RIPE NCC
RIPE NETWORK COORDINATION CENTRE

BGP Operations and Security

Training Course

Training Services | RIPE NCC | January 2017

Schedule

09:00 - 09:30	**Coffee, Tea**
11:00 - 11:15	**Break**
13:00 - 14:00	**Lunch**
15:30 - 15:45	**Break**
17:30	**End**

Introductions

- Name
- Number on the list
- Experience
  - Routing
  - BGP
- Does your organisation have an AS number ?
- Do you have RIPE NCC Access account ?
- Goals

Overview

- Day 1
  - Introduction to BGP
  - BGP Operations
  - BGP Attributes
  - Traffic Engineering
  - BGP Scalability
  - Multiprotocol BGP
- Day 2
  - Routing Security
  - Filtering
  - IRR
  - RPKI and BGPSEC
  - BGP Software
  - Tips & Tricks

Introduction to BGP

Section 1

The Internet

- Who runs the Internet?
  - No one (in particular), not ICANN, nor the RIRs, nor the EU
- How does it keep working?
  - Internet by and large functions for the common good
  - Business relationships and the need for reachability
- Any help to keep it working?
  - No central coordination
  - Many individuals and organisations

IGP vs EGP

- IGP (OSPF or ISIS)
  - Reachability and path info WITHIN a network domain
  - Provides a Next Hop address and an egress interface to any known destination address
- EGP
  - Reachability and path info BETWEEN network domains
  - Only provides a Next Hop address to a destination prefix
  - This has to be resolved to an egress interface using a second route lookup

Border Gateway Protocol

- A Routing Protocol for exchanging routing information between networks (RFC4271)
  - RFC4276 gives an implementation report on BGP
  - RFC4277 describes operational experiences using BGP
- The Autonomous System is used to uniquely identify networks with a common routing policy
- Path vector protocol (RFC1322)
  - A path vector protocol defines a route as a pairing between a destination and the attributes of the path to that destination.

Autonomous Systems

- A network - or a group of networks - controlled by a single entity
  - With the same interior and exterior routing policy
  - Under the same administrative control
  - Identified by number
- AS numbers are distributed by Regional Internet Registries
  - RIPE NCC in our region

AS Numbers

- Two ranges
  - 0 to 65535 (16-bit ASN)
  - 65536 to 4294967285 (32-bit ASN)
- Unlike IPv4 and IPv6, they are interoperable
- Special use:
  - 64496 to 64511 / 65536 to 65551 - Documentation
  - 64512 to 65534 / 4200000000 to 429496729 - Private use
  - 0, 23456, 65535 - Reserved

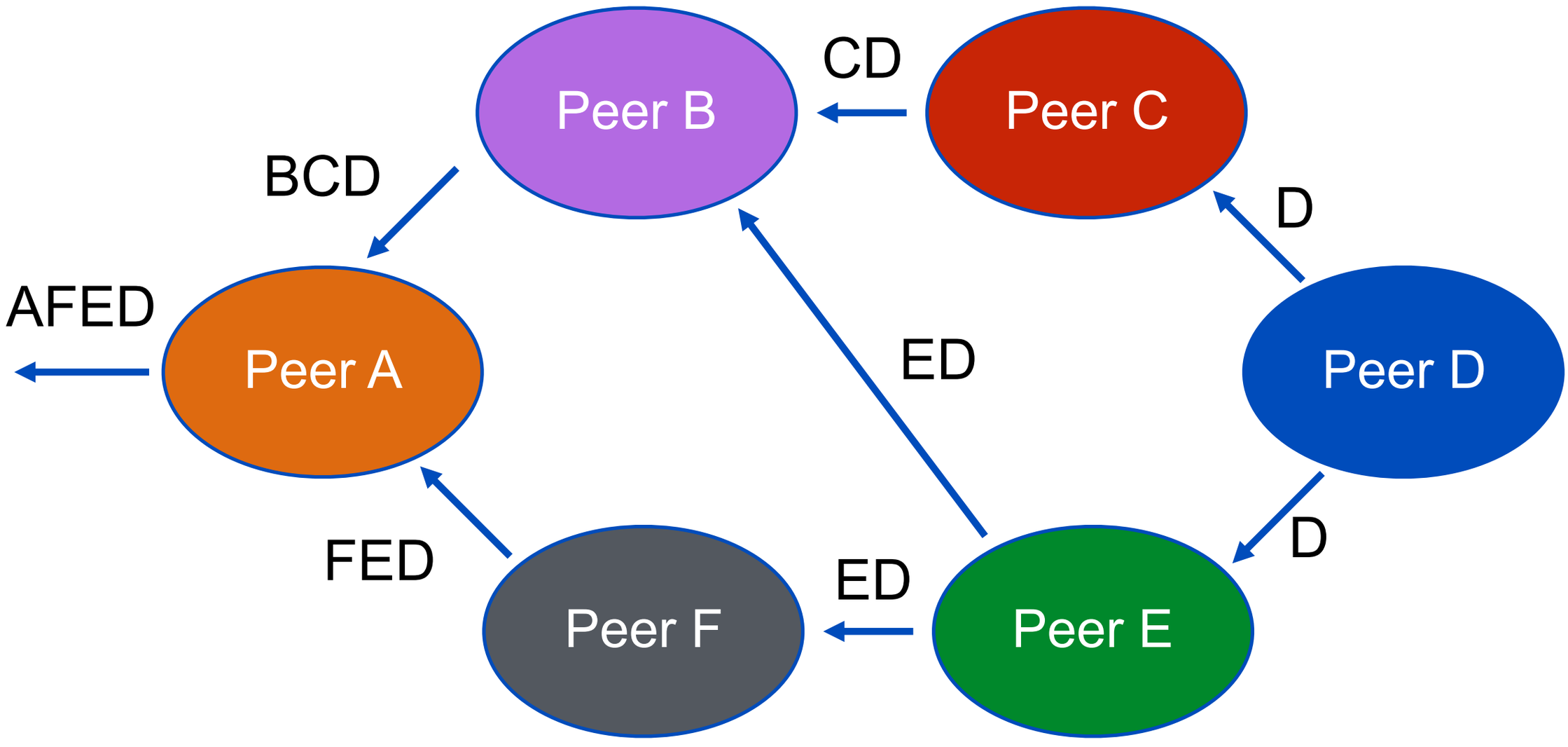
AS23456

- Nearly all software now supports 32-bit ASns
  - Unlike in the past
- AS23456 could be seen/used as a placeholder for 32 bit AS numbers
  - On devices who do not yet support AS32

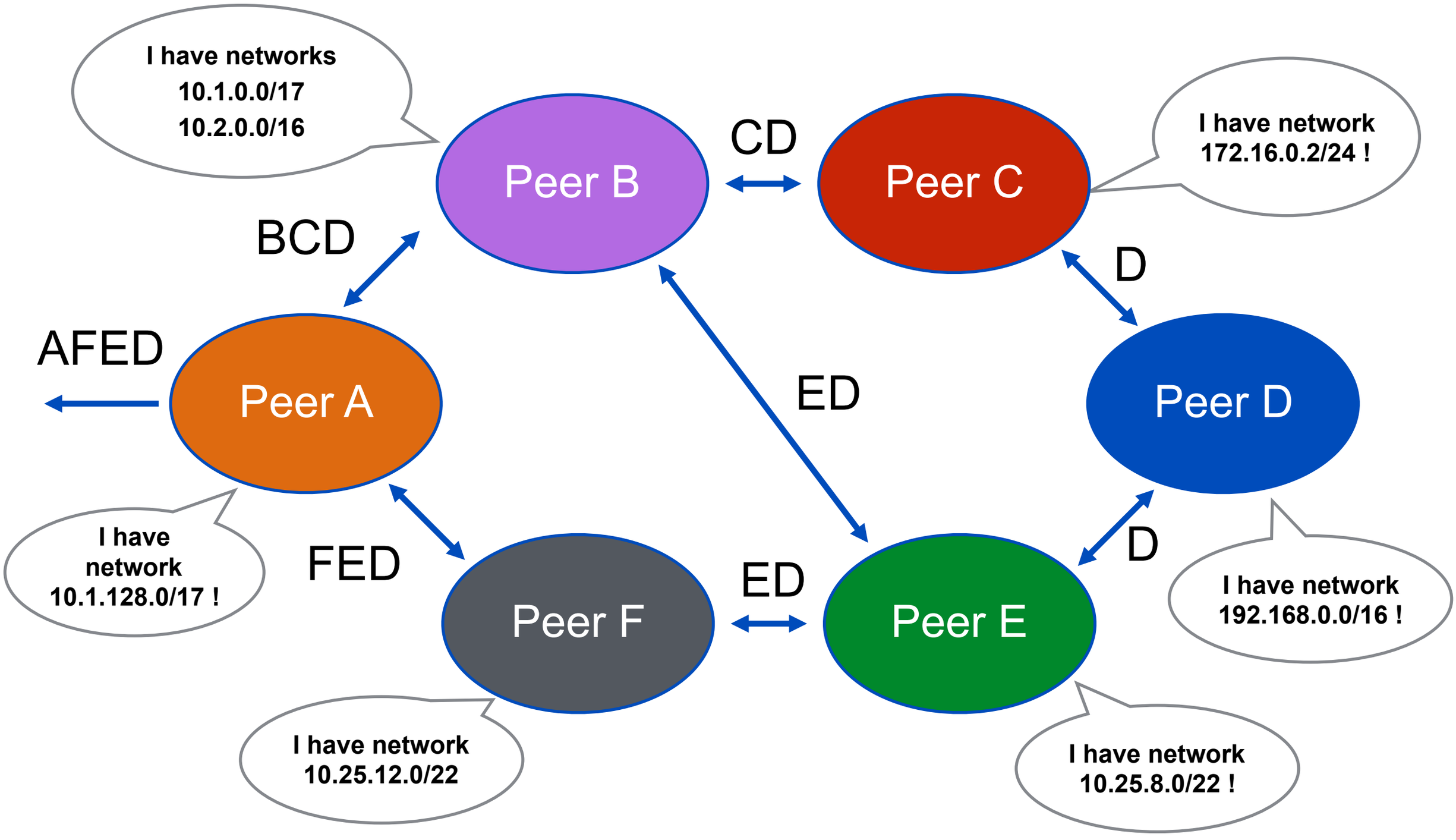
Path Vector Protocol

- AS_PATH
  - one of the attributes
  - sequence of AS numbers
- If own AS detected the path is discarded
  - simple loop detection mechanisms
- Shorter paths are preferred
  - not the most important attribute

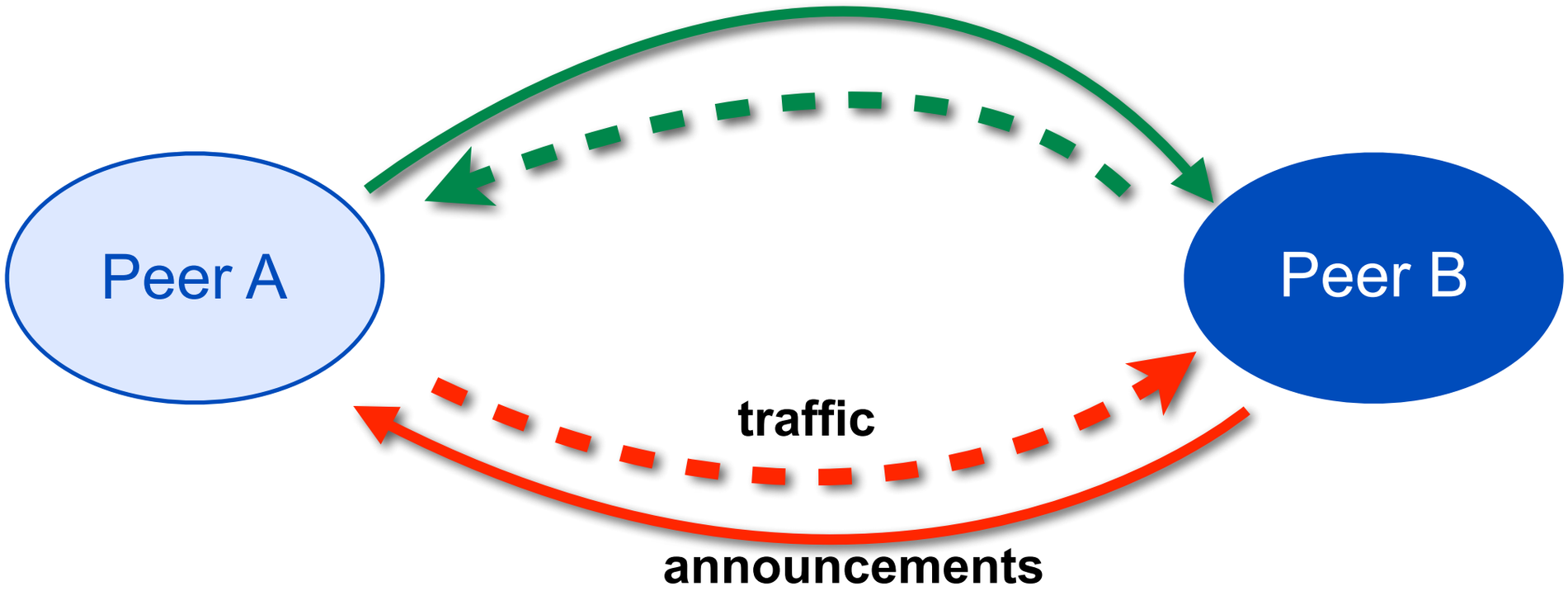
AS Path



Announcements



Traffic Direction vs Announcement



Default Free Zone

- The default free zone is made up of Internet routers which have explicit routing information about the rest of the Internet

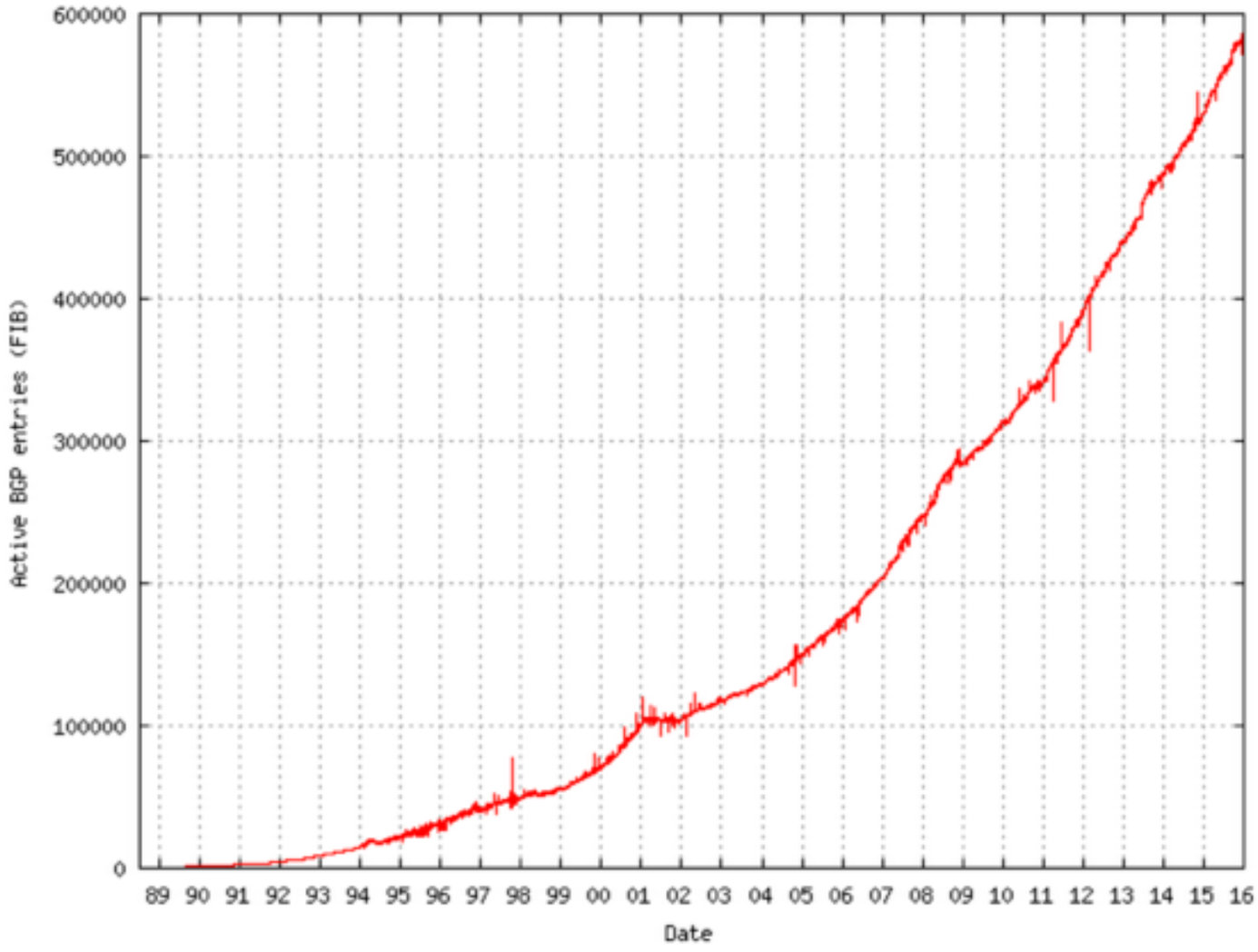
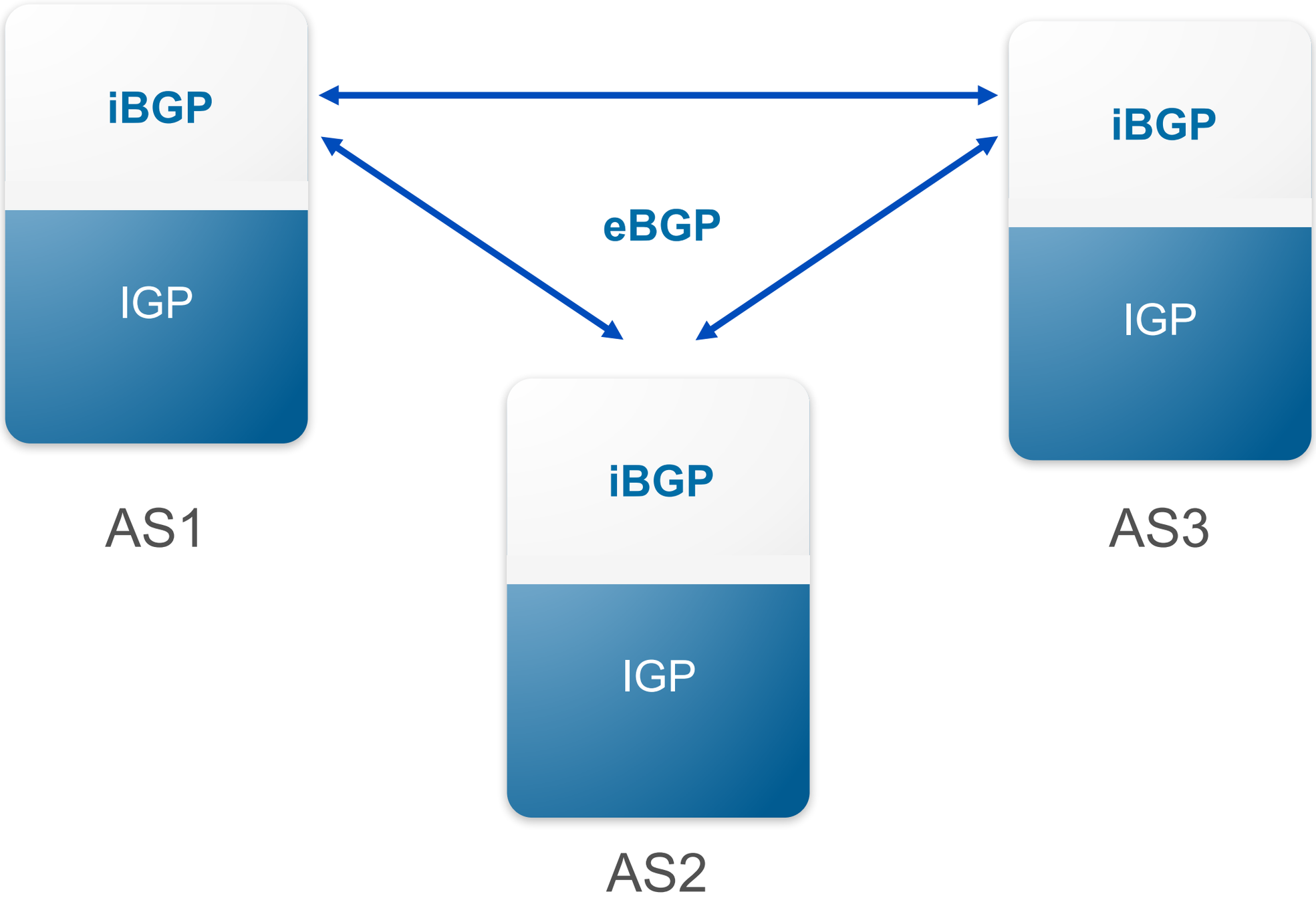


Image: http://www.cidr-report.org/

External and Internal BGP

- External BGP (eBGP)
  - BGP neighbor relationship between two peers belonging to different AS
  - Prefix interchange with external peers and upstreams
  - Most routing policy located here
- Internal BGP (iBGP)
  - BGP neighbor relationship within the same AS
  - Routes customer prefixes around internal infrastructure
  - Is NOT congruent with physical connectivity

Operator Model



Do I need BGP?

- Single homed
  - Static default route or distributed via IGP or private ASN
  - Operator takes responsibility for reachability of your prefix
- Multihomed with the same transit
  - Multiple default routes to different networks
  - Required on downstream and upstream

Do I need BGP?

