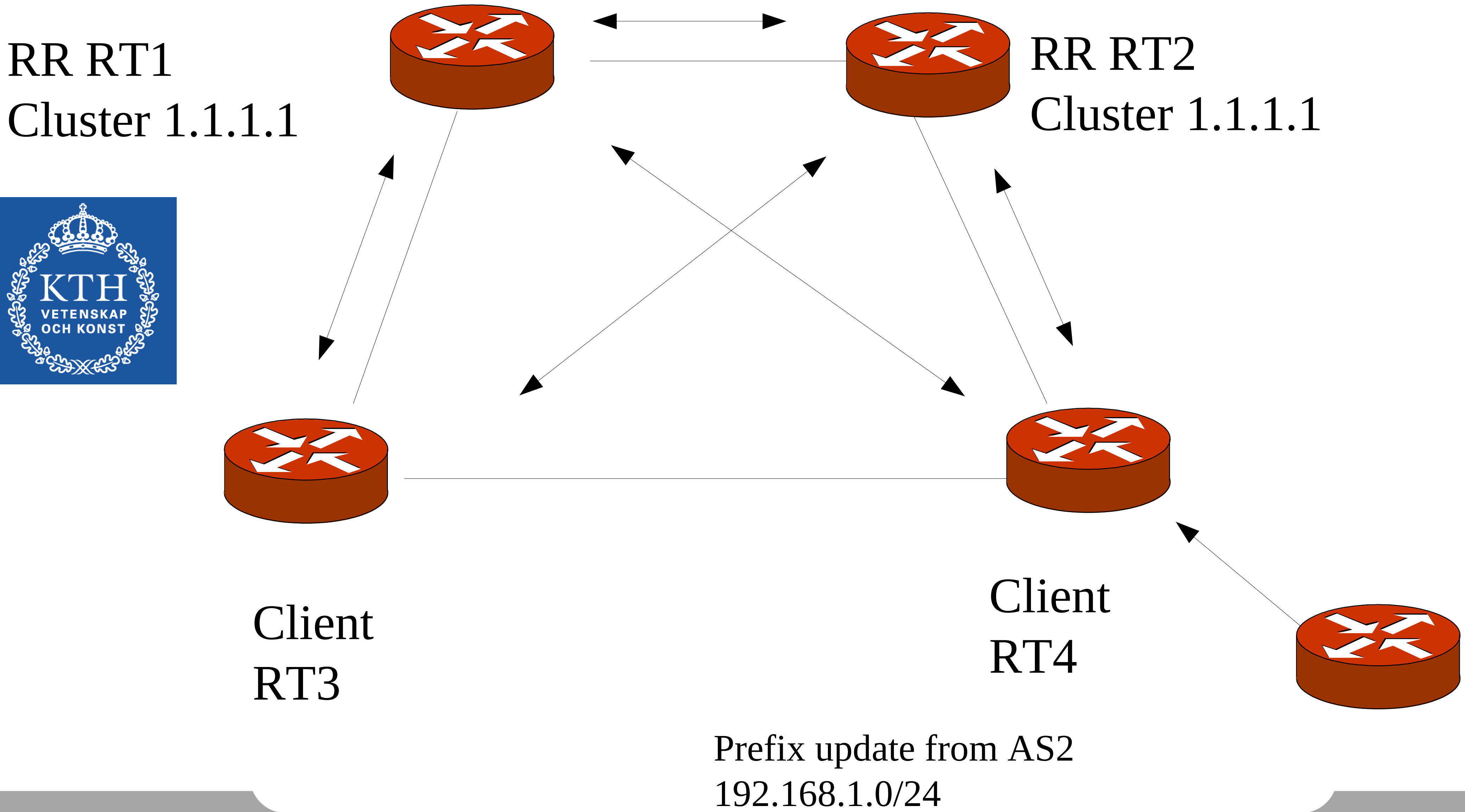
The client configuration is not 'aware' of route reflection