- Multiple upstreams, no transit
  - BGP is optional
  - No need to receive full global routing table
  - Still control over its own routing policy
- Multiple upstreams, providing transit
  - BGP is required
  - Need to announce foreign prefixes

Connecting Outside

Exercise

Login to Labs

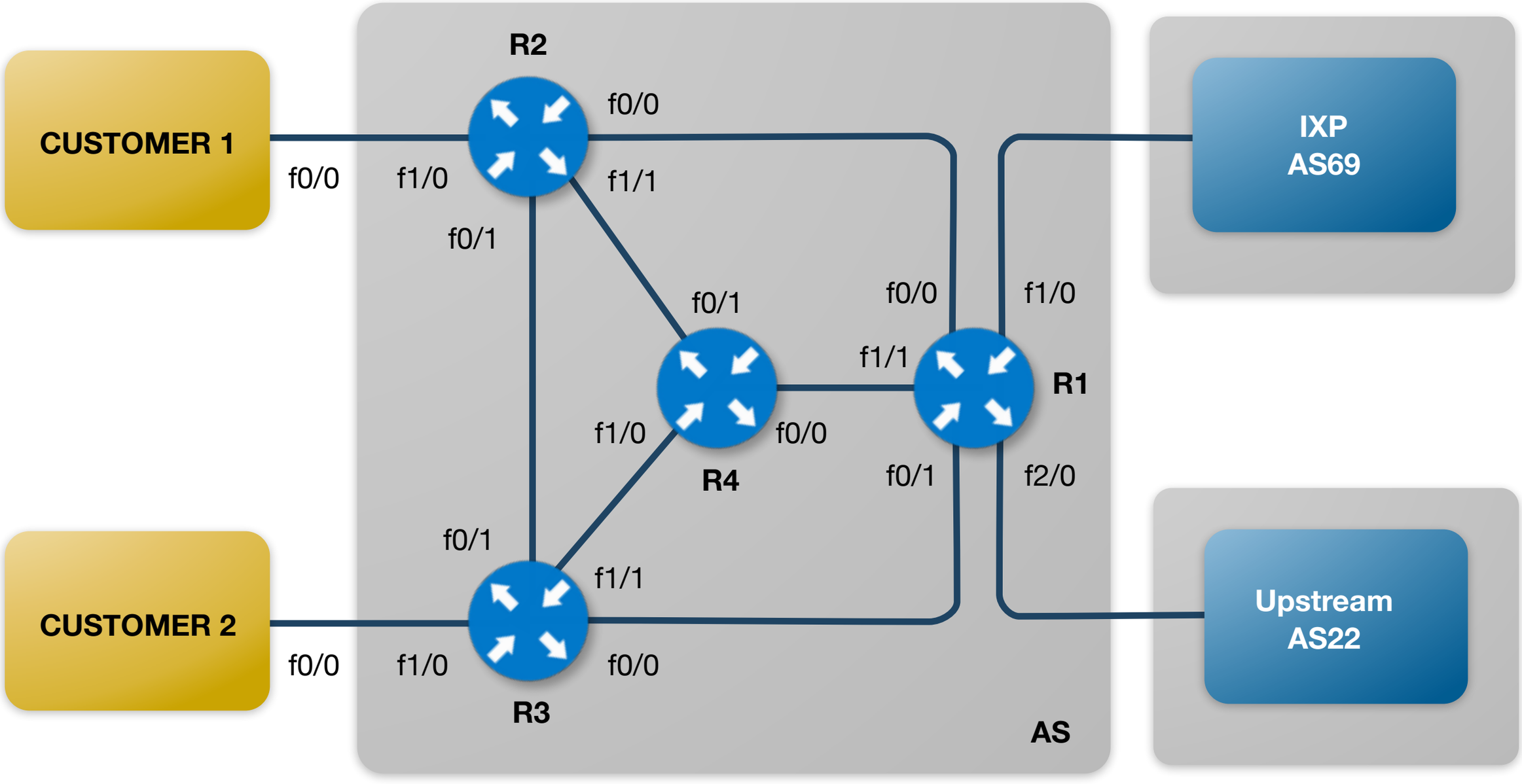
- Make sure you have connectivity
- Go to: workbench.ripe.net
  - Your login is your number on the attendee list
  - We will provide you with the password
- Read instructions carefully
  - First discover, then configure

Discover the Network

- Routing Protocol
  - IGP (OSPF) is used for loopback addresses and point-to-point links
  - No EGP (BGP) configuration

- R1 announces a default route via OSPF
  - Keeps routing tables in the area smaller
  - All inter-area traffic must pass R1

Network Diagram



Assignment

- Connect your network to transit provider
- Connect you network to Internet Exchange

- Data needed
  - Your AS number
  - Your IP address space (IPV4 and IPv6)
  - The AS number of your neighbors
  - The IP address of your neighbors BGP routers

Preparation (on R1)

- Insert static Null route
  - Before BGP advertised its network, it checks for an exact match of network number and mask on router's routing table

```
(config)# ip route 10.X.0.0 255.255.252.0 null0 250
```

Configure IXP Interface (on R1)

- Identify and enable your IXP interface

```
(config)# interface FastEthernet1/0
(config-if)# no shutdown
```

- Configure IXP interface IP address

```
(config)# interface FastEthernet1/0
(config-if)# ip address 172.16.0.X 255.255.255.0
```

- Test if IXP routers are reachable

```
# ping 172.16.0.66
# ping 172.16.0.99
```

Configure Transit Interface (on R1)

- Identify and enable your transit interface

```
(config)# interface FastEthernet2/0
(config-if)# no shutdown
```

- Configure transit interface IP address

```
(config)# interface FastEthernet2/0
(config-if)# ip address 10.132.X.2 255.255.255.252
```

- Test if transit provider router is reachable

```
# ping 10.132.X.1
```

Create a filter (on R1)

- BGP sends the best paths to all neighbours

```
(config)# ip prefix-list transit-out-v4 seq 5 permit 10.X.0.0/22
(config)# ip prefix-list ixp-out-v4 seq 5 permit 10.X.0.0/22
```

Configure Transit Session (on R1)

- Configure BGP session with AS22

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes
(config-router)# neighbor 10.132.X.1 remote-as 22
(config-router)# neighbor 10.132.X.1 prefix-list transit-out-v4 out
```

- How to advertise a route
  - redistribution
  - network statement

```
(config-router)# network 10.X.0.0 mask 255.255.252.0
```

Configure IXP Sessions

- Configure BGP session with AS69

```
(config)# router bgp 1XX
(config-router)# neighbor 172.16.0.66 remote-as 69
(config-router)# neighbor 172.16.0.66 prefix-list ixp-out-v4 out
(config-router)# neighbor 172.16.0.99 remote-as 69
(config-router)# neighbor 172.16.0.99 prefix-list ixp-out-v4 out
```

Verify

- Check sessions summary

```
# show ip bgp summary
```

- Check BGP and routing table

```
# show ip bgp
# show ip route
# show ip bgp neighbor <peer IP> advertised-routes
```

- Verify reachability

```
# ping 10.132.32.1
# ping <your colleague R1 IP>
```

- Show logged events

```
# show logging
```

BGP Operations

Section 2

ASN Types

- Multihomed
  - Multiple neighbors
- Stub
  - Single neighbor
- Transit
  - Offers connectivity between ASes
- Internet Exchange Point
  - Offers direct connectivity between ASes
  - Usually transparent

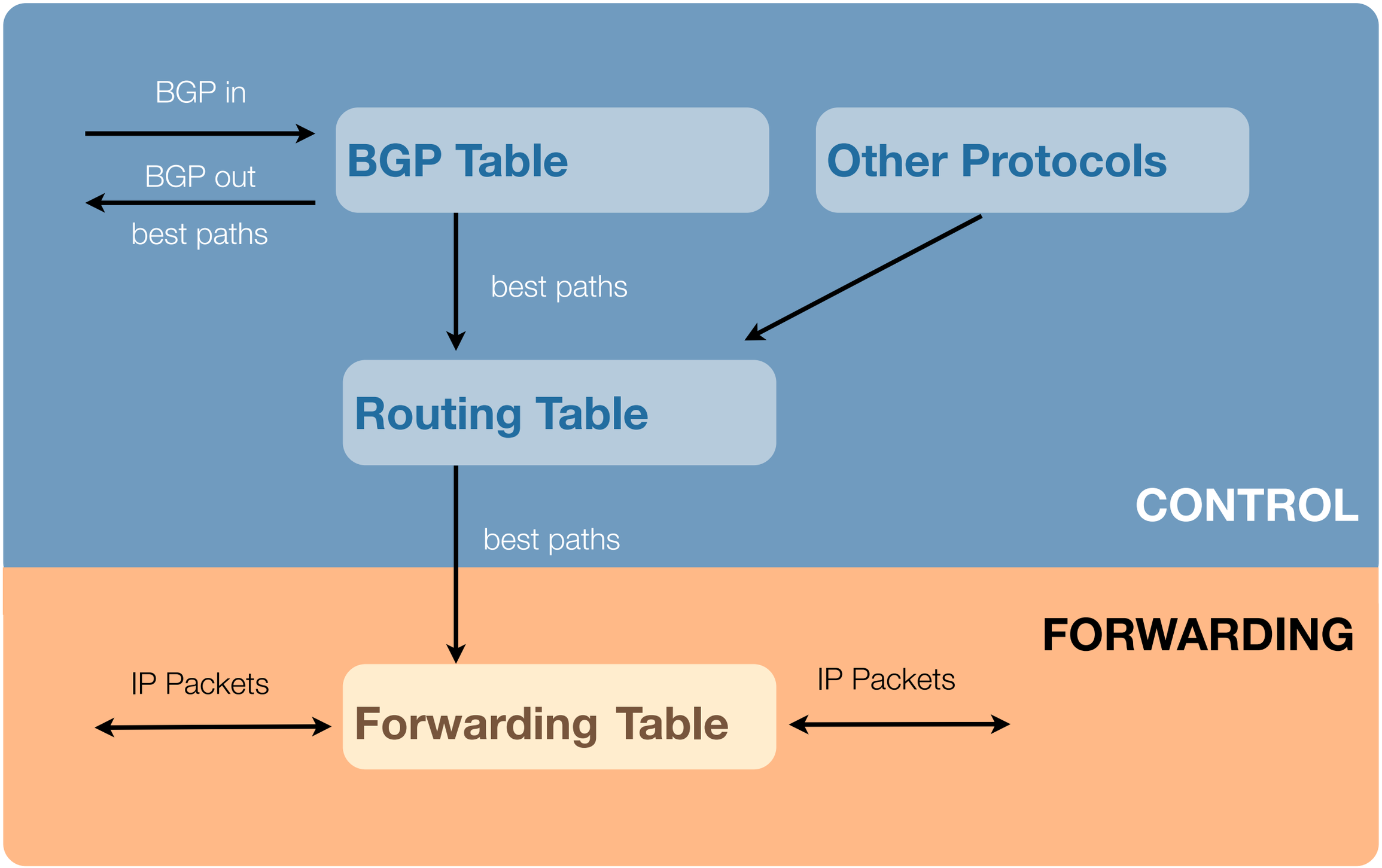
BGP Operations

- Neighbours open TCP connection (port 179)
- BGP exchanges routes with neighbours
  - Subsequently, only incremental updates are sent
- Informations about routes are kept in separate routing tables (BGP table)
  - The best path is installed in the routing table (RIB)
- Best path is sent to BGP neighbours
- BGP neighbours exchange periodically keep alive messages

BGP Messages

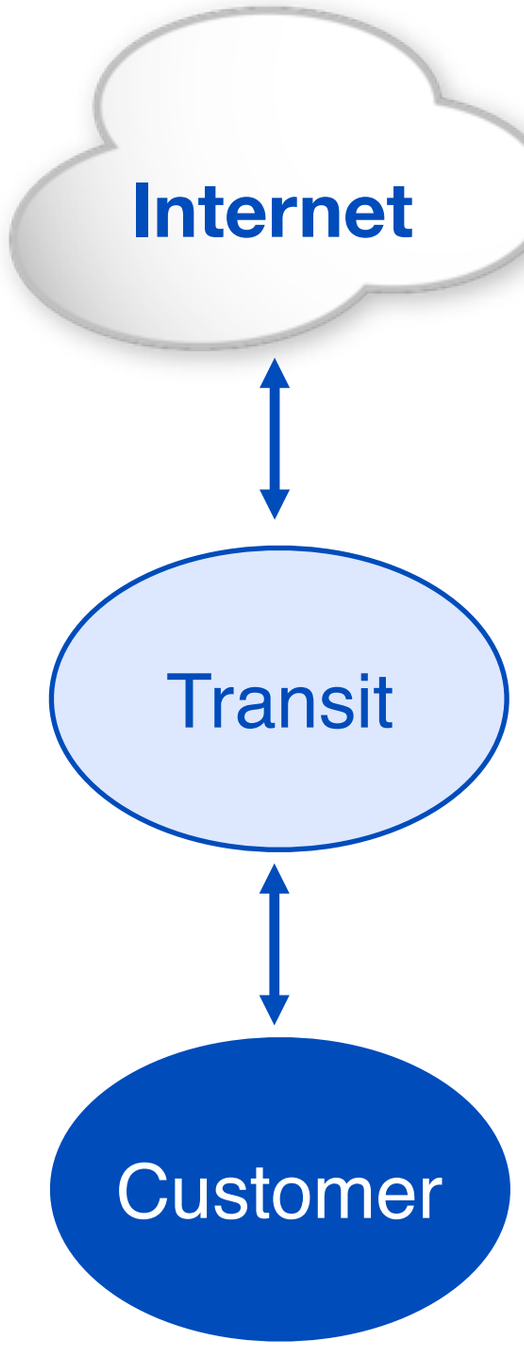
- Open
  - Information about the local BGP speaker
    - Version and hold time
    - AS number and Router id
  - BGP Capabilities Advertisement (RFC 2842)
    - Multiprotocol
    - Route Refresh
    - 32 bit ASN
- Keepalive
  - Verify BGP session
- Update
  - New or unreachable routes and path attributes
- Notification
  - Indicate an error condition

RIB and FIB


BGP in
BGP out
best paths
BGP Table
Other Protocols
best paths
Routing Table
CONTROL
best paths
FORWARDING
IP Packets
Forwarding Table
IP Packets

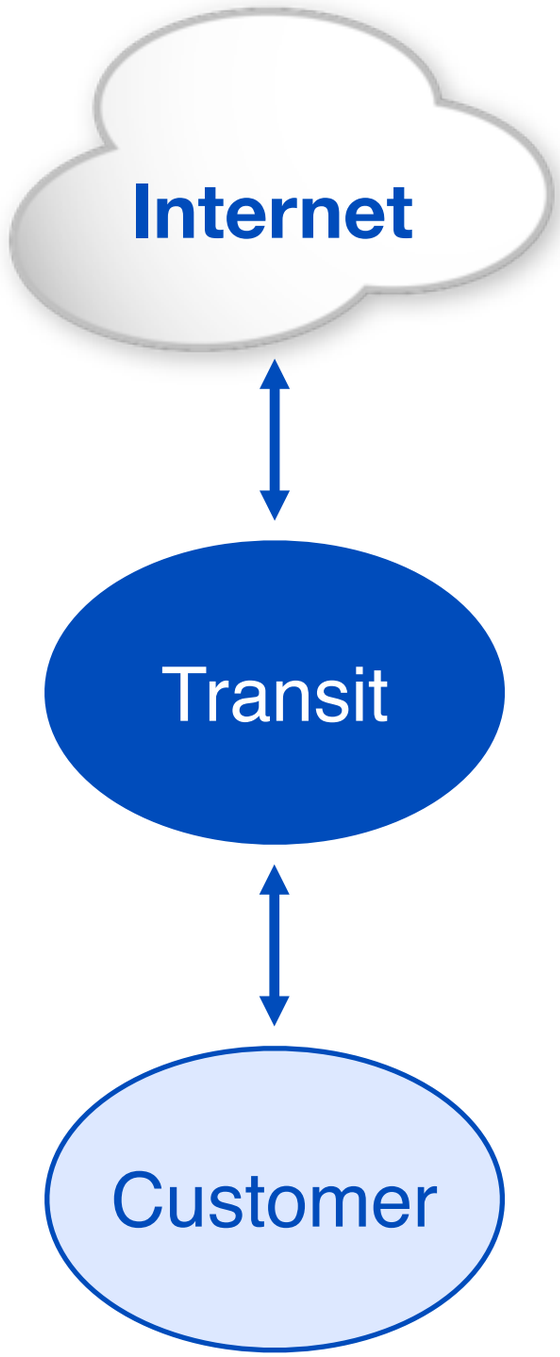
You receive BGP Transit

- “Upstream” network
- Connects you to the rest of the internet
  - By giving you a full BGP routing table
  - Or just by providing you the default route
- You announce them your prefixes
  - And your customers
  - But not your peers



You have BGP Customer

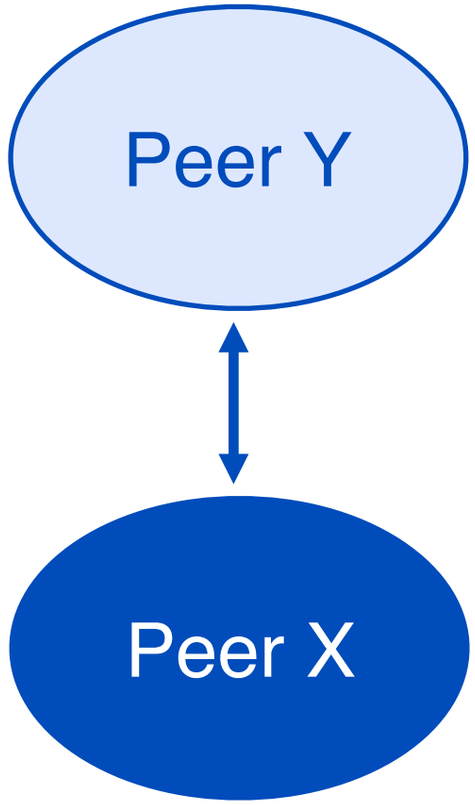
- "Downstream" network
  - You connect them to the internet
  - By providing them with a full BGP table
  - or a default route
- You generally receive from them only their routes
  - And/or their customers'



BGP Peering

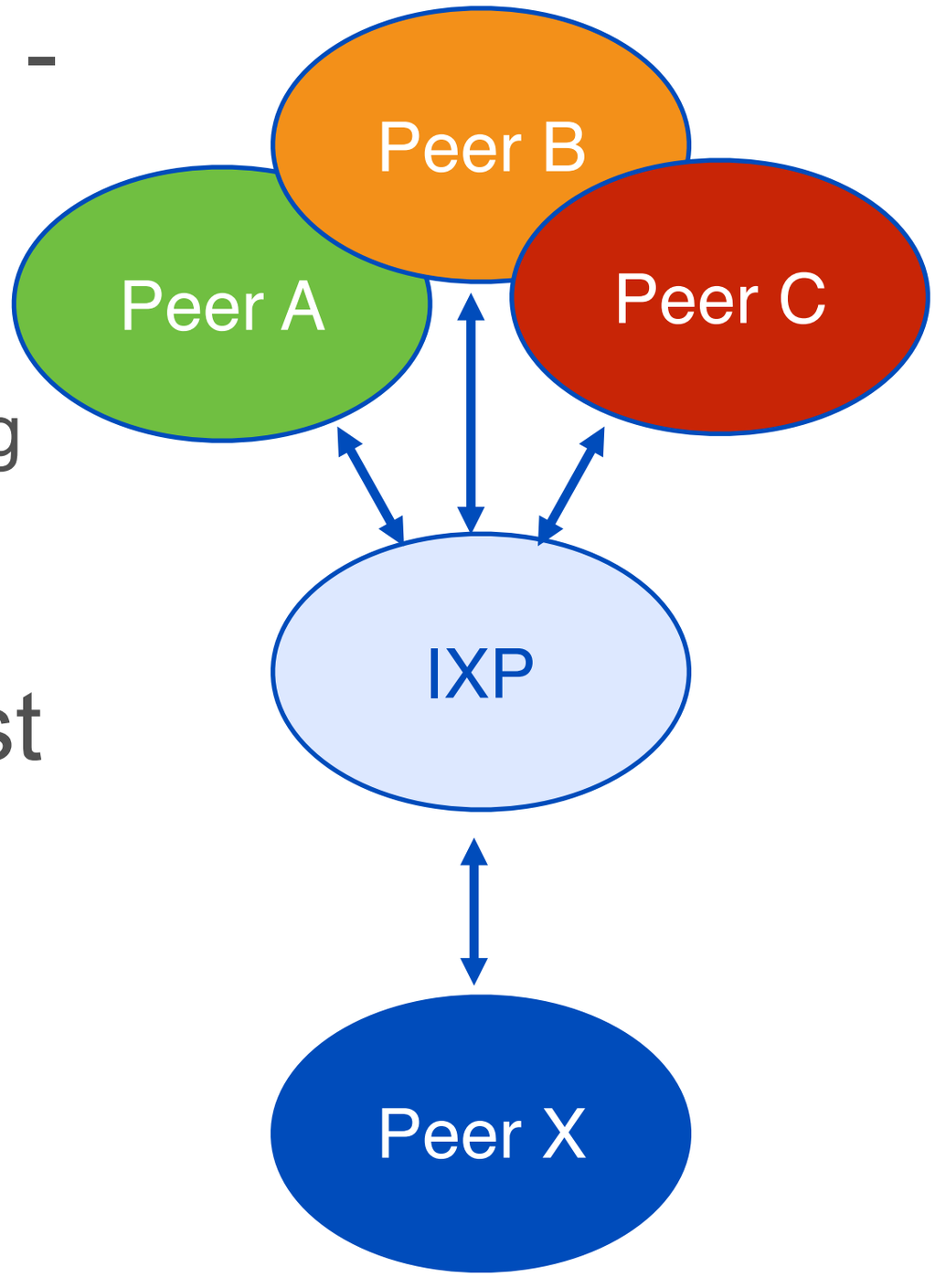
- Usually peer with you at Internet Exchanges
  - Gives you access to its network
  - And/or its customers

- You announce them only your route
  - And your customers



Internet Exchanges (IX or IXP)

- A switch - or set of switches - that permit members to exchange traffic directly
  - Meeting point through BGP peering



- Many countries have at least one
  - AMS-IX, LINX, VIX, MIX, etc

Internet Exchanges - Why

- IXes enable traffic to remain local
  - Improves routing efficiency and fault-tolerance
  - Reducing the average per-bit delivery cost (no transit)
- Often non-profit, membership organisations
- Cater to the local ISPs, Content Providers, Academy, Governments, Others
  - But also to big networks

Internet Exchanges - Architecture

- A switch, or a group of switches
  - Range is generally from 100Mb to 100Gb ports
- Switches are in colocation facilities
  - Easy to reach them
  - Can be spread in different facilities across a city or region
- Some IXes have two LANs for redundancy

Route Servers

- A server running a BGP Daemon

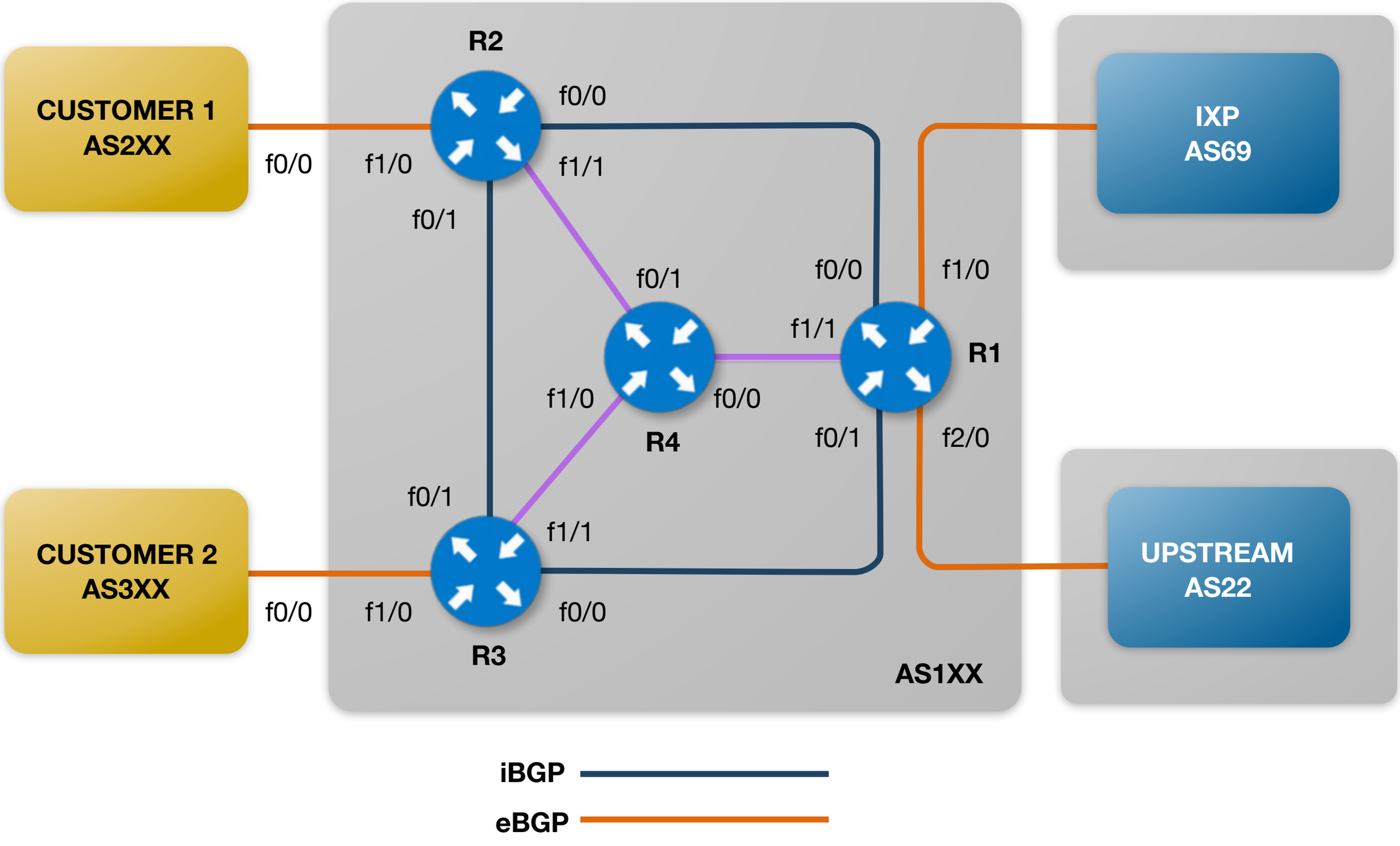
- Helps networks who peer at many IXes
  - Avoids setting up a meshed environment
  - Eases management

- Sets next-hop as announcer, leaving itself out
  - Traffic does not flow through the route server

Connecting BGP Customers

Exercise

Network Diagram



Assignment

- Connect Customer 1 and 2 using BGP
  - Customers will use prefixes from your address space

- Data needed
  - Your AS number
  - Your IP address space
  - The AS number of your customers
  - The IP address of your customers BGP routers

Preparation

- Remove default routes from C1 and C2