Multiple route reflectors

- For redundancy, you can have more than one route reflector in a cluster.
- Otherwise, the RR is a 'single-point-of-failure'
- You can choose to have the same cluster ID on the RRs in a cluster, or different cluster IDs

See practical BGP p 143-144

Cluster with multiple RRs



Multiple RRs (1)

- In the example both RRs add the same Cluster ID
- This will result in
  - RT4 gets a prefix on an eBGP peer and sends the update to its iBGP peers (RT1 and RT2)
  - RT1 and RT2 adds Cluster ID 1.1.1.1 to the CLUSTER_LIST and adds RT4's Router ID to ORIGINATOR_ID.
  - RT1 and RT2 reflects the update to all iBGP peers and RR clients (in JunOS RT4 will not get the update back)
  - RT1 receives an update from RT2 with the same information in CLUSTER_LIST and ORIGINATOR_ID as it already had and therefore drops it

Multiple RRs (2)

- RT3 receives updates from both RT1 and RT2 with the same information in CLUSTER_LIST and ORIGINATOR_ID. RT3 will install one of the updates and drop the other.
- What will happen if a router RTx (who use RT2 to get to the prefix 192.168.1.0/24) send packets to the destination in AS2 and the iBGP peer between RT2 and RT4 was down?

Multiple RRs: different cluster IDs

- If instead RT1 added the Cluster ID 1.1.1.1 and RT2 the Cluster ID 2.2.2.2
  - RT2 would still have a valid information on where to forward the packets
  - But we would have duplicated paths
  - We would be using additional memory and processor overhead due to the duplicated paths.

Nested RRs

- To further scale a network using RRs.
  - You can use nested RRs
    - An RR client can be an RR for another cluster
  - The Cluster ID must at least be unique per cluster within the AS
  - A RR could be RR for more than one cluster
- Design considerations
  - Always follow the physical topology