```
(config)# no ip route 0.0.0.0 0.0.0.0
```

Using loopbacks

- Better to use Loopback address than Interface address
  - Session is not dependent on state of a single interface
  - Session is not dependent on physical topology
- Can be propagated by IGP
  - IS-IS or OSPF

iBGP Configuration R1

- BGP configuration of Router 1 on top of IP core

```
(config)# router bgp 1XX
(config-router)# neighbor 172.X.255.2 remote-as 1XX
(config-router)# neighbor 172.X.255.2 update-source lo0
(config-router)# neighbor 172.X.255.2 next-hop-self
(config-router)# neighbor 172.X.255.3 remote-as 1XX
(config-router)# neighbor 172.X.255.3 update-source lo0
(config-router)# neighbor 172.X.255.3 next-hop-self
```

BGP Configuration R2 and C1

- BGP configuration of Router 2

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes
(config-router)# network 10.X.0.0 mask 255.255.252.0
(config-router)# neighbor 172.X.255.1 remote-as 1XX
(config-router)# neighbor 172.X.255.1 update-source lo0
(config-router)# neighbor 172.X.255.1 next-hop-self
(config-router)# neighbor 172.X.255.3 remote-as 1XX
(config-router)# neighbor 172.X.255.3 update-source lo0
(config-router)# neighbor 172.X.255.3 next-hop-self
(config-router)# neighbor 10.X.0.26 remote-as 2XX
```

- BGP configuration of Customer 1

```
(config)# router bgp 2XX
(config-router)# bgp log-neighbor-changes
(config-router)# network 10.X.1.0 mask 255.255.255.0
(config-router)# neighbor 10.X.0.25 remote-as 1XX
```

BGP Configuration R3 and C2

- BGP configuration of Router 3

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes
(config-router)# network 10.X.0.0 mask 255.255.252.0
(config-router)# neighbor 172.X.255.1 remote-as 1XX
(config-router)# neighbor 172.X.255.1 update-source lo0
(config-router)# neighbor 172.X.255.1 next-hop-self
(config-router)# neighbor 172.X.255.2 remote-as 1XX
(config-router)# neighbor 172.X.255.2 update-source lo0
(config-router)# neighbor 172.X.255.2 next-hop-self
(config-router)# neighbor 10.X.0.30 remote-as 3XX
(config-router)# neighbor 10.X.0.30 default-originate
```

- BGP configuration of Customer 2

```
(config)# ip prefix-list GW seq 5 permit 0.0.0.0/0
(config)# router bgp 3XX
(config-router)# bgp log-neighbor-changes
(config-router)# network 10.X.2.0 mask 255.255.255.0
(config-router)# neighbor 10.X.0.29 remote-as 1XX
(config-router)# neighbor 10.X.0.29 prefix-list GW in
```

Verify

- Check sessions in summary

```
# show ip bgp neighbors | include BGP
```

- Check BGP and routing table

```
# show ip bgp
# show ip route
```

- Verify reachability from customer

```
# ping 10.132.32.1
# ping 1.1.1.1
```

- Show logged events

```
# show logging
```

Create a filter for customers (on R1)

- Allow BGP to send paths of the customers

```
(config)# ip prefix-list transit-out-v4 seq 10 permit 10.X.1.0/24
(config)# ip prefix-list transit-out-v4 seq 15 permit 10.X.2.0/24
(config)# ip prefix-list ixp-out-v4 seq 10 permit 10.X.1.0/24
(config)# ip prefix-list ixp-out-v4 seq 15 permit 10.X.2.0/24
```

- And clear all the sessions

```
# clear ip bgp 10.132.X.1 soft out
# clear ip bgp 172.16.0.66 soft out
# clear ip bgp 172.16.0.99 soft out
```

BGP Attributes

Section 3

BGP Attributes

- Every prefix has a number of attributes
  - BGP packs multiple prefixes in single update packet associated with the same attributes
- Used by local AS and remote AS for traffic engineering

BGP Attributes Classification

- Well-known mandatory
  - In every update
  - Compatible with all BGP implementations
  - Example: AS_PATH
- Well-known discretionary
  - Might be but doesn’t have to be in every update
  - Have to be compatible with all BGP implementations
  - Example: LOCAL_PREF

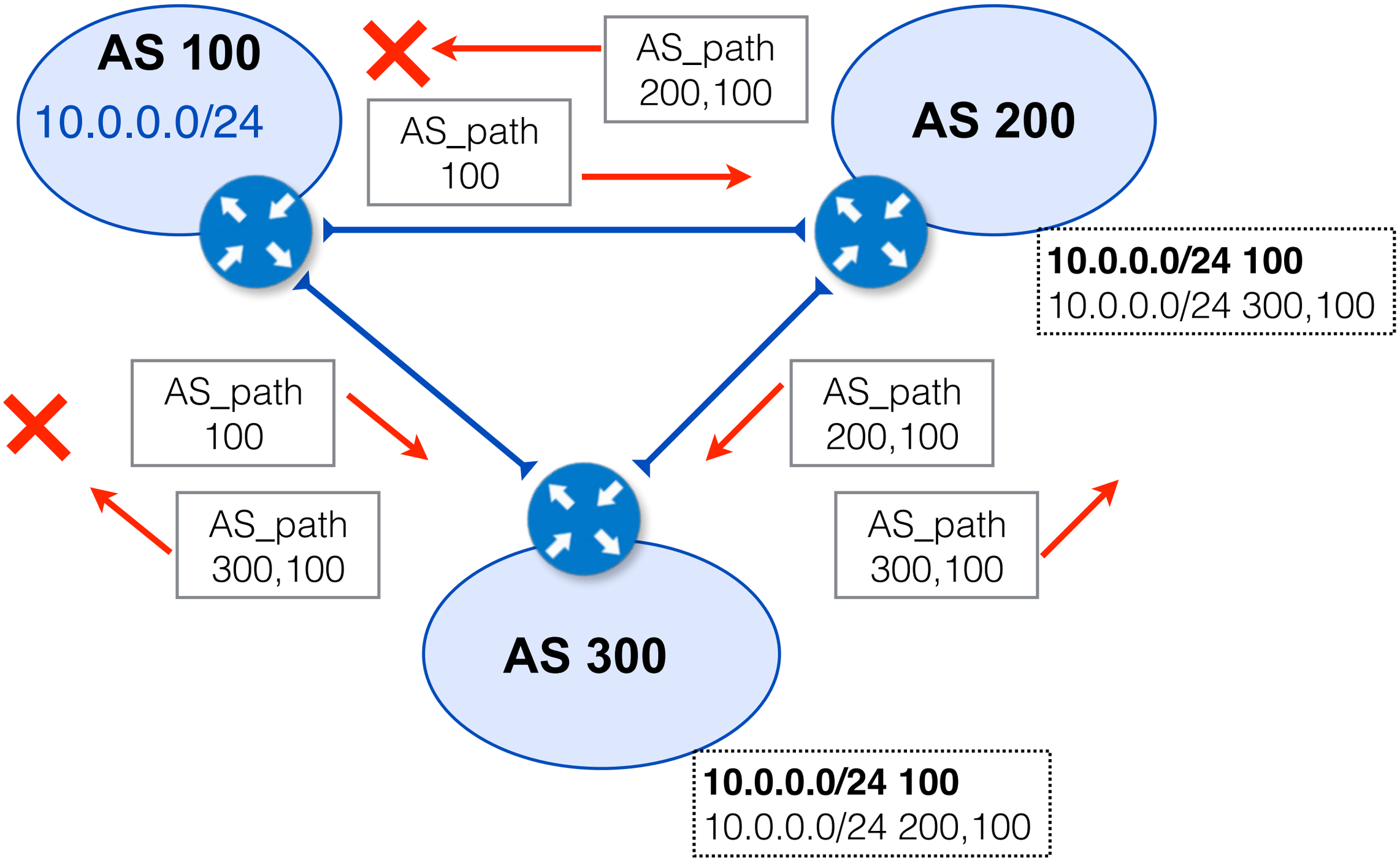
BGP Attributes Classification

- Optional transitive
  - Might be but doesn’t have to be in every update
  - Doesn’t have to be compatible with all BGP implementations if not recognised marked as partial
  - Example: COMMUNITY
- Optional non-transitive
  - Might be, but doesn’t have to be in every update
  - Doesn’t have to be compatible with all BGP implementations and exchanged only by neighbours in AS
  - Example:MED

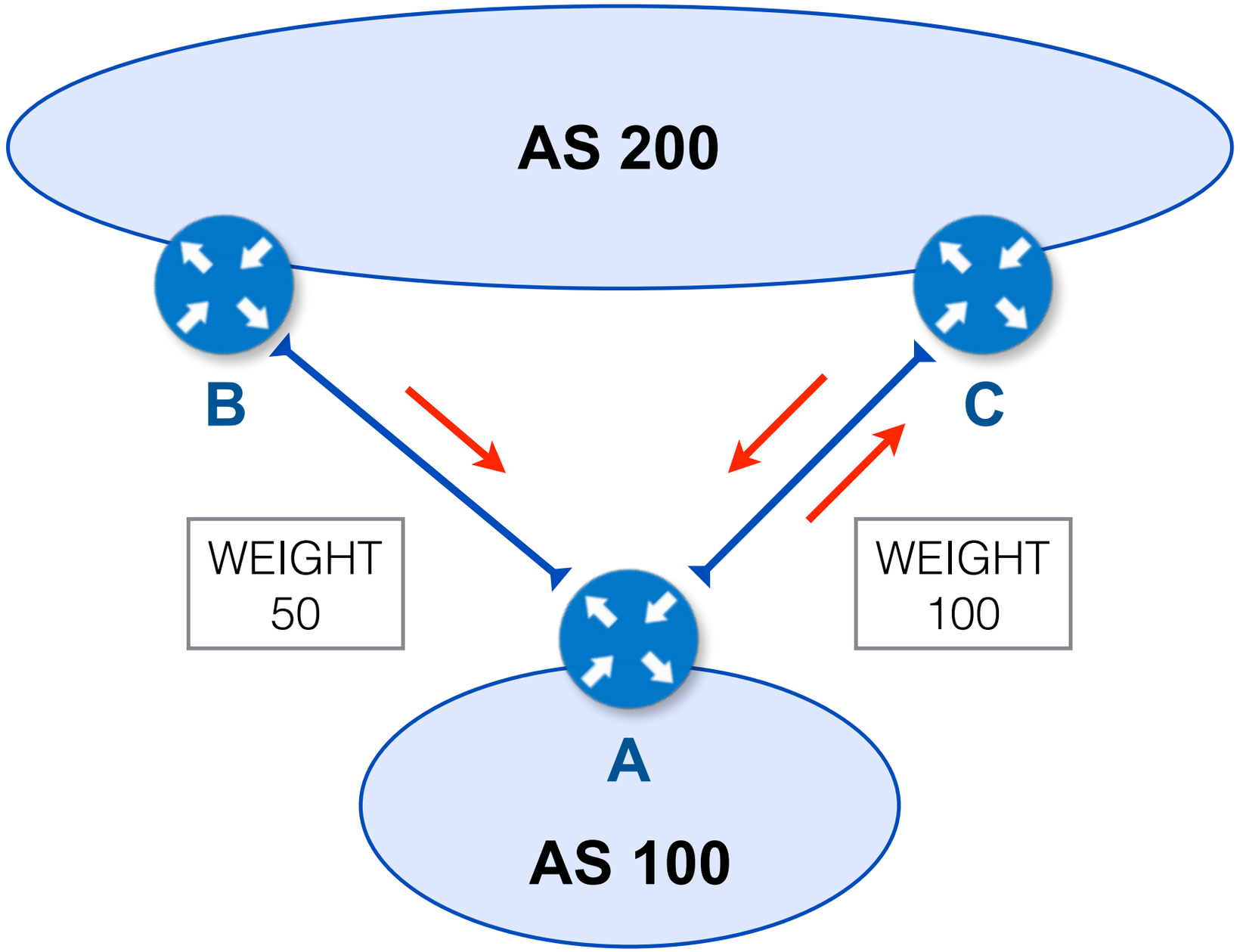
Attribute Propagation

Router:	Local AS:	Local AS + neighbour:	Not limited:
weight	local-pref updated: Next-hop AS-Path	MED	origin communities

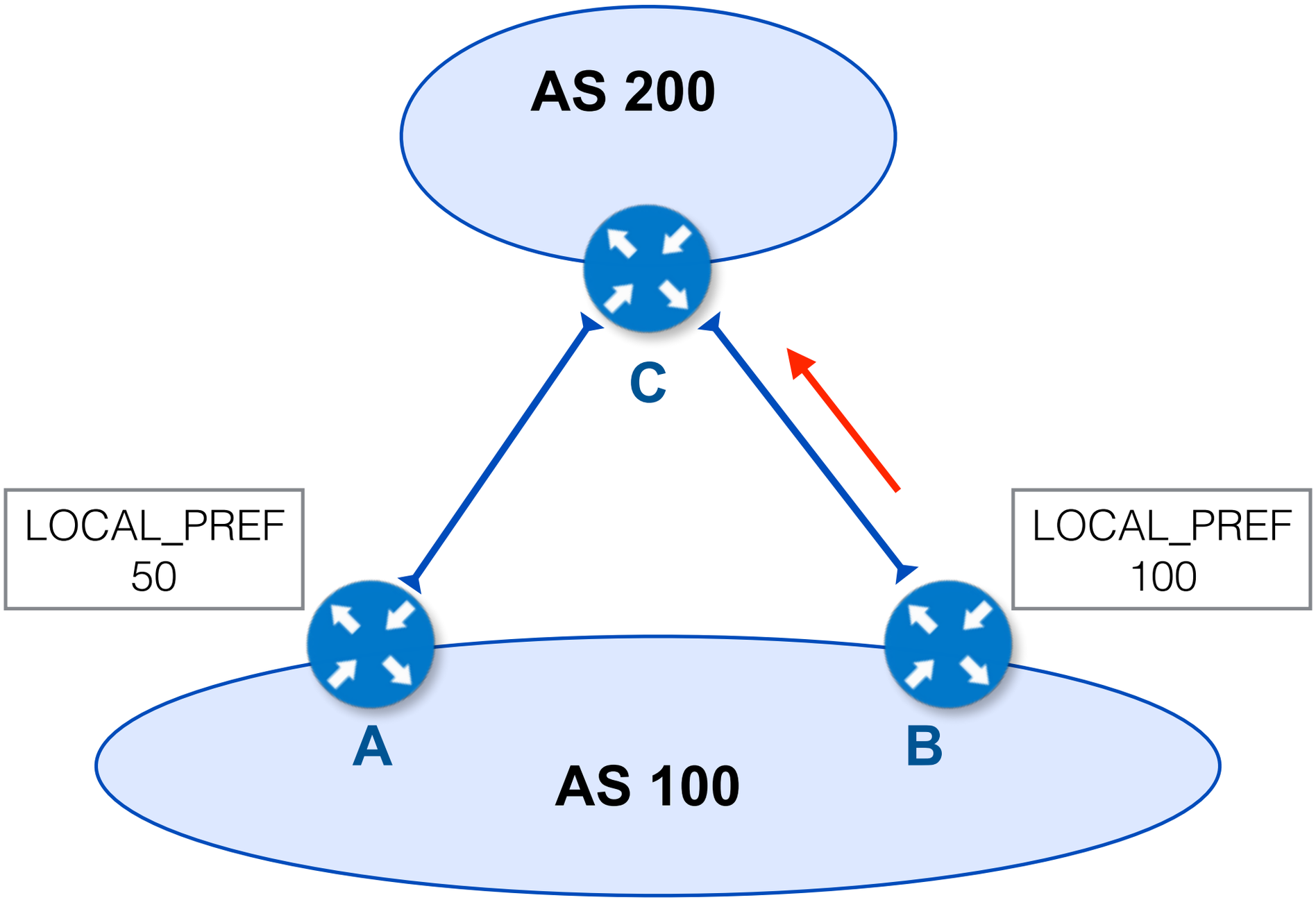
AS Path


AS 100
10.0.0.0/24
AS_path
200,100
AS_path
100
AS 200
10.0.0.0/24 100
10.0.0.0/24 300,100
AS_path
100
AS_path
200,100
AS_path
300,100
AS_path
300,100
AS 300
10.0.0.0/24 100
10.0.0.0/24 200,100

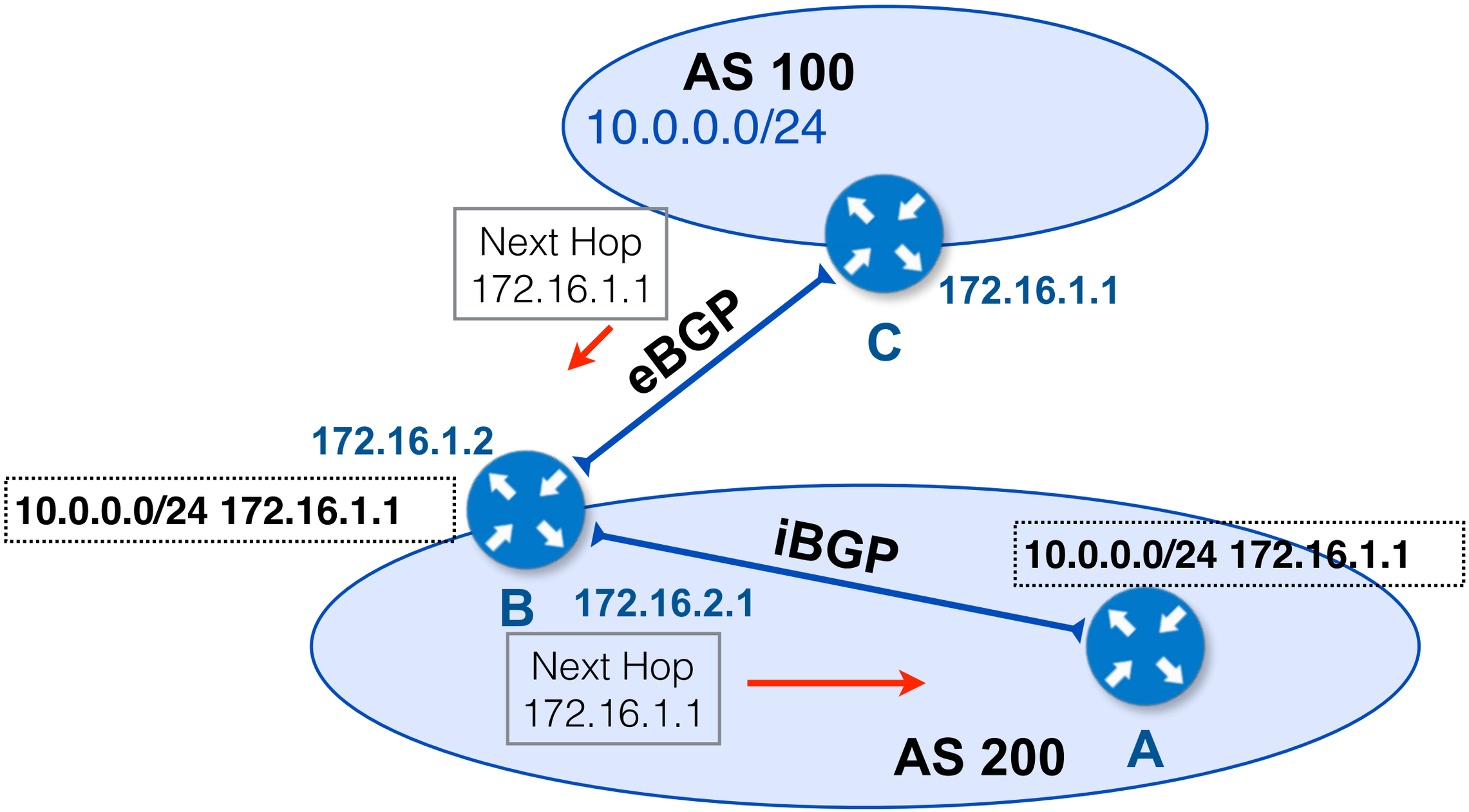
Weight



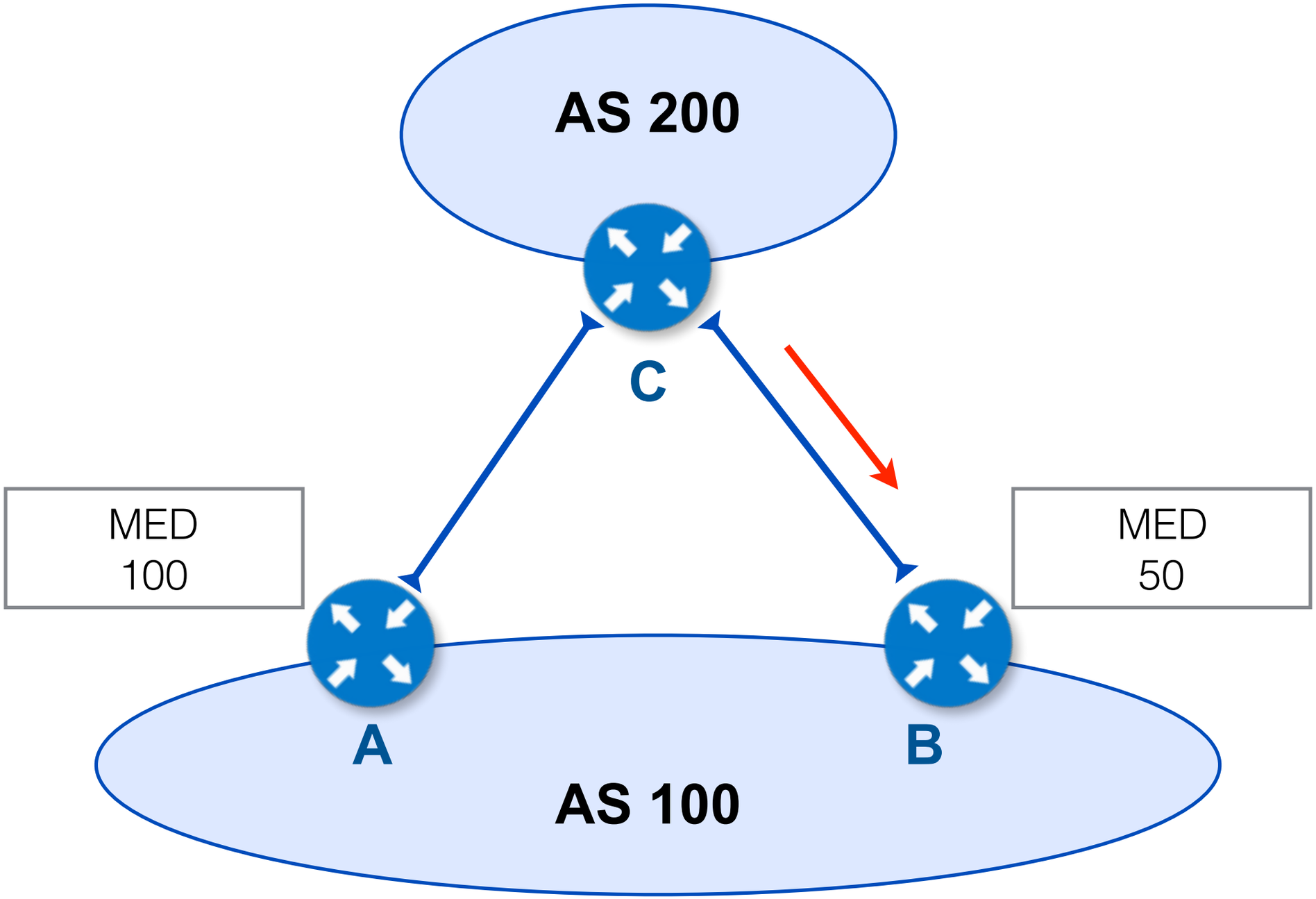
Local Preference



Next Hop


AS 100
10.0.0.0/24
Next Hop
172.16.1.1
eBGP
172.16.1.1
C
172.16.1.2
10.0.0.0/24 172.16.1.1
iBGP
10.0.0.0/24 172.16.1.1
B
172.16.2.1
Next Hop
172.16.1.1
AS 200
A

MED



Origin

- “Historical” attribute
- Three values: IGP, EGP, incomplete
  - IGP – generated by BGP network statement
  - EGP – generated by EGP
  - incomplete – redistributed from another routing protocol

Communities

- Community is a tagging technique to mark a set of routes
  - 32-bit number, the most-significant 16 bits, by convention, represent an AS number (<local-ASN>:<value>)
  - Neighbor routers can use these tags to apply specific routing polices within their network
- Predefined community attributes:
  - www.iana.org/assignments/bgp-well-known-communities

Extended Communities

- Communities are widely used for encoding operator routing policy but most-significant 16-bits, by convention, represent an AS number
  - 32-bit ASNs cannot be encoded
- Extended Communities are 64-bit (RFC4360):
  - An extended range, ensuring that communities can be assigned for many uses, without fear of overlap
  - The addition of a Type field provides structure for the community space

Routing Policy

- A routing policy describes how a network works
  - Who do you connect with
  - Which prefixes or routes do you announce
  - Which routes do you accept from others
  - What are your preferences
- In your router, this is your BGP configuration
  - neighbours
  - route-maps
  - prefix lists

Traffic Engineering

Section 4

Why do Traffic Engineering?

- Manage your capacity
- Ensure service quality
- Manage service cost
- Recover from failures

Intra-domain Traffic Engineering

- You control the network:
  - You know the reliability of the network
  - You know the price of all paths
  - IGP knows the capacity and reliability of all paths and let you map price, reliability, capacity to shape routing using cost

Inter-domain Traffic Engineering

- You DO NOT control the network
  - BGP have no metrics, capacity or cost
  - High volume of traffic and number of routes with simplicity of the protocol imposes some limitations
- Large volume of informations pass small number of ASNs
  - Tier 1,2,3 operators
  - Internet Exchange Points

The BGP Decision Algorithm

- BGP router receives new destinations from neighbors, the protocol will have to decide which paths to choose
  - Only single path to reach a specific destination is needed