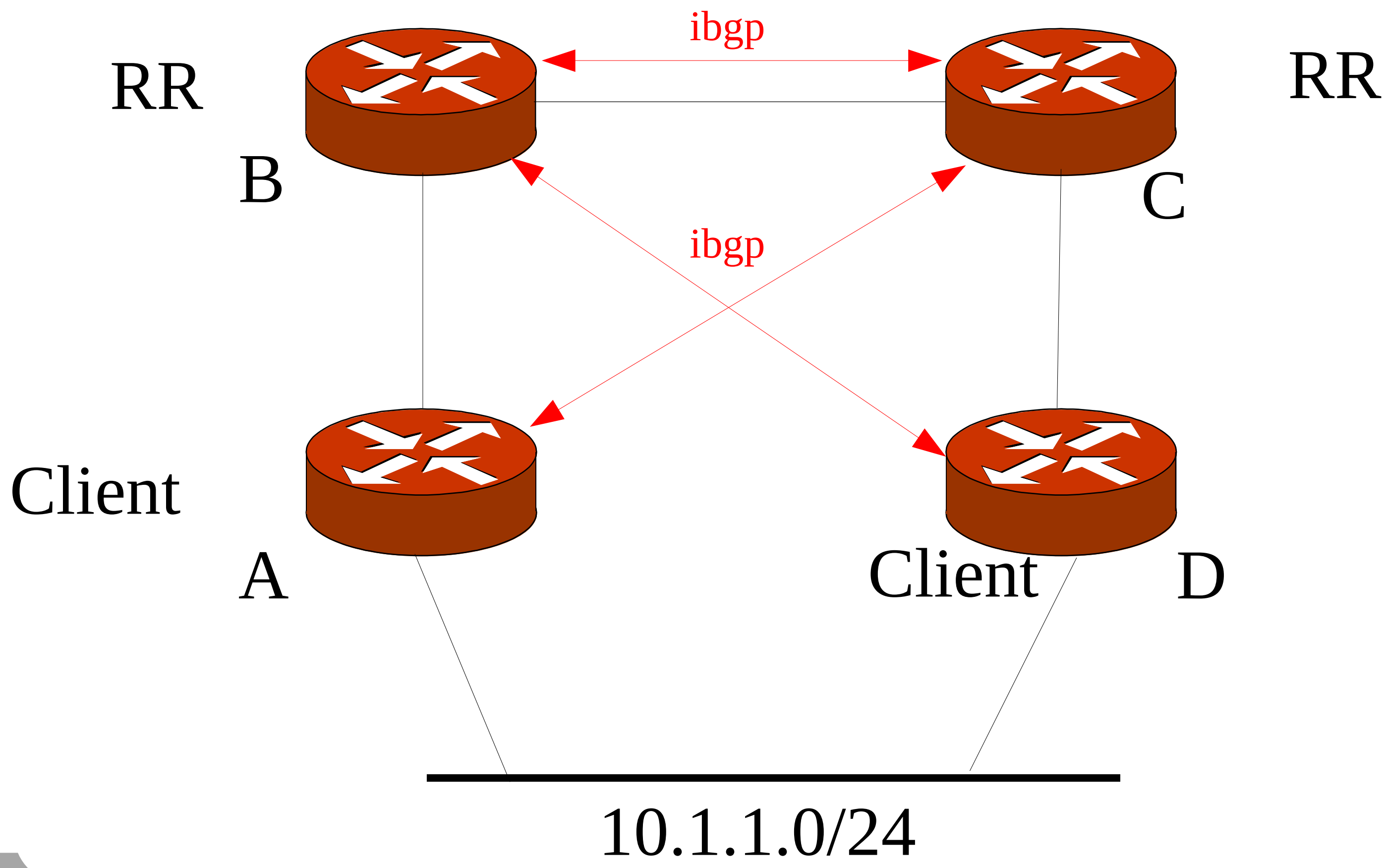
Route reflectors

- There is a possibility to have all the clients within a cluster to have full mesh iBGP
  - If they have a full mesh there is no need to reflect client updates from clients for the RR
  - "set protocols bgp group *group-name* no-client-reflect"

Why follow physical topology?