  - The decision process is based on attributes
  - The best path will get propagated to its neighbors

Best Path Calculation

- Drop if own AS in AS_PATH
- Prefer path with highest WEIGHT
- Highest LOCAL_PREF
- Shortest AS_PATH
- Prefer IGP ORIGIN
- Lowest MED
- Prefer eBGP over iBGP

Best Path Calculation - Tiebreakers

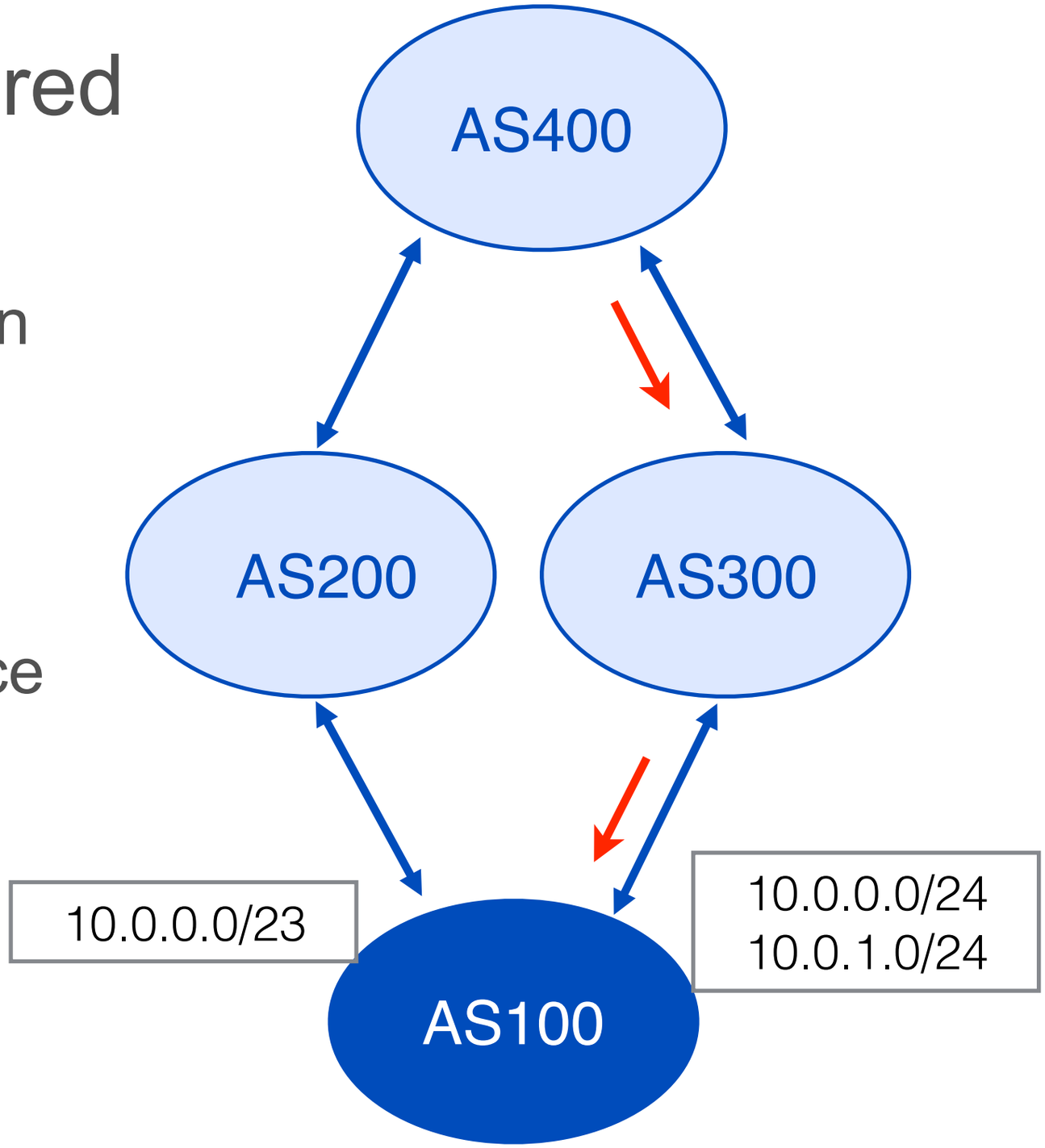
- Path with shortest next hop metric (minimum IGP cost)
- Oldest received path
- Lowest router ID
- Path from lowest neighbour address

Local Preference and Weight

- Outbound traffic control
  - Prefer to send traffic to customers, peerings, then transits
  - Requires additional inbound TE to avoid asymmetric traffic
  - Requires tight capacity management on all peerings
- Use hot potato routing for best effect
  - Nearest exit routing

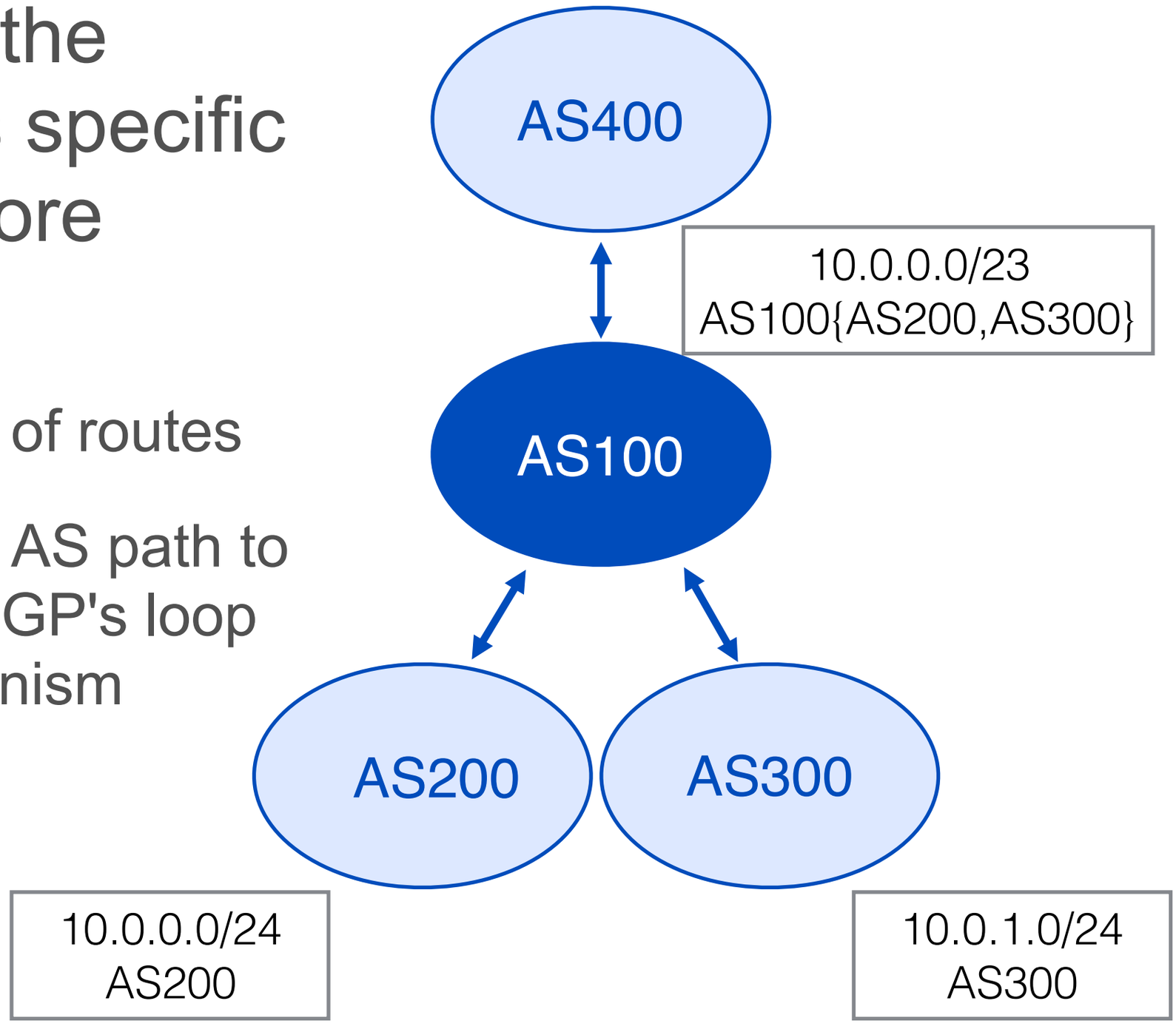
More specific announcements

- Prefix length considered before BGP
  - Considered rude and often filtered
- Effective tool
  - Might be used to announce regional prefix
  - Never announce globally



Aggregation

- Aggregation is the creation of less specific routes when more specific exist:
  - Reducing number of routes
  - Injecting AS set in AS path to keep integrity of BGP's loop prevention mechanism

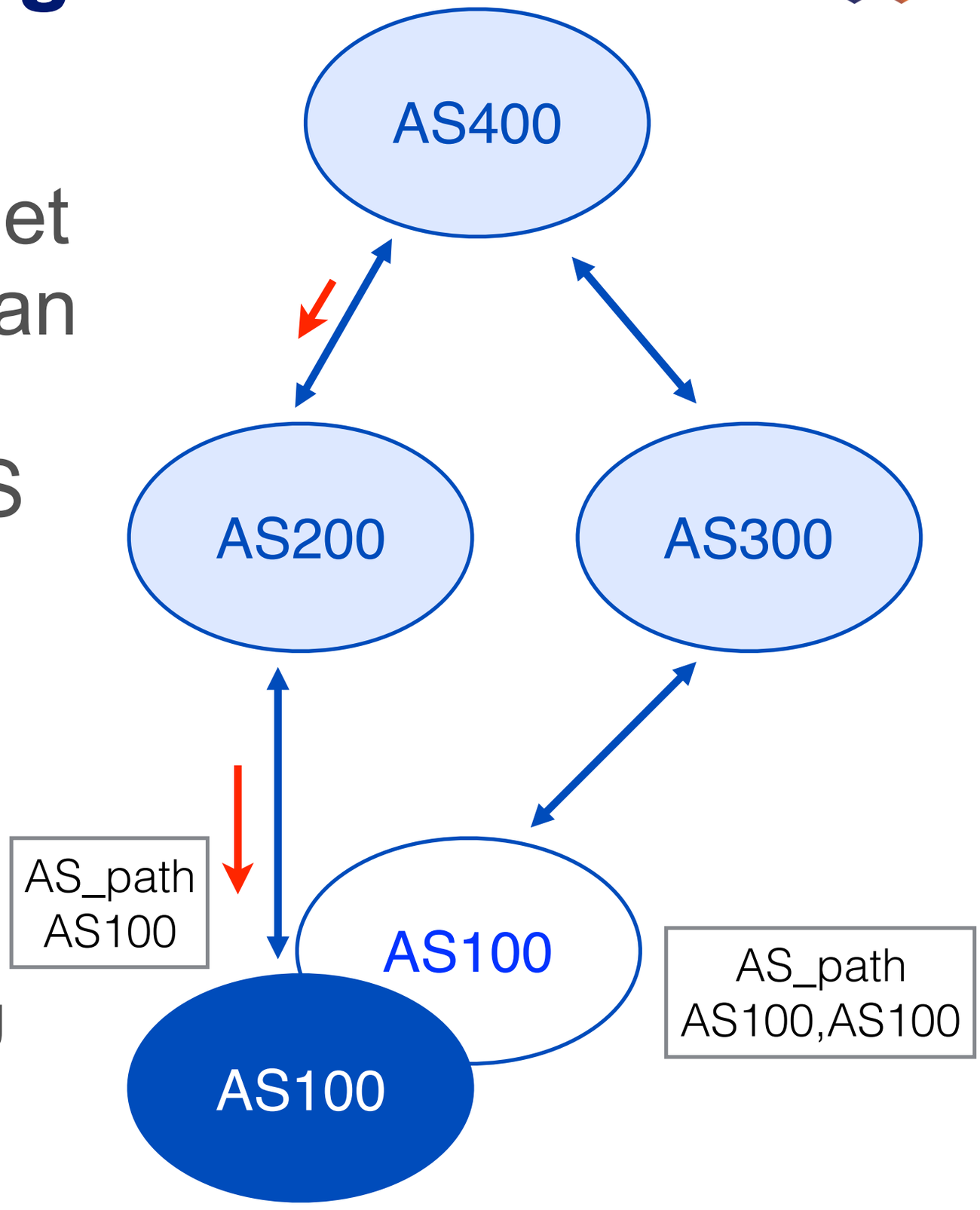


Anycast

- Just a configuration methodology
  - Announce the same prefix from multiple locations
  - The routing infrastructure directs any packet to the topologically nearest router (AS path)
  - Mentioned, although not described in detail, in many RFCs
- Used for redundancy, reduced latency
  - Not a protocol, not a different version of IP
  - Doesn’t require any special hardware or software capabilities
  - Doesn’t break or confuse existing infrastructure

AS Path Prepending

- BGP prefers the shortest AS path to get to a destination we can manipulate this by virtually extending AS path
  - Very often marginal effect
  - Requires continuous monitoring
  - Very hard Load Balancing



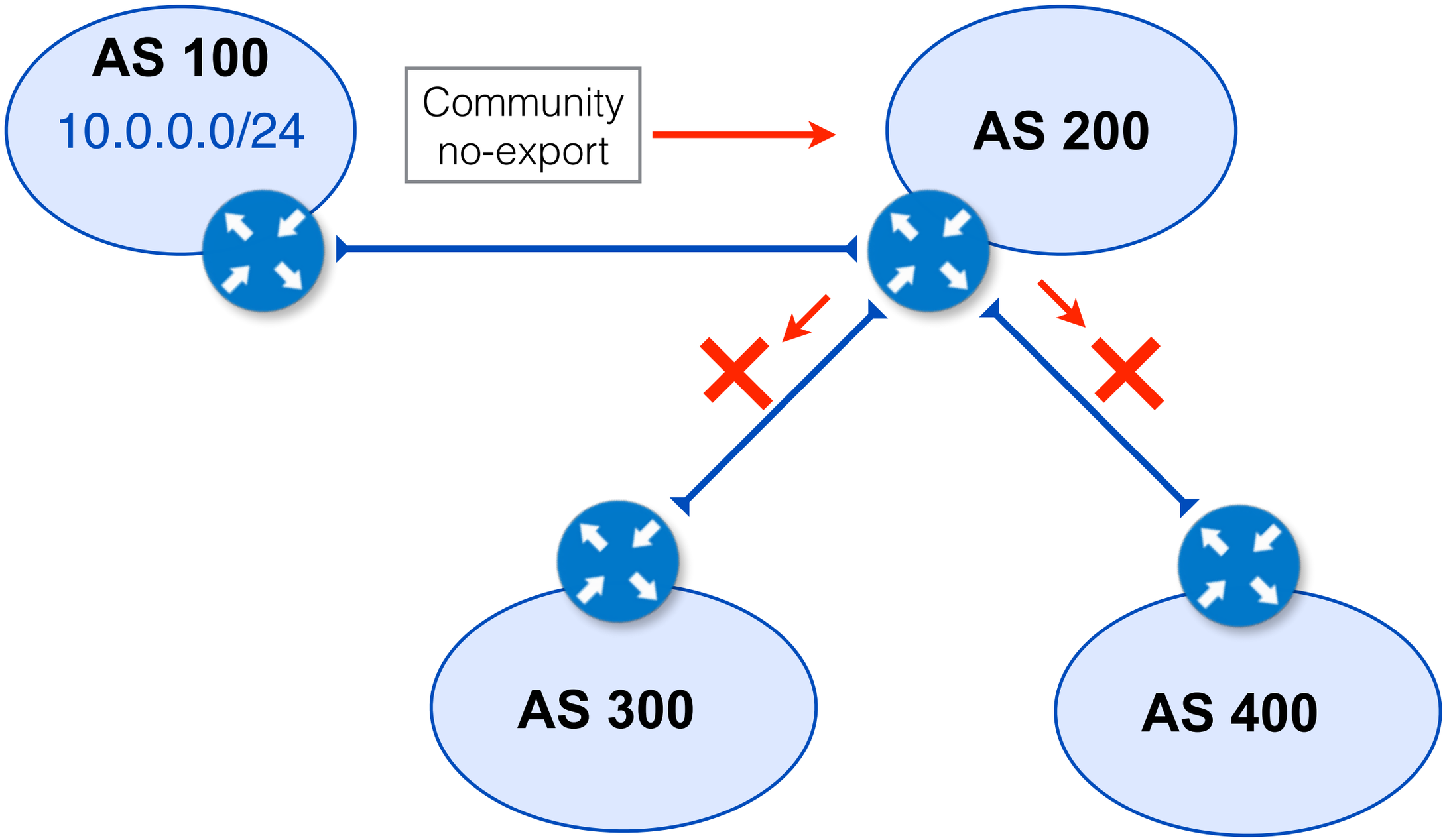
Communities Usage

- Assign prefixes to pre-defined groups
  - Local significance only
- Control how prefix is advertised by peer
  - Control your neighbors LOCAL_PREF for the specific prefix
  - Signal neighbor to prepend multiple ASNs to AS_PATH
  - Blackhole all traffic to specific prefix

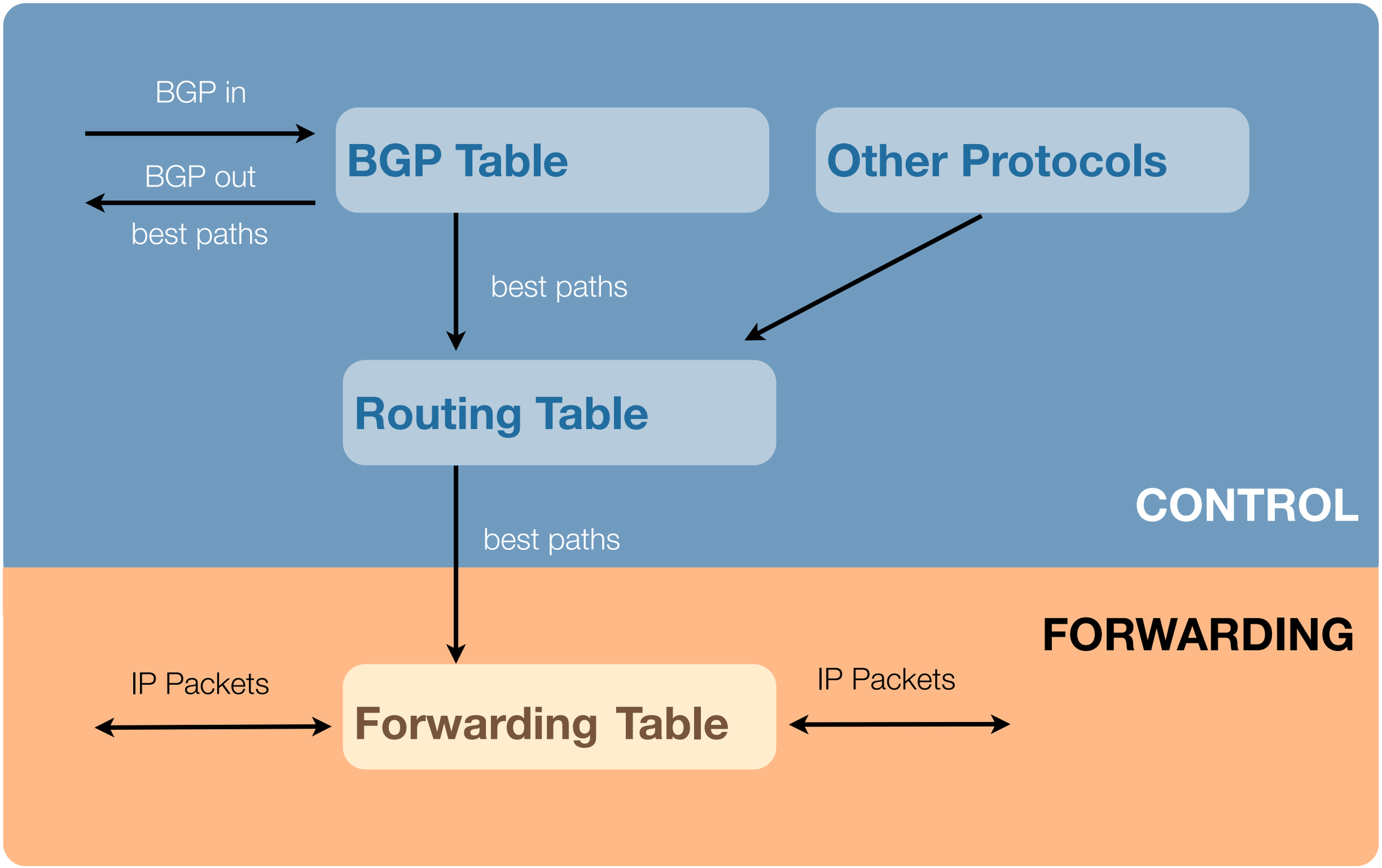
Well-Known Communities

- 65535:65281 - no-export
  - do not advertise to any eBGP peers
- 65535:65282 - no-advertise
  - do not advertise to any BGP peer
- 65535:65283 - no-export-subconfed
  - do not advertise outside local AS (confederations)
- 65535:65284 - no-peer
  - do not advertise to bi-lateral peers (RFC3765)

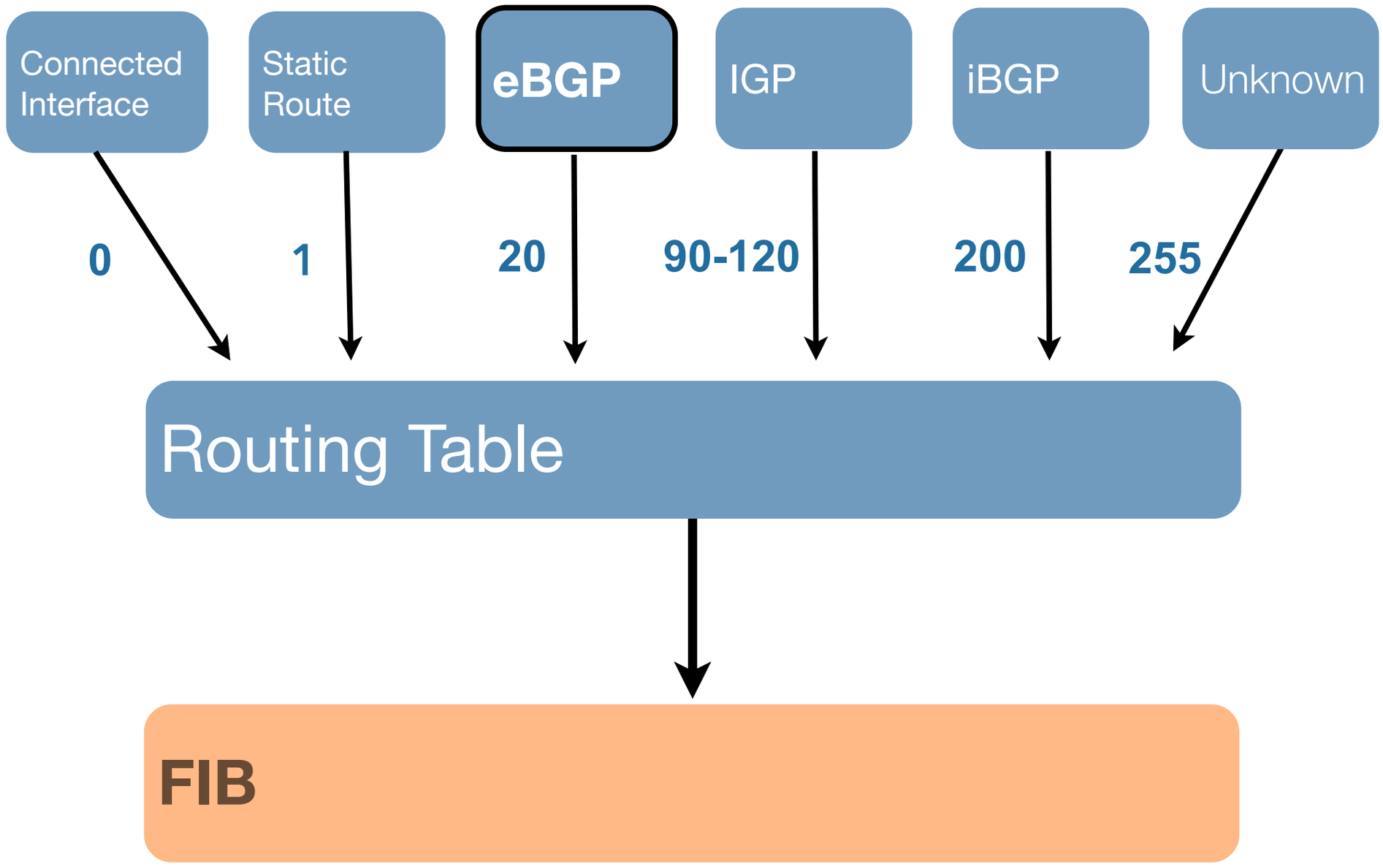
Community Example



RIB and FIB


BGP in
BGP out
best paths
BGP Table
Other Protocols
best paths
Routing Table
CONTROL
best paths
FORWARDING
IP Packets
Forwarding Table
IP Packets

Administrative Distance



BGP Multipath

- The best selection algorithm in BGP selects one route no load balancing from a single router to a single prefix possible
  - Unless “outside” BGP using loopback peering
- BGP Multipath enables load balancing between “equal” paths
  - All attributes must be the same to the level of router ids
  - The next hop router for each multipath must be different

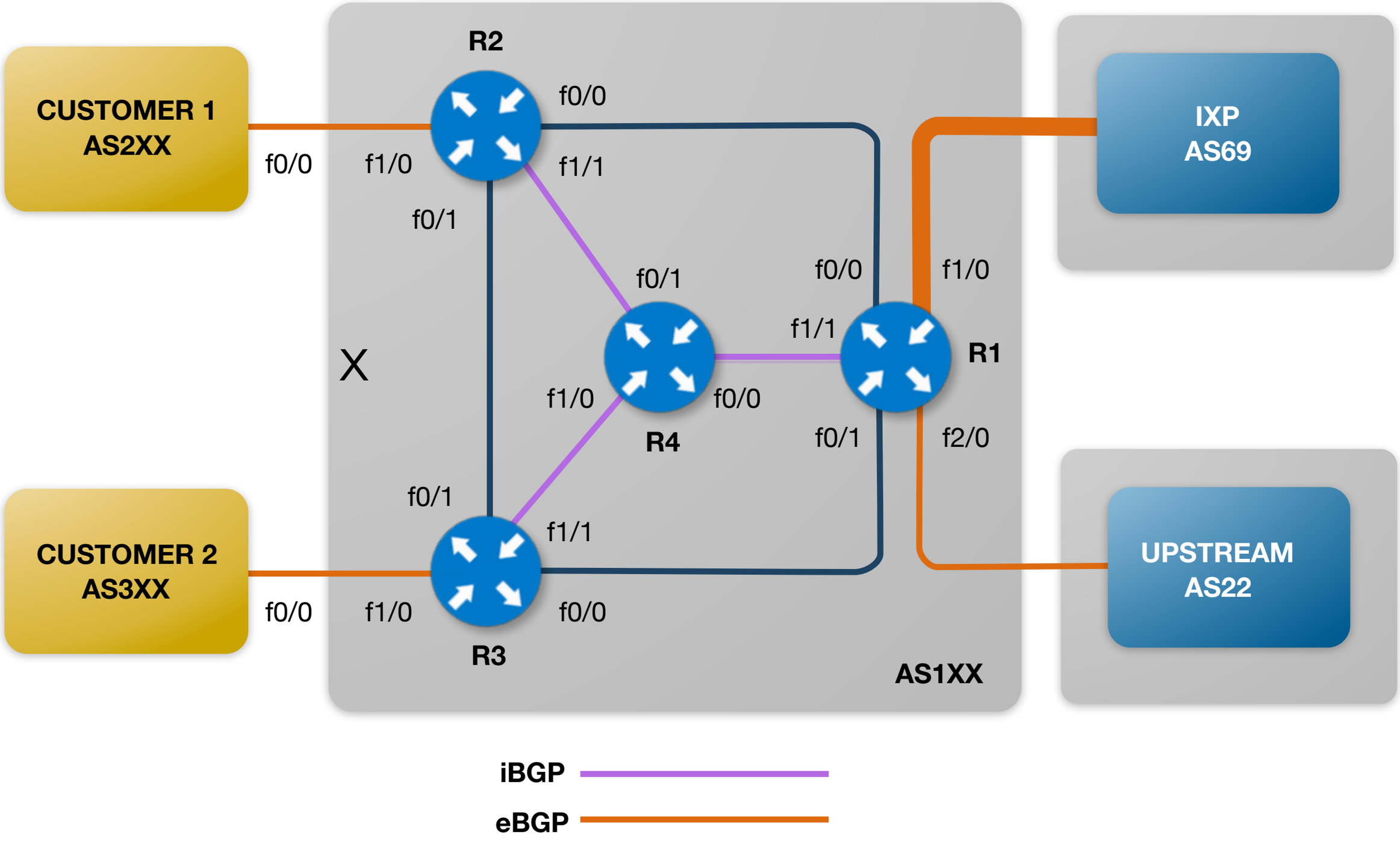
Traffic Engineering and CDNs

- Standard BGP traffic engineering will very often not have the expected results and changes in announcements will have a delayed effect
- Mapping is based on resolving name server
  - Based on location
  - Very often based on other (SDN) metrics
- Not all CDN clusters have a full table
  - selective announcement over multiple upstreams might result in lack of connectivity

Using Attributes

Exercise

Network Diagram



Assignment

- Prefer routes received from Internet Exchange
  - Use local-preference
  - Use AS path prepending
- Data needed
  - The IP address of IXP BGP routers
  - The AS number of IXP
  - Routing policy

Preparation (on R1)

- Examine routing tables

```
# show ip route
```

```
# show ip bgp
```

```
# show ip bgp 10.66.0.1
```

- Which routes are you using to reach other Internet Exchange members?

Outgoing Traffic (on R1)

- Create a route map

```
(config)# route-map local-pref-150 permit 5
(config-route-map)# set local-preference 150
```

- Apply map to incoming routes from IXP

```
(config)# router bgp 1XX
(config-router)# neighbor 172.16.0.66 route-map local-pref-150 in
(config-router)# neighbor 172.16.0.99 route-map local-pref-150 in
```

- Session must be cleared, for the new policy

```
# clear ip bgp 172.16.0.66 soft in
# clear ip bgp 172.16.0.99 soft in
```

Incoming Traffic (on R1)

- Create a route map

```
(config)# route-map PREPEND permit 5
(config-route-map)# match ip address prefix-list transit-out-v4
(config-route-map)# set as-path prepend 1XX 1XX 1XX
```

- Add route map outgoing routes to Transit router

```
(config)# router bgp 1XX
(config-router)# neighbor 10.132.X.1 route-map PREPEND out
```

- Session must be cleared, for the new policy

```
# clear ip bgp 10.132.X.1 soft out
```

Verification (on R1)

- Examine routing tables

```
# show ip route
```

```
# show ip bgp
```

```
# show ip bgp 10.66.0.1
```

- Make sure that routes received from Internet Exchange are preferred
- Ask your colleague to show route to your network

BGP Scalability

Section 5

Networks Grow

- 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?

Scaling Techniques

- Current best practice:
  - Route Refresh capability
  - Peer-groups
  - Route Reflectors
  - Confederations
- Deprecated practice:
  - Soft Reconfiguration
  - Route Flap Damping

Dynamic Reconfiguration

- Routing Policy change:
  - 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
  - Consider the impact to be equivalent to a router reboot

Route Refresh

- Facilitates non-disruptive policy changes
- No configuration is needed
  - Automatically negotiated at peer establishment
  - Requires peering routers to support “route refresh capability” (RFC2918)
- No additional resources used

Network Effect

- iBGP needs full mesh
  - Too many sessions
  - Slow to build
  - iBGP neighbours receive the same update
  - Router CPU wasted on repeat calculations
- How scalable it is n(n-1)/2 ?
  - 2 speakers: 1 peer
  - 5 speakers: 10 peers
  - 14 speakers: 91 peers

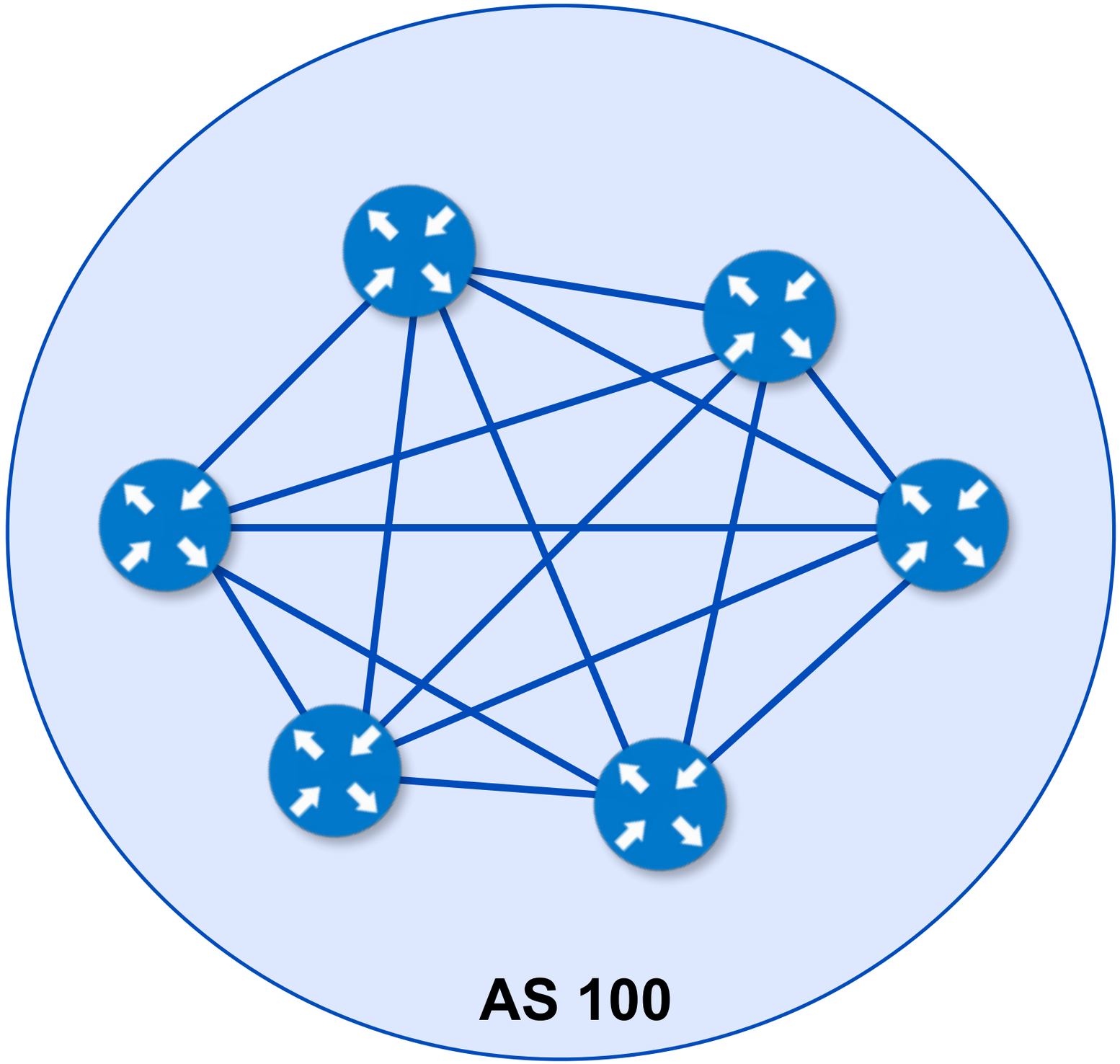
Peer Groups

- Makes configuration easier and more readable
  - Group peers with the same outbound policy
  - Updates are generated once per group (Lower router CPU)
  - iBGP mesh builds more quickly
  - Members can have different inbound policy
- Can be used for eBGP neighbours
  - Consider using peer-groups when policy is generally the same to each peer (ie IXP)

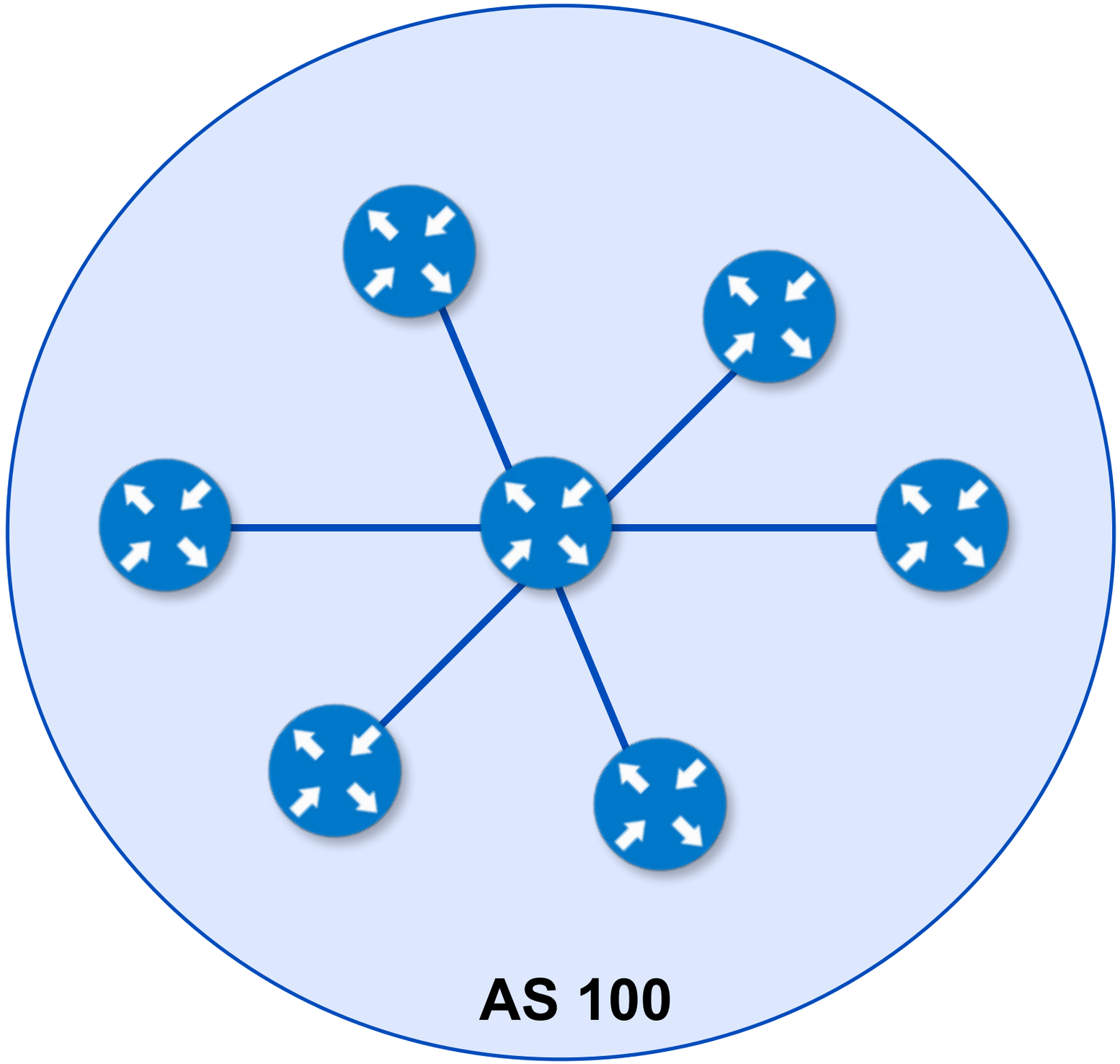
Router Reflectors

- Solves iBGP mesh problem
  - Easy migration
- Packet forwarding is not affected
- Route reflector client is a iBGP peer
  - No special configuration needed
- Redundancy
  - Multiple reflectors for redundancy
  - Multiple levels of route reflectors

iBGP without Route Reflector



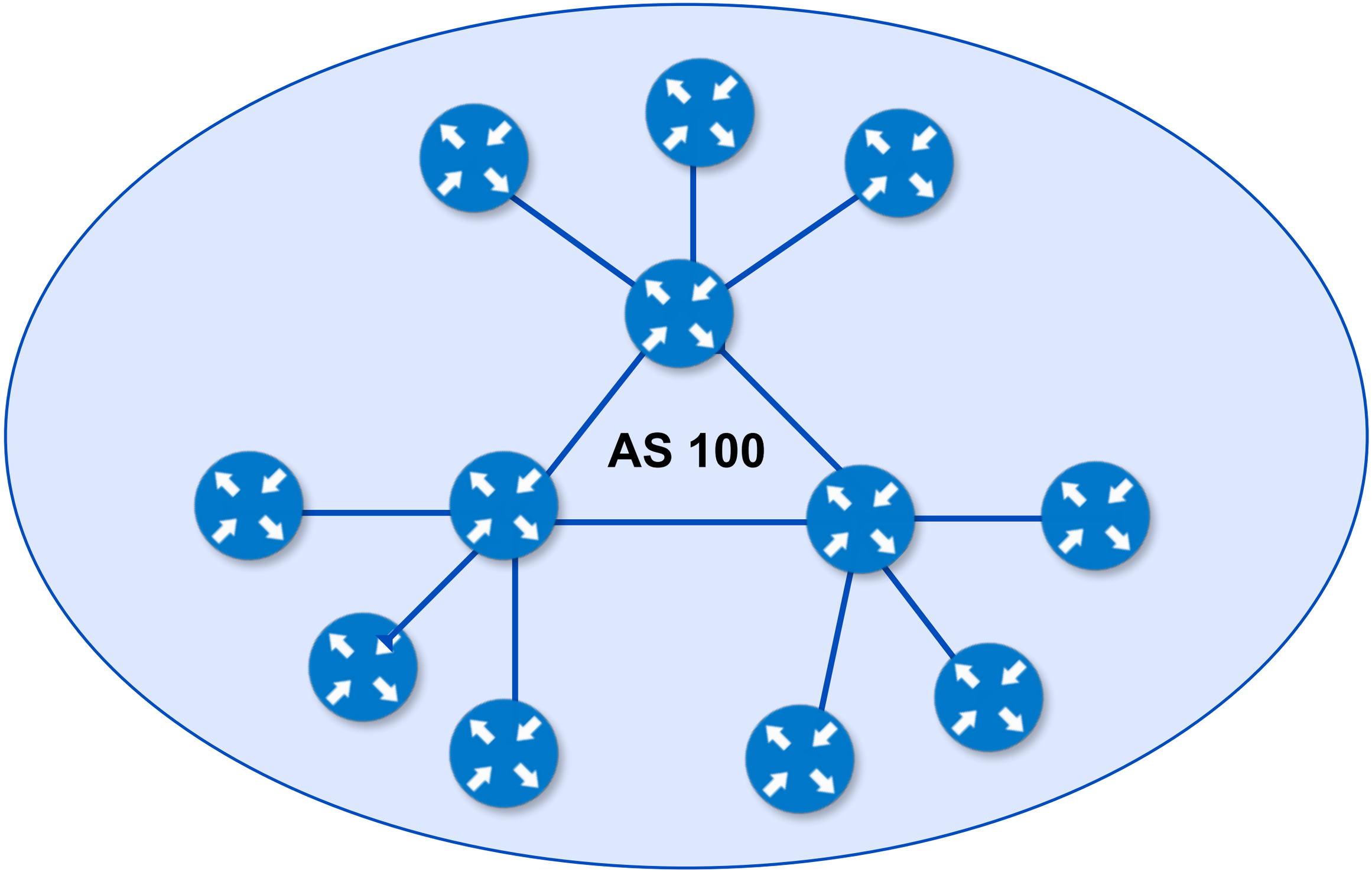
iBGP with Route Reflector



Route Reflector Operations

- Reflector receives path from 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 RFC4456

Route Reflectors Topology



Route Reflector Best Practice

- 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
- IGP to carry next hop and local routes

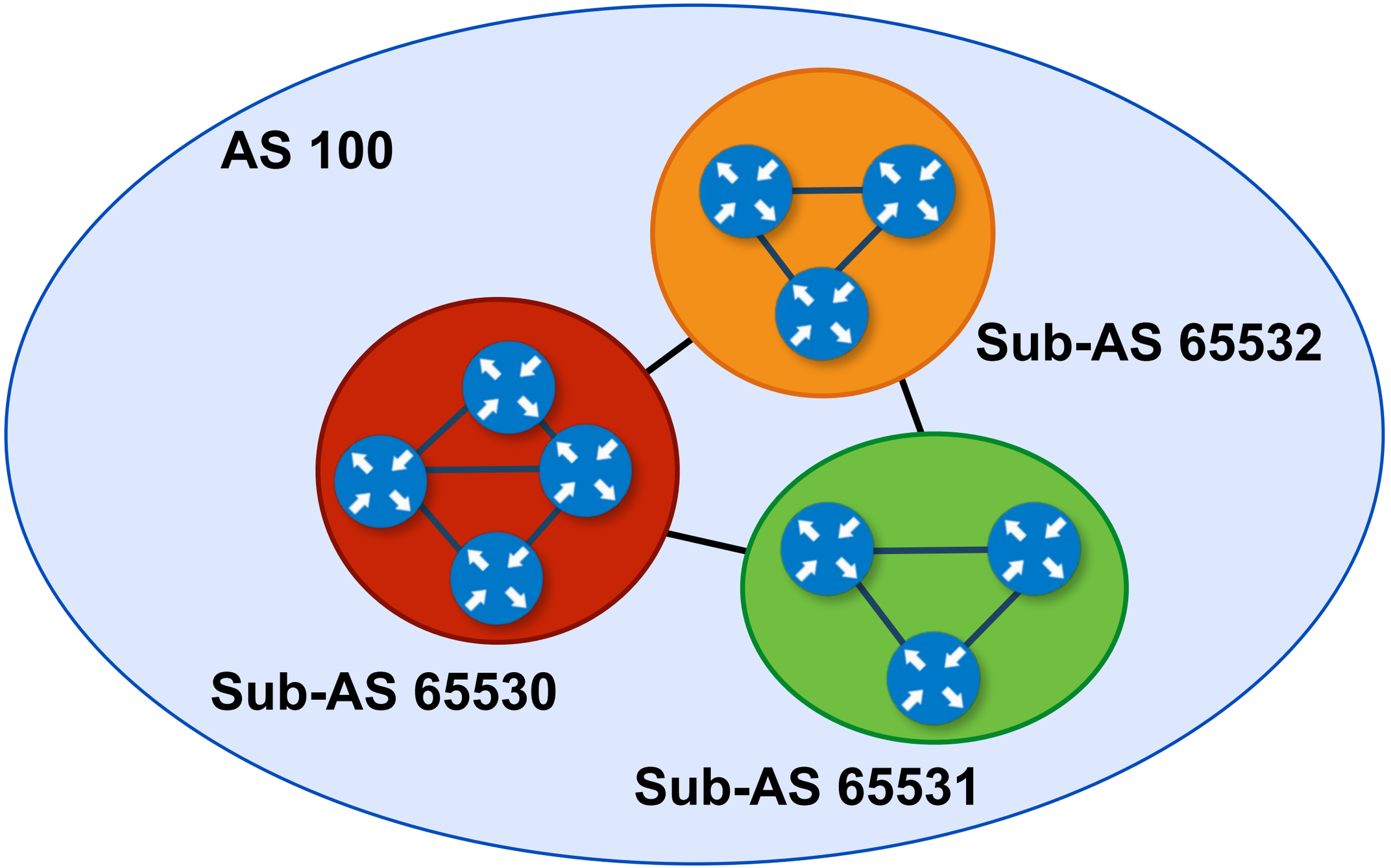
Confederations

- Divide the AS into sub-ASes which are visible to outside world as single AS – “Confederation Identifier”
  - sub-AS use the private space (64512-65534)
  - eBGP between sub-AS, 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 RFC5065

Confederations

- iBGP speakers in 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
- Route propagation
  - 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

Confederations



Confederations and RRs

- The goal is to make it so that your network scales

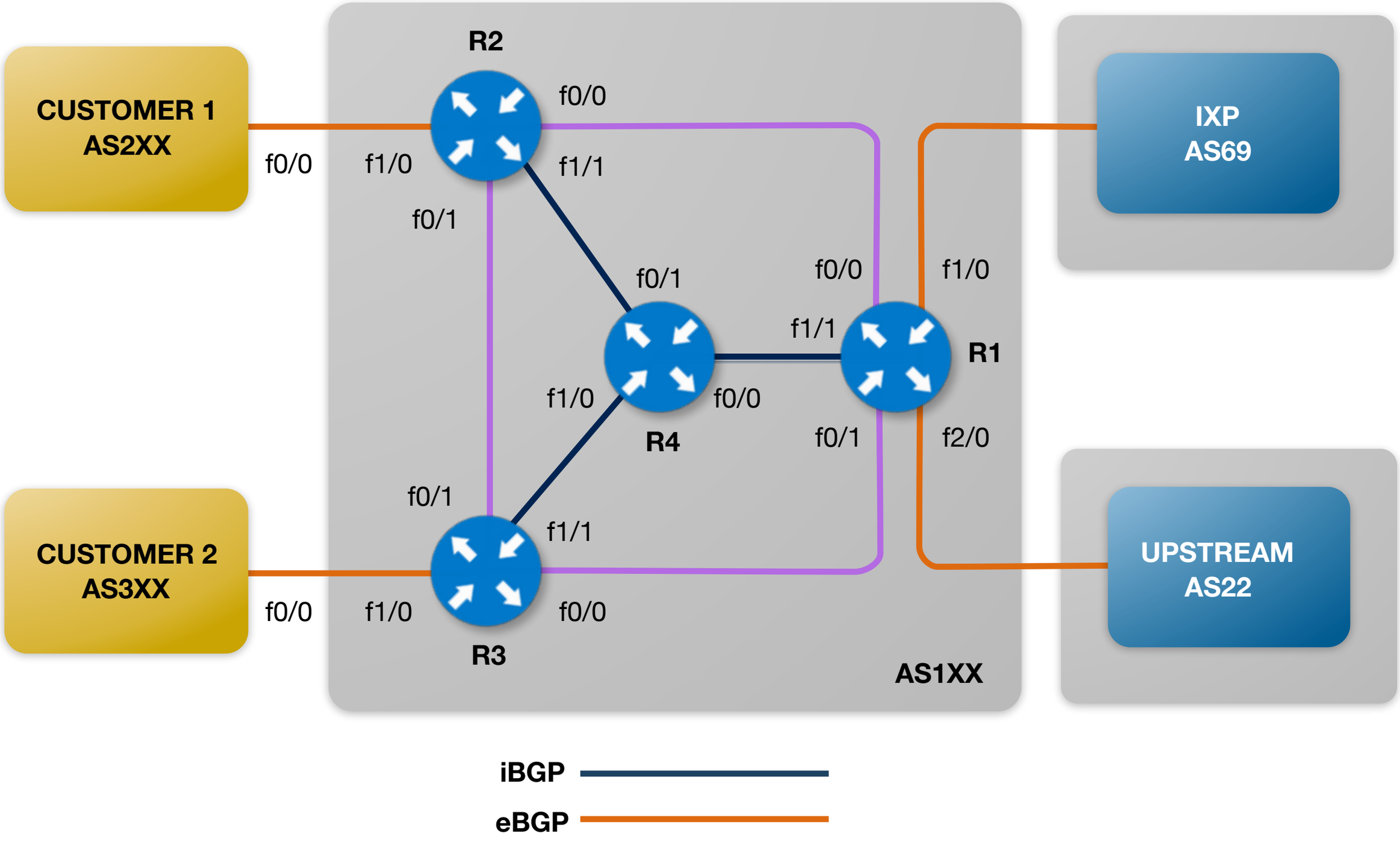
- BGP configuration is not easier with them
  - Only routing management is

- In some cases, requesting a different ASN for a backbone and an access network is possible

Using a Route Reflector

Exercise

Network Diagram



Assignment

- Simplify your internal BGP network by using Router 4 as a Route Reflector

- Data needed
  - Your AS number
  - Your IP address space
  - Loopback address of the Route Reflector

Configure Route Reflector (on R4)

- Router 4 will reflect routes to other iBGP speakers - Router Reflector Clients

```
(config)# router bgp 1XX
(config-router)# bgp log-neighbor-changes

(config-router)# neighbor RR-GROUP peer-group
(config-router)# neighbor RR-GROUP remote-as 1XX
(config-router)# neighbor RR-GROUP update-source lo0
(config-router)# neighbor RR-GROUP route-reflector-client

(config-router)# neighbor 172.X.255.1 peer-group RR-GROUP
(config-router)# neighbor 172.X.255.2 peer-group RR-GROUP
(config-router)# neighbor 172.X.255.3 peer-group RR-GROUP
```

- Configuration simplified with peer-group

Remove iBGP Mesh

- Router 1

```
(config)# router bgp 1XX
(config-router)# no neighbor 172.X.255.2
(config-router)# no neighbor 172.X.255.3
```

- Router 2

```
(config)# router bgp 1XX
(config-router)# no neighbor 172.X.255.1
(config-router)# no neighbor 172.X.255.3
```

- Router 3

```
(config)# router bgp 1XX
(config-router)# no neighbor 172.X.255.1
(config-router)# no neighbor 172.X.255.2
```

Add Route Reflector Clients

- Router 1, Router 2, Router3

```
(config-router)# neighbor 172.X.255.4 remote-as 1XX
(config-router)# neighbor 172.X.255.4 next-hop-self
(config-router)# neighbor 172.X.255.4 update-source lo0
```

Verify

- Check sessions in summary

```
# show ip bgp neighbors | include BGP
```

- Check BGP and routing table

```
# show ip bgp
# show ip route
```

- Verify reachability from customer

```
# ping 10.132.32.1
# ping <your colleague Customer 1 or 2 IP>
```

- Show logged events

```
# show logging
```

Multiprotocol BGP

Section 6

Multiprotocol BGP (MP-BGP)

- Extension to the BGP protocol

- MP-BGP two type protocol:
  - Carrier protocol
  - Passenger protocol

- Negotiated at sessions set up (BGP OPEN message) when CAPABILITIES contain Multiprotocol Extensions

MP-BGP

- New BGP features in OPEN message:
  - BGP Capabilities Advertisement:
  - Address Family Identifier (**AFI**)
  - Subsequent Address Family Identifier (**SAFI**)
  - Multiprotocol Reachable Network Layer Reachability Information (**MP_UNREACH_NLRI and MP_REACH_NLRI**)

AFI / SAFI

- Address Family Identifier (AFI)
  - Identifies Address Type
  - AFI = 1 (IPv4)
  - AFI = 2 (IPv6)
- Subsequent Address Family Identifier (SAFI)
  - Sub category for AFI Field
  - Address Family Identifier (AFI)
    - Sub-AFI = 1 (NLRI is used for unicast)
    - Sub-AFI = 2 (NLRI is used for multicast RPF check)
    - Sub-AFI = 3 (NLRI is used for both unicast and multicast RPF check)
    - Sub-AFI = 4 (label)
    - Sub-AFI = 128 (VPN)

Multiprotocol BGP

Exercise

Assignment

- Enable Multiprotocol BGP
- Using IPv6
  - Connect your network to Transit Provider
  - Connect you network to Internet Exchange
- Data needed
  - Your AS number
  - Your IPv6 address space
  - The AS number of your neighbors
  - The IPv6 address of your neighbors BGP routers

Preparation (on R1)

- Insert static Null route
  - Before BGP advertised its network, it checks for an exact match of network number and mask on router's routing table

```
(config)# ipv6 route 2001:ffXX::/32 null0 250
```

Enable Multiprotocol BGP (on R1)

- Enable MP-BGP

```
(config)# router bgp 1XX
(config-router)# no bgp default ipv4-unicast
```

- Examine your router BGP configuration

```
# show running-config | section router bgp
```

Interface IPv6 Settings (on R1)

- Your network is already dual stacked
  - IGP and Loopbacks
- Configure IPv6 on your IXP interface