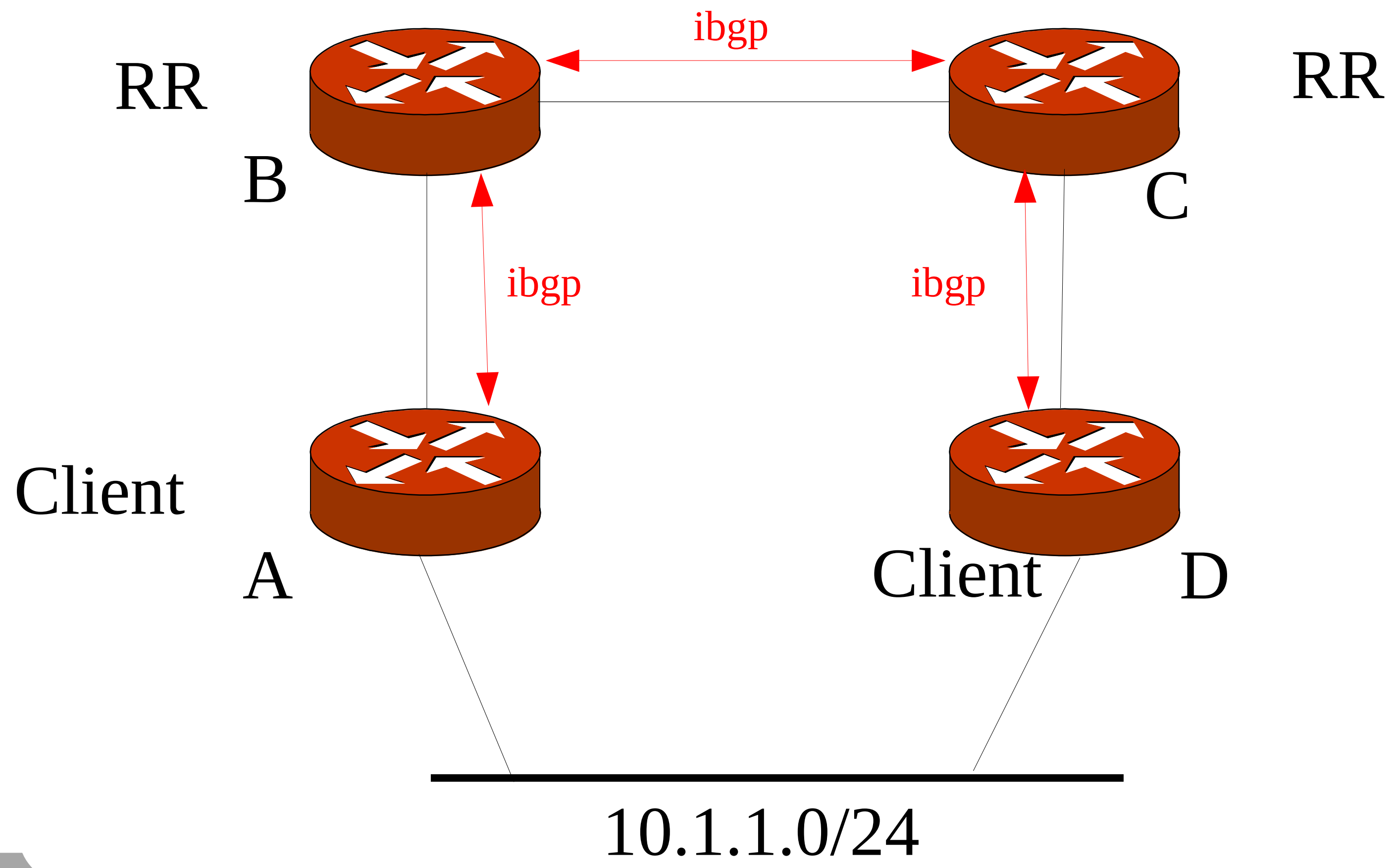
- If you dont, you may have routing loops.
- B will prefer D and C will prefer A => routing loop!

  (The figure replaces practical BGP Fig 4.12)



Follow physical topology?

- This is how the clusters should be defined: the bgp peerings follow the physical links