```
(config)# interface FastEthernet1/0
(config-if)# ipv6 address 2001:ff69::XX/64
(config-if)# no ipv6 redirects
(config-if)# ipv6 nd ra suppress all
```

- Configure IPv6 on your Transit interface

```
(config)# interface FastEthernet2/0
(config-if)# ipv6 address 2001:ff32:0:XX::b/64
(config-if)# no ipv6 redirects
(config-if)# ipv6 nd ra suppress all
```

Create a filter (on R1)

- BGP sends the best paths to all neighbours

```
(config)# ipv6 prefix-list transit-out-v6 seq 5 permit
2001:ffXX::/32
(config)# ipv6 prefix-list ixp-out-v6 seq 5 permit
2001:ffXX::/32
```

Configure Transit Session (on R1)

- Configure BGP session with AS22

```
(config)# router bgp 1XX
(config-router)# neighbor 2001:ff32:0:XX::a remote-as 22
(config-router)# address-family ipv6
(config-router-af)# neighbor 2001:ff32:0:XX::a activate
(config-router-af)# neighbor 2001:ff32:0:XX::a prefix-
list transit-out-v6 out
```

- Advertise route

```
(config-router-af)# network 2001:ffXX::/32
```

Configure IXP Sessions (on R1)

- Configure BGP sessions with AS69

```
(config)# router bgp 1XX
(config-router)# neighbor 2001:ff69::66 remote-as 69
(config-router)# address-family ipv6
(config-router-af)# neighbor 2001:ff69::66 activate
(config-router-af)# neighbor 2001:ff69::66 prefix-list
ixp-out-v6 out
(config-router)# exit
(config-router)# neighbor 2001:ff69::99 remote-as 69
(config-router)# address-family ipv6
(config-router-af)# neighbor 2001:ff69::99 activate
(config-router-af)# neighbor 2001:ff69::99 prefix-list
ixp-out-v6 out
```

Verify

- Check sessions summary

```
# show bgp ipv6 unicast summary
```

- Check BGP and routing table

```
# show bgp ipv6
# show ipv6 route
```

- Verify reachability

```
# ping 2001:ff32::a
# ping <your colleague R1 IPv6>
```

- Show logged events

```
# show logging
```

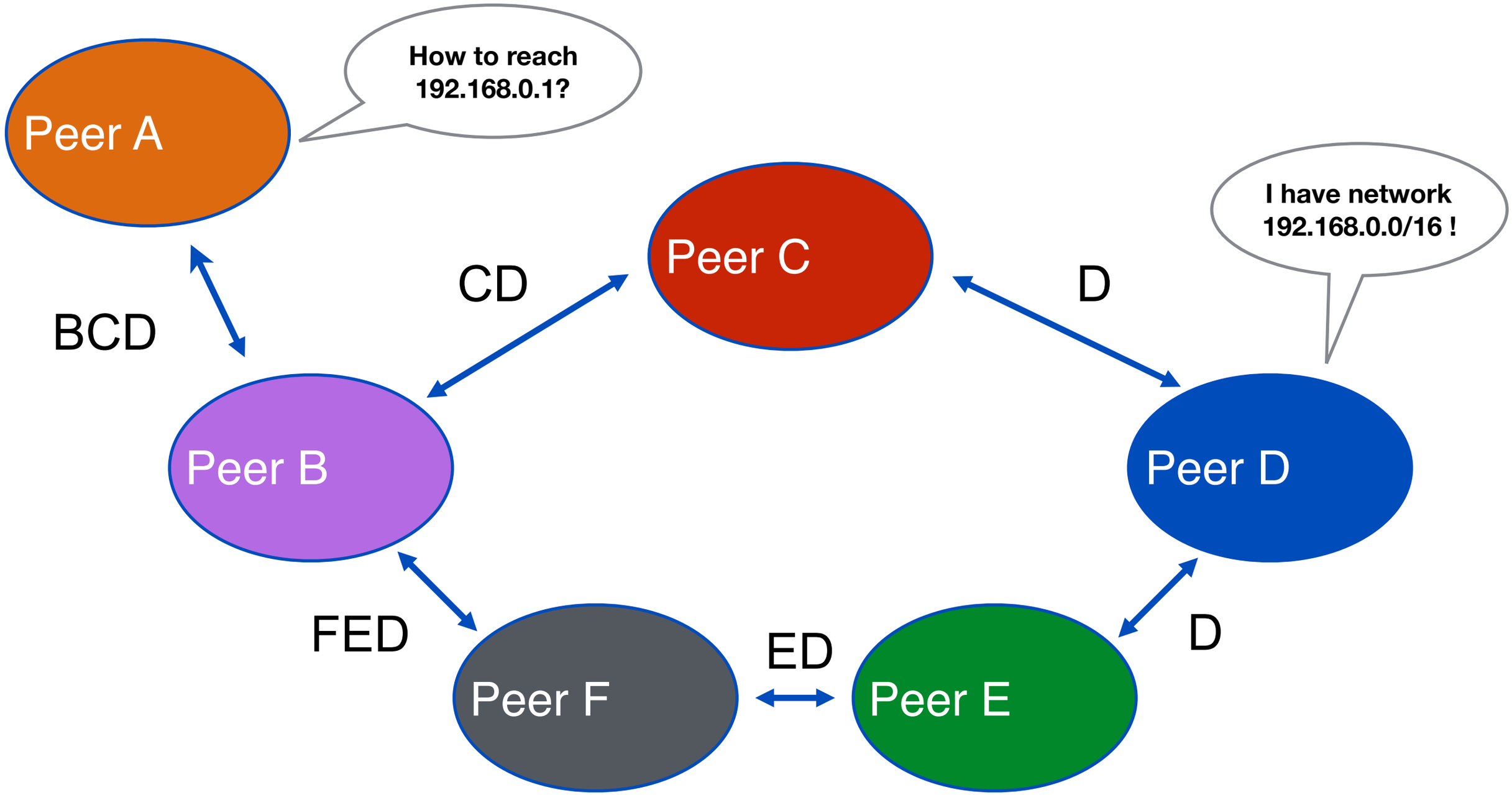
Routing Security

Section 7

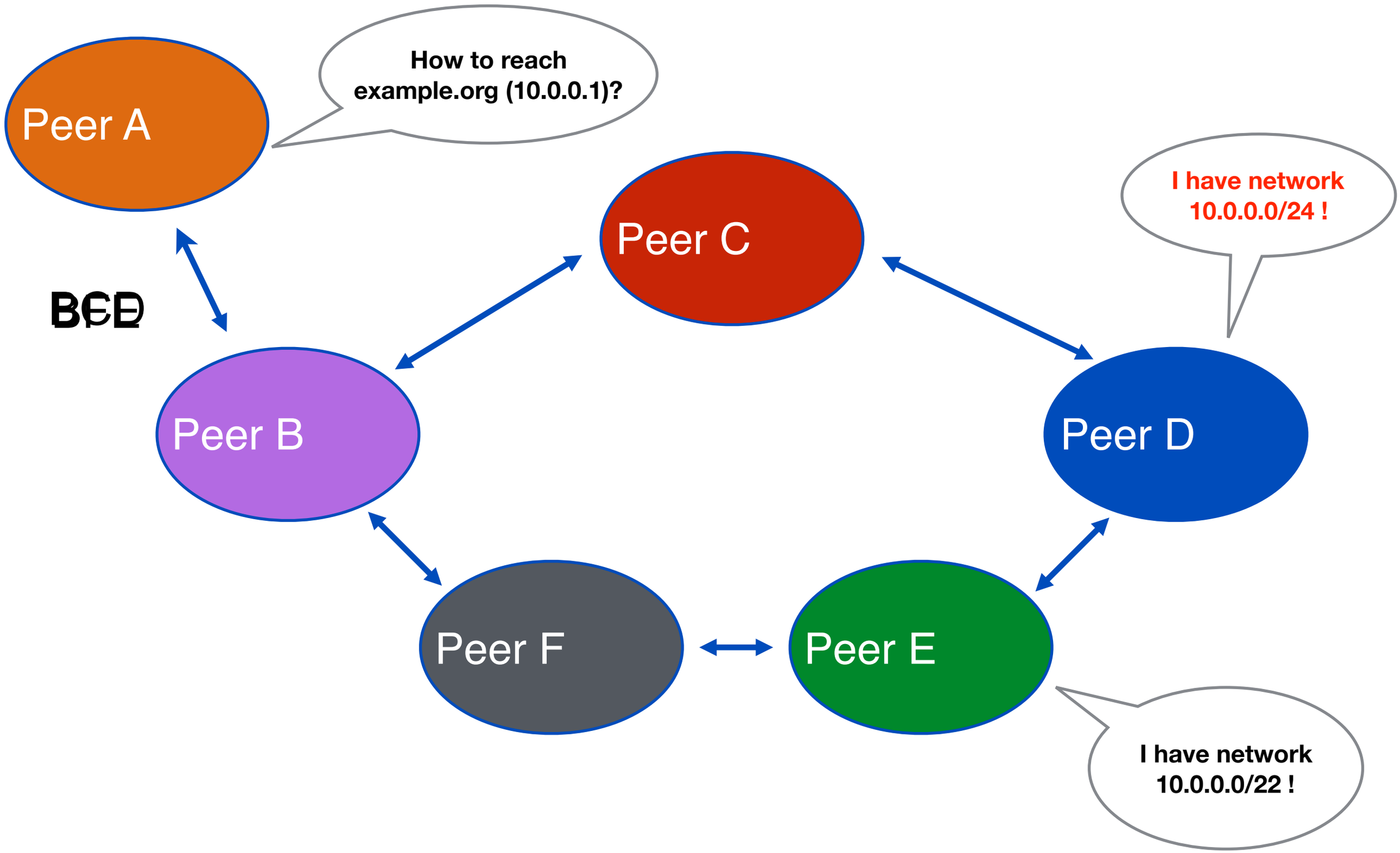
Threats to Routing

- BGP is not secure by default
- Cryptography (TLS/IPsec) can mitigate effects, but not stop them
- BGP security can be achieved using:
  - Filters
  - RPKI
  - BGPSEC

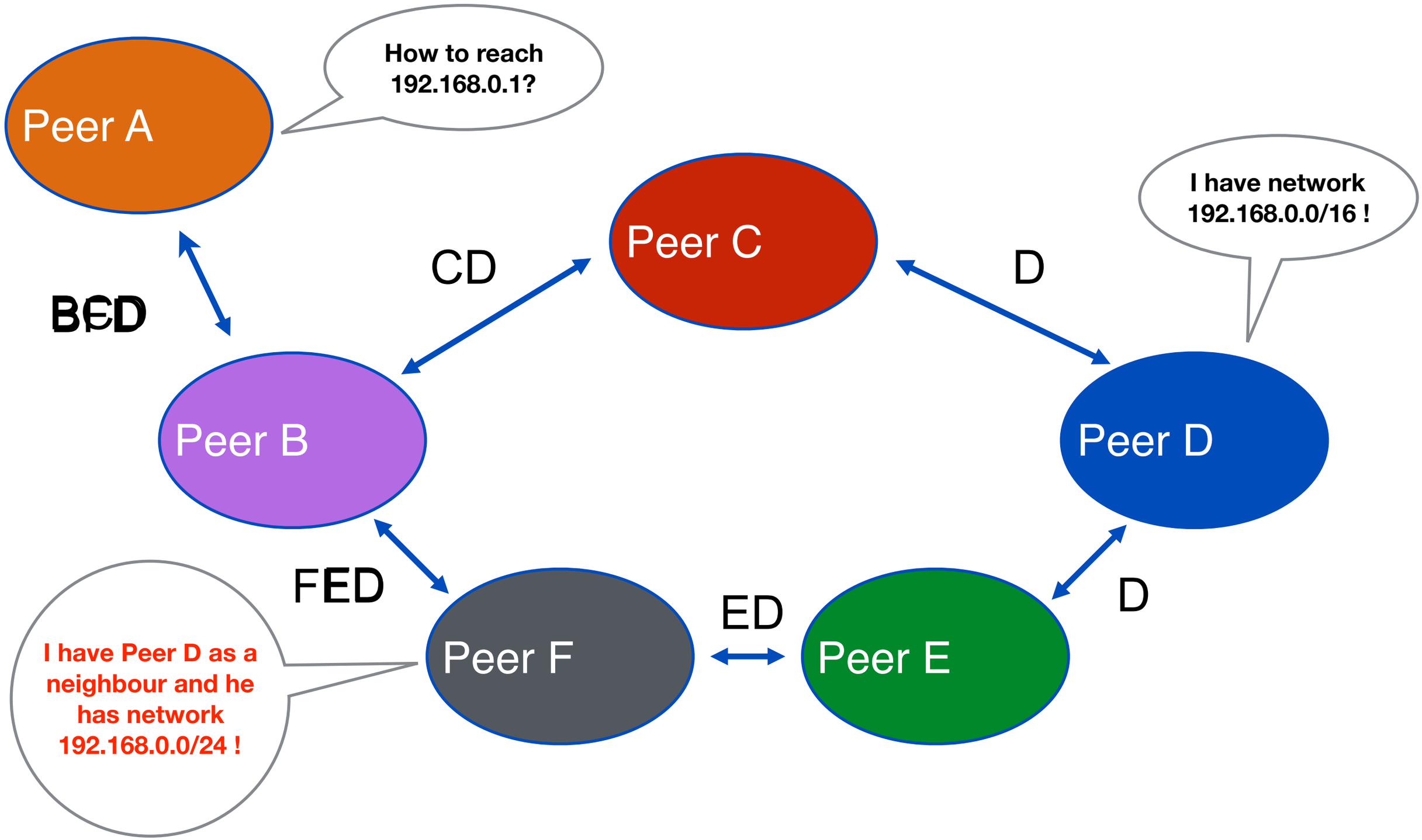
BGP Path and Origin



False Origin

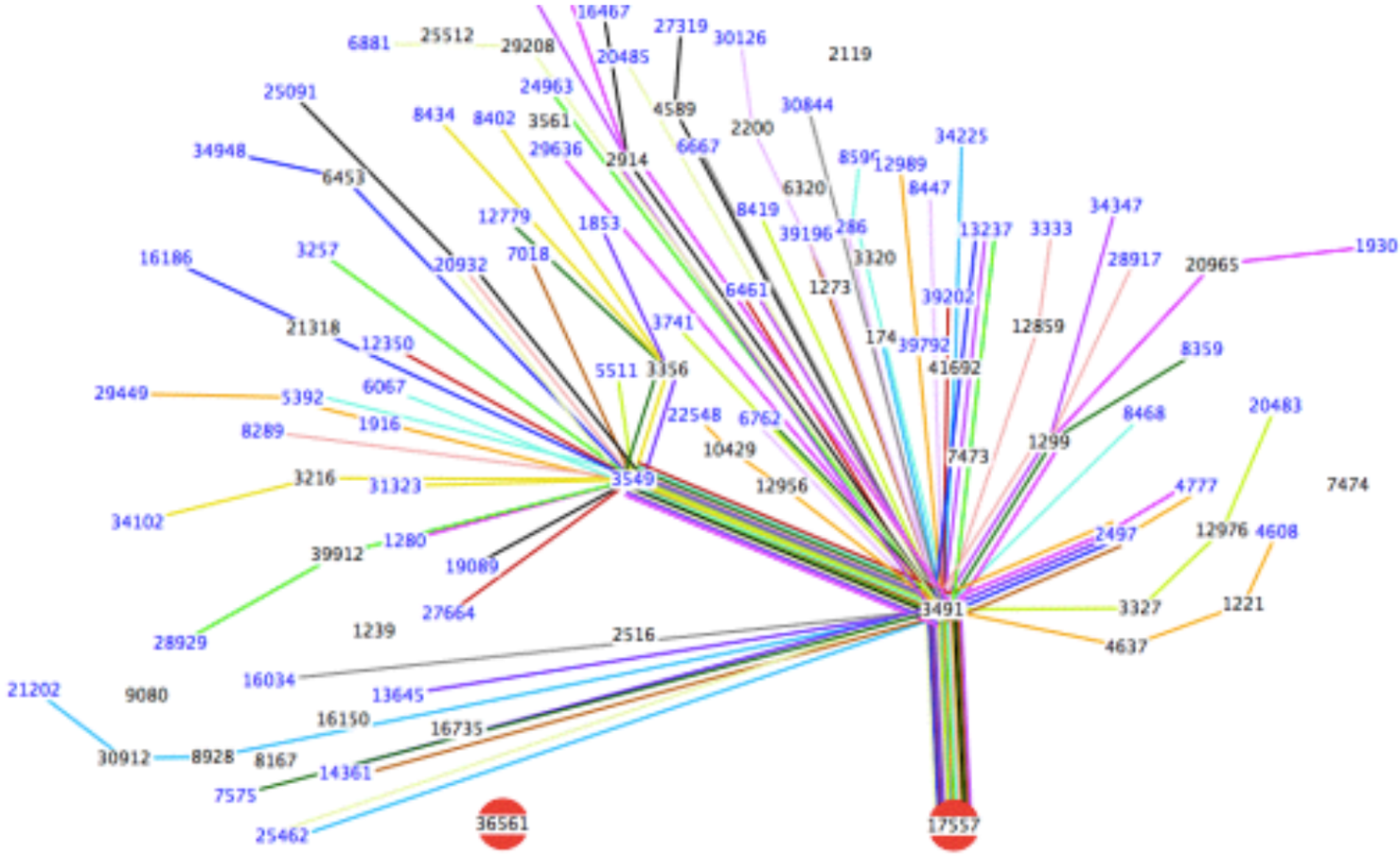


False Path


How to reach 192.168.0.1?
Peer A
BED
CD
Peer C
D
I have network 192.168.0.0/16 !
Peer B
Peer D
FED
ED
D
I have Peer D as a neighbour and he has network 192.168.0.0/24 !
Peer F
Peer E

Routing Incidents Types

- Misconfiguration
  - No malicious intention
  - Software bugs
- Malicious
  - Competition
  - Claiming “unused” space
- Targeted Traffic Misdirection
  - Collect and/or tamper with data



Filtering

Section 8

Filtering Principles

- Filter as close to the edge as possible
- Filter as precisely as possible
- Filter both source and destination where possible

- Two filtering techniques:
  - Explicit Permit (permit then deny any)
  - Explicit Deny (deny then permit any)

Bogons

- Routes you shouldn't see in the routing table
  - Private addresses
  - Non-allocated space
  - Reserved space (Future use, Multicast, etc.)
- You should have filters applied so that these routes are not advertised to or propagated through the Internet
- Team Cymru provides list or BGP feed
  - http://www.team-cymru.org/bogon-reference-bgp.html

Prefix-lists

- Prefix lists are lists of routes you want to accept or announce
- Easy to use but not highly scalable
- You can create them manually or automatically
  - With data from RIPE DB or other Internet Routing Registry
- Or using a tool
  - Level3 Filtergen
  - bgpq3
  - IRRexplorer

Filtering AS Path

- Filter routes based on AS path
- Widely used and highly scalable
- Applied same way as prefix-list filters

```
router bgp 65564
  network 10.0.0.0 mask 255.255.255.0
  neighbor 172.16.1.1 remote-as 65563
  neighbor 172.16.1.1 filter-list 1 out
  neighbor 172.16.1.1 filter-list 2 in

ip as-path access-list 1 permit ^65564$
ip as-path access-list 2 permit ^65563$
```

Regular Expressions

- Most router OS uses 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

Filtering AS-PATH Example

- You can use regular expression to match AS

_100_	Via AS100
_(100_)+	Multiple AS100 (prepending)
^100$	Connected to AS100
_100$	Originated by AS100
^100_	Received from AS100
^[0-9]+$	AS-PATH of single AS
^$	Local AS prefixes
.*	Any AS-PATH

Reverse Path Forwarding

- Called uRPF (Unicast Reverse Path Forwarding)
- Checks if an entry exists in the routing table before accepting the packet and forwarding it
- Two modes
  - Loose
  - Strict

Strict and Loose RPF

- Strict
  - Checks if the entry is in the routing table
  - and the route points to the receiving interface
- Loose
  - Simply checks that an entry exists for the route in the routing table

Best Current Practice 38

- Defines some steps to take in order to have a “cleaner” routing table
- Restricting forged traffic (TCP and UDP)
- Implies the use of:
  - Prefix filters
  - Bogon filters
  - uRPF
- http://tools.ietf.org/html/bcp38

Ingress filters

- Best Practices:
  - Don’t accept RFC1918 etc prefixes
  - Don’t accept your own prefix
  - Don’t accept default (unless you requested it)
  - Don’t accept IPv4 prefixes longer than /24
  - Don’t accept IPv6 prefixes longer than /48
  - Consider Net Police Filtering

BGP ASN Bogons

- 0
  - Reserved RFC7607
- 23456
  - AS_TRANS RFC6793
- 64496-64511 and 65536-65551
  - Reserved for use in docs and code RFC5398
- 64512-65534 and 4200000000-4294967294
  - Reserved for Private Use RFC6996
- 65535 and 4294967295
  - Reserved RFC7300
- 65552-131071
  - Reserved

Defining Filters

Exercise

Preparation (on R1)

- Examine your routing table

```
# show ip route bgp
# show ip bgp
# show ipv6 route bgp
# show bgp ipv6
```

- Do you see any prefix that is too specific?

Filter More Specifics (on R1)

- Filtering of the prefixes that are too specific

```
(config)# ip prefix-list transit-in-v4 seq 10 permit
0.0.0.0/0 le 24
(config)# ip prefix-list ixp-in-v4 seq 10 permit
0.0.0.0/0 le 24
(config)# ipv6 prefix-list transit-in-v6 seq 10 permit
2000::/3 le 48
(config)# ipv6 prefix-list ixp-in-v6 seq 10 permit
2000::/3 le 48
```

Filter More Specifics

- Add incoming policy to the neighbors