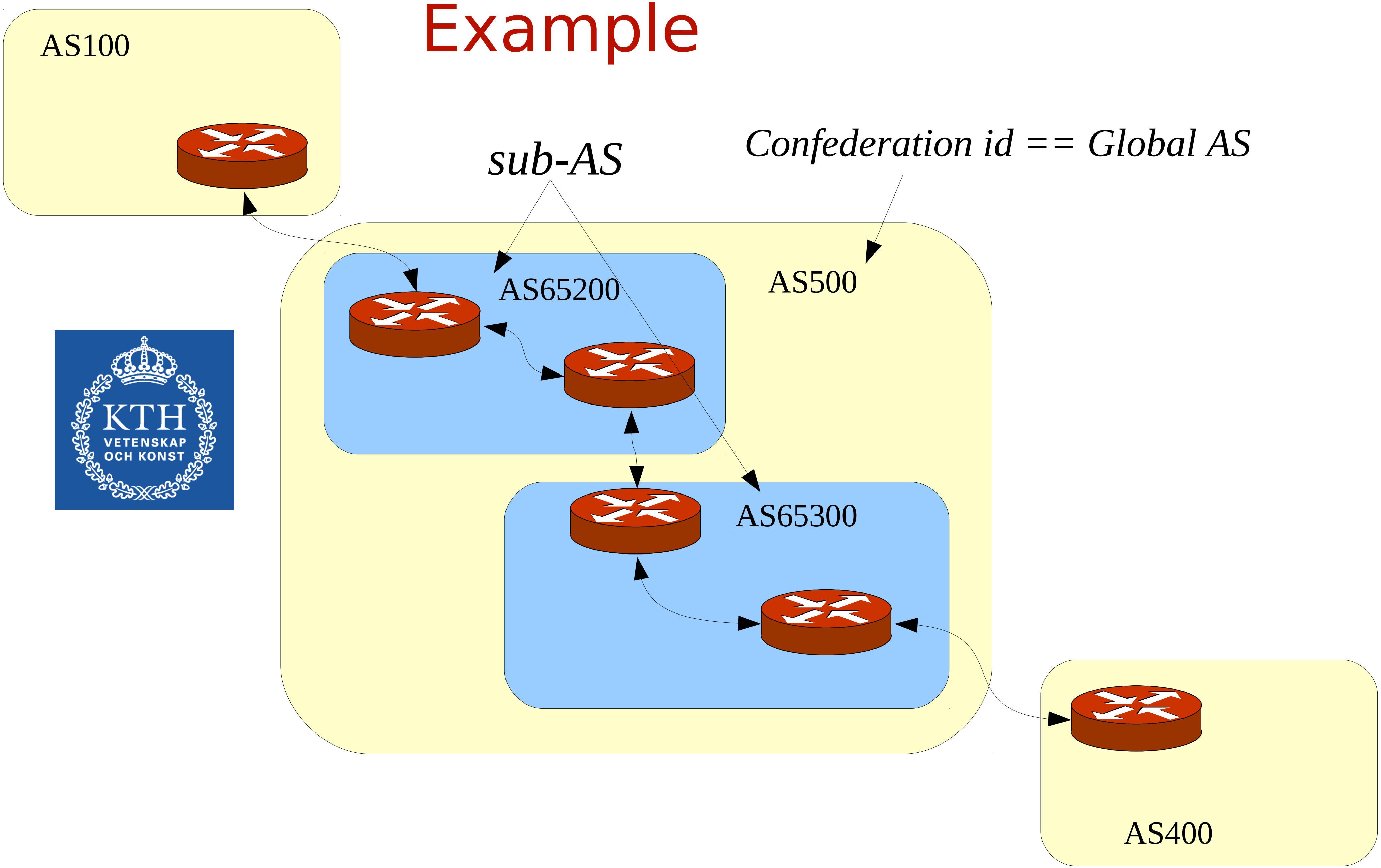


Confederations

- 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-AS's
  - One IGP must usually be run in the whole confederation to support connectivity
  - LOCAL-PREF and NEXTHOP is preserved through the confederation

See practical BGP p 153-160

Example



See practical BGP p 154

Confederations configuration in JunOS

```
routing-options {
    autonomous-system 65200;
    confederation 500 members [65200 65300];
}
protocols {
    bgp {
        group external_ebgp {   # ebgp peering
            type external;
             peer-as 100;
        }
         group internal {         # ibgp peering
            type internal;
        }
         group external_eibgp {  # eibgp peering
            type external;
             peer-as 65300;
        }
    }
}
```

Details omitted and example is a mix of the previous figure

Mechanism

- You need to prevent loops within the confederation
- Two new *segments* of the AS_PATH are added (apart from AS_SEQU
and AS_SET):
    - AS-CONFED-SET
    - AS-CONFED-SEQUENCE
- BGP speakers add sub-AS numbers to these within the confederati
- These are stripped when announced over eBGP

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 it's eBGP peers
    - What happens if you use public AS numbers that belonged to someone else?

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
- You have to know if you should speak eiBGP or eBGP to your neighbor:
  - Share AS confederation identifier -> eiBGP

Sub-hierarchies

- You cannot make sub-hierarchies using confederations
- You can use route reflection within a sub_AS
    - And even sub-route reflector hierarchies,...

IBGP scaling: summary

- IBGP is necessary for core routers in a transit network
- BGP loop detection mechanism is based on ASPATH->
  - IBGP peering must be fully meshed
- This leads to scaling problems
- Solutions:
  - Route reflectors
  - AS confederations
  - BGP-free core
- BGP free core
  - Dont use BGP in the core routers, use some other mechsnism to relay transit traffic
  - MPLS for example