```
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# neighbor 10.132.X.1 prefix-list transit-
in-v4 in
(config-router-af)# neighbor 172.16.0.66 prefix-list ixp-in-
v4 in
(config-router-af)# neighbor 172.16.0.99 prefix-list ixp-in-
v4 in
(config-router-af)# address-family ipv6
(config-router-af)# neighbor 2001:ff32:0:XX::a prefix-list
transit-in-v6 in
(config-router-af)# neighbor 2001:ff69::66 prefix-list ixp-
in-v6 in
(config-router-af)# neighbor 2001:ff69::99 prefix-list ixp-
in-v6 in
```

Clear the BGP Sessions (on R1)

```
# clear ip bgp 172.16.0.66 soft in
# clear ip bgp 172.16.0.99 soft in
# clear ip bgp 10.132.X.1 soft in
# clear bgp ipv6 unicast 2001:ff69::66 soft in
# clear bgp ipv6 unicast 2001:ff69::99 soft in
# clear bgp ipv6 unicast 2001:ff32:0:XX::a soft in
```

Verify

- Check BGP and routing table

```
# show ip bgp
# show bgp ipv6 unicast
# show ip route bgp | i /25
# show ipv6 route | include /64
```

Filter Customer 1 on Router 2

- Create prefix-list

```
(config)# ip prefix-list c1-in-v4 seq 5 permit 10.X.1.0/24
```

- Add incoming policy to the neighbor

```
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# neighbor 10.X.0.26 prefix-list c1-in-v4 in
```

Filter Customer 2 on Router 3

- Create prefix-list

```
(config)# ip prefix-list c2-in-v4 seq 5 permit 10.X.2.0/24
```

- Add incoming policy to the neighbor

```
(config)# router bgp 1XX
(config-router)# address-family ipv4
(config-router-af)# neighbor 10.X.0.30 prefix-list c2-in-v4 in
```

IPv4 Reserved Prefix Filtering

- Example list

```
ip prefix list ipv4-list deny 0.0.0.0/8 le 32
ip prefix list ipv4-list deny 10.0.0.0/8 le 32
ip prefix list ipv4-list deny 100.64.0.0/10 le 32
ip prefix list ipv4-list deny 127.0.0.0/8 le 32
ip prefix list ipv4-list deny 169.254.0.0/16 le 32
ip prefix list ipv4-list deny 172.16.0.0/12 le 32
ip prefix list ipv4-list deny 192.0.0.0/24 le 32
ip prefix list ipv4-list deny 192.0.2.0/24 le 32
ip prefix list ipv4-list deny 192.168.0.0/16 le 32
ip prefix list ipv4-list deny 198.18.0.0/15 le 32
ip prefix list ipv4-list deny 198.51.100.0/24 le 32
ip prefix list ipv4-list deny 203.0.113.0/24 le 32
ip prefix list ipv4-list deny 224.0.0.0/4 le 32
ip prefix list ipv4-list deny 240.0.0.0/4 le 32
```

IPv6 Reserved Prefix Filtering

- Example list

```
ipv6 prefix-list ipv6-list deny 3ffe::/16 le 128
ipv6 prefix-list ipv6-list deny 2001:db8::/32 le 128
ipv6 prefix-list ipv6-list permit 2001::/32
ipv6 prefix-list ipv6-list deny 2001::/32 le 128
ipv6 prefix-list ipv6-list permit 2002::/16
ipv6 prefix-list ipv6-list deny 2002::/16 le 128
ipv6 prefix-list ipv6-list deny 0000::/8 le 128
ipv6 prefix-list ipv6-list deny fe00::/9 le 128
ipv6 prefix-list ipv6-list deny ff00::/8 le 128
ipv6 prefix-list ipv6-list permit 2000::/3 le 48
ipv6 prefix-list ipv6-list deny 0::/0 le 128
```

Internet Routing Registry

Section 9

Internet Routing Registry

- Number of public databases that contain routing policy information which mirror each other:
  - RIPE, APNIC, RADB, JPIRR, Level3, …
  - http://www.irr.net
- RIPE NCC operates the RIPE Routing Registry
  - Part of the RIPE Database
  - Part of the Internet Routing Registry

RIPE Database Objects

- inetnum ➡ IPv4 address range
- inet6num ➡ IPv6 address range
- **aut-num** ➡ single AS number and routing policy
- **route, route6** ➡ glue between IP address range and an AS number announcing it

- person ➡ contact info for other objects
- role ➡ group of person objects
- maintainer ➡ protects all other objects

Registering IPv4 Routes

inetnum:	192.168.0.0/24
tech-c:	LA789-RIPE
admin-c:	JD1-RIPE
mnt-by:	RIPE-NCC-HM-MNT
mnt-routes:	LIR-MNT

aut-num:	AS64512
as-name:	GREEN-AS
tech-c:	LA789-RIPE
admin-c:	JD1-RIPE
mnt-by:	LIR-MNT

route:	192.168.0.0/24
tech-c:	LA789-RIPE
admin-c:	JD1-RIPE
origin:	AS64512
mnt-by:	LIR-MNT

Registering IPv6 Routes

inet6num:	2001:db8::/32
tech-c:	LA789-RIPE
admin-c:	JD1-RIPE
mnt-by:	RIPE-NCC-HM-MNT
mnt-routes:	LIR-MNT

aut-num:	AS64512
as-name:	GREEN-AS
tech-c:	LA789-RIPE
admin-c:	JD1-RIPE
mnt-by:	LIR-MNT

route6:	2001:db8::/32
tech-c:	LA789-RIPE
admin-c:	JD1-RIPE
origin:	AS64512
mnt-by:	LIR-MNT

aut-num Object and Routing Policy

aut-num:	AS64512
descr:	RIPE NCC Training Services
as-name:	GREEN-AS
tech-c:	LA789-RIPE
admin-c:	JD1-RIPE
import:	from AS64444 accept ANY
import:	from AS64488 accept ANY
export:	to AS64444 announce AS64512
export:	to AS64488 announce AS64512
mnt-by:	LIR-MNT
source:	RIPE

Why Publish Your Routing Policy?

- Some transit providers and IXPs (Internet Exchange Points) require it
  - They build their filters based on the Routing Registry
- Contributes to routing security and stability
  - Let people know about your intentions
- Can help in troubleshooting
  - Which parties are involved?

RIPE Database

- Close relation between registry information and routing policy
  - The holder of the resources knows how they should be routed

- The Routing Policy Specification Language (RPSL) originates from a RIPE Document
  - Shares attributes with the RIPE Database

RPSL

- Routing Policy Specification Language
- Language used by the IRRs
- Not vendor-specific
- Documented in RFC 2622
  - and RFC 2650 “Using RPSL in practice”
- Can be translated into router configuration

Objects Involved

- **route** or **route6** object
  - Connects a prefix to an origin AS
- **aut-num** object
  - Registration record of an AS
  - Contains the routing policy
- Sets
  - Objects can be grouped in sets, i.e. as-set, route-set
- Keywords
  - “ANY” matches every route

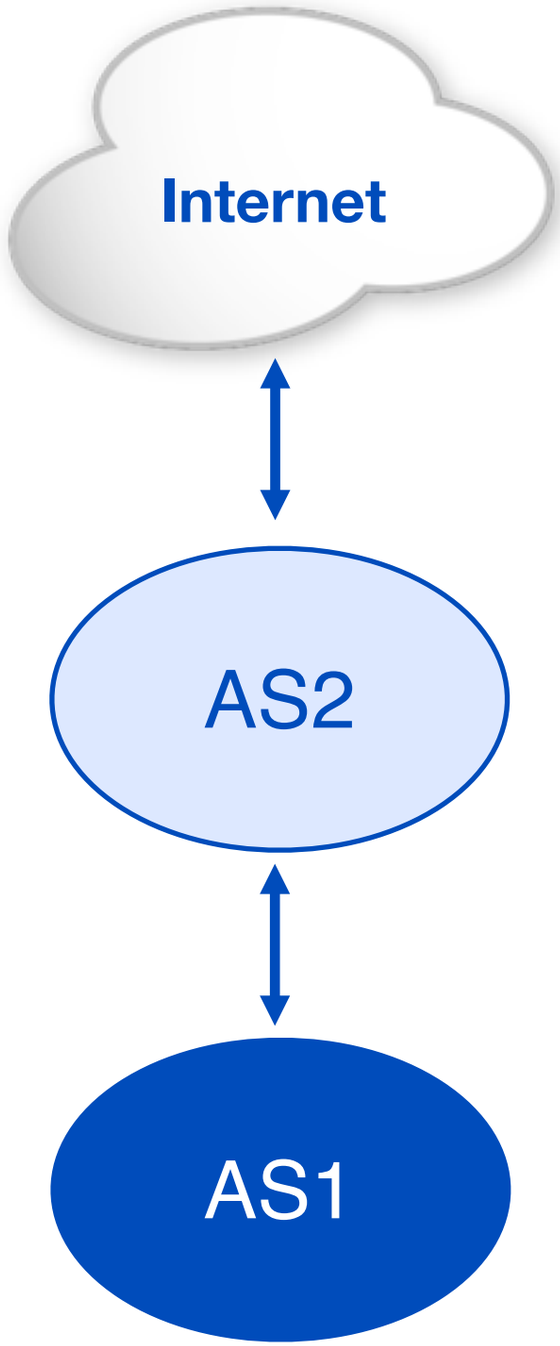
Notation

- AS Numbers are written as ASxxx
- Prefixes are written in CIDR notation
  - i.e.193.0.4.0/24
- Any value can be replaced by a list of values of the same type
  - AS1 can be replaced by “AS1 AS2 AS3”
- You can reference a set instead of a value
  - “...announce AS1” or “...announce as-myname”

Import and Export Attributes

- You can document your routing policy in your **aut-num** object in the RIPE Database:
  - Import lines describe what routes you accept from a neighbour and what you do with them
  - Export lines describe which routes you announce to your neighbour

Example: You Are Customer

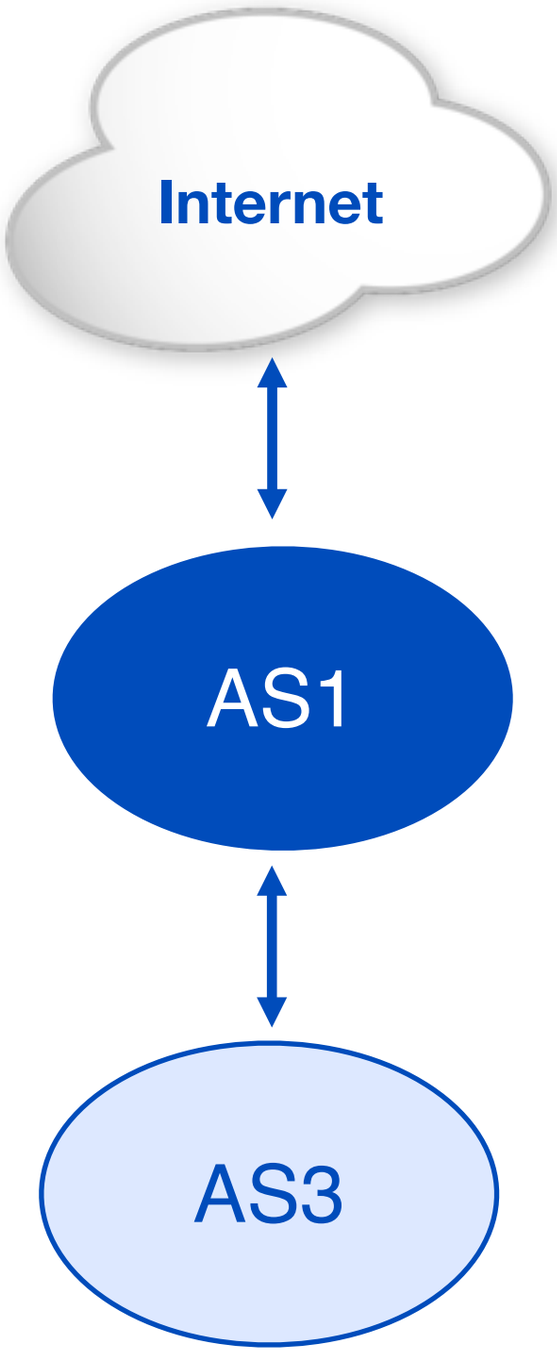


Transit provider

You

```
aut-num:  AS1
import: from AS2 accept ANY
export: to AS2 announce AS1
```

Example: You Are Transit

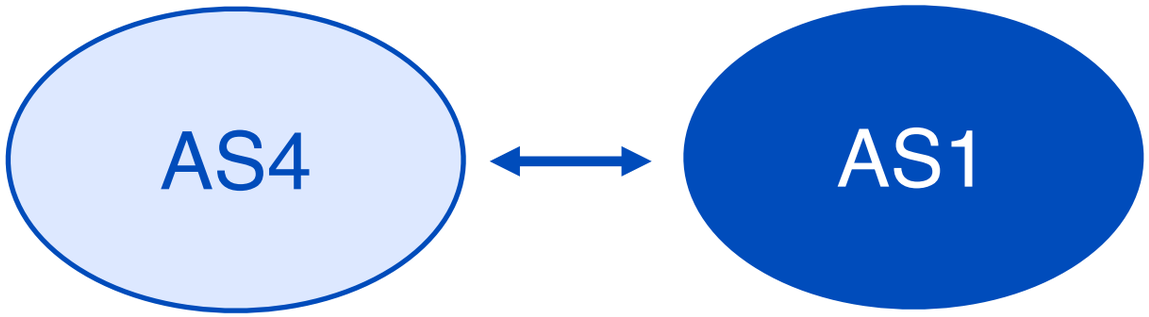


You

```
aut-num:  AS1
import: from AS3 accept AS3
export: to AS3 announce ANY
```

Downstream customer

Example: Peering

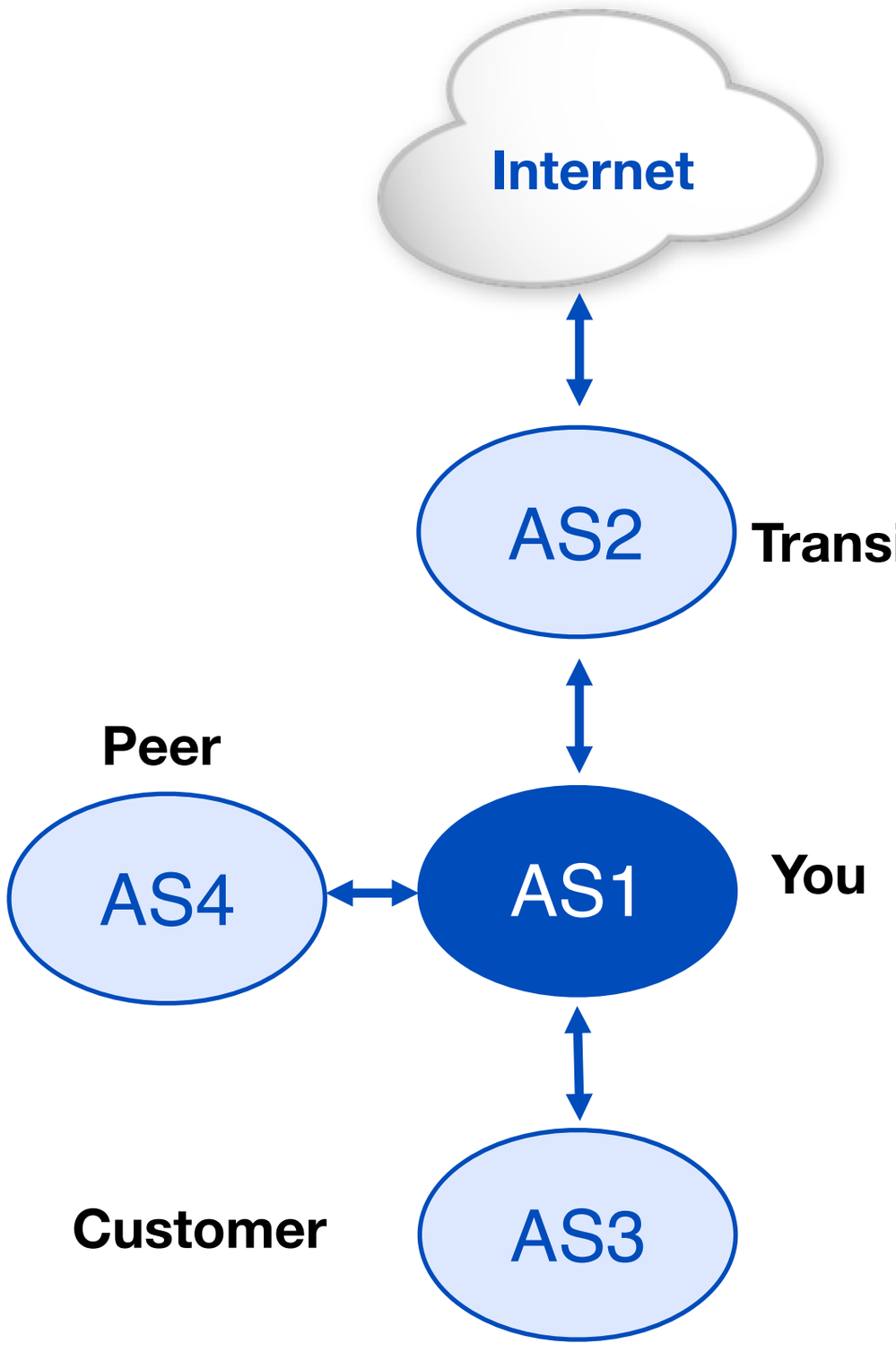


Peer

You

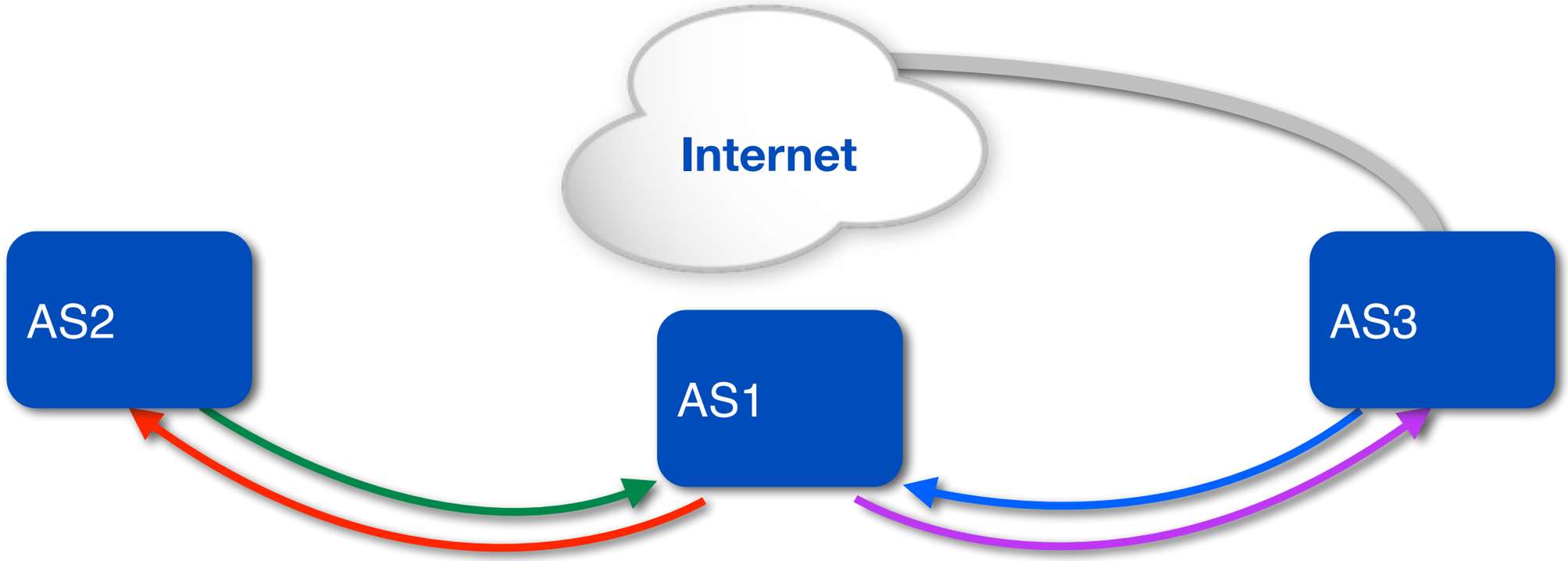

```
aut-num:  AS1
import: from AS4 accept AS4
export: to AS4 announce AS1
```

Example: Summary




```
aut-num:  AS1
import: from AS2 accept ANY
export: to AS2 announce AS1 AS3
import: from AS3 accept AS3
export: to AS3 announce ANY
import: from AS4 accept AS4
export: to AS4 announce AS1 AS3
```

Building an aut-num Object



aut-num: AS2

import: from AS1 accept AS1

export: to AS1 announce AS2

aut-num: AS1

export: to AS2 announce AS1

import: from AS2 accept AS2

import: from AS3 accept ANY

export: to AS3 announce AS1

aut-num: AS3

export: to AS1 announce ANY

import: from AS1 accept AS1

RPSLng

- RPSL is older than IPv6, the defaults are IPv4
- IPv6 was added later using a different syntax
- You have to specify that it's IPv6

```
mp-import:  afi ipv6.unicast from AS201 accept AS201
mp-export:  afi ipv6.unicast to AS201 announce ANY
```

- More information in RFC4012 RPSLng

Routing Registries Challenges

- Accuracy and completeness
- Not every Routing Registry is linked directly to an Internet Registry
  - Offline verification of the resource holder is needed
- Different authorisation methods
- Mirrors are not always up to date

Describing Your Routing Policy

Exercise

Assignment

- Create **route** and **route6** objects for your announcements
- Describe your Routing Policy in aut-num
- Data needed
  - Your AS number
  - The AS number of your neighbors
  - The IPv6 address of your neighbors BGP routers

Preparation

- Create RIPE Access account

- Using your number on the participants list, identify your IPv4 and IPv6 allocations in RIPE TEST Database
- Find out your AS Number using the same method
- Find out the name and password of your maintainer object

Create a route and route6 Objects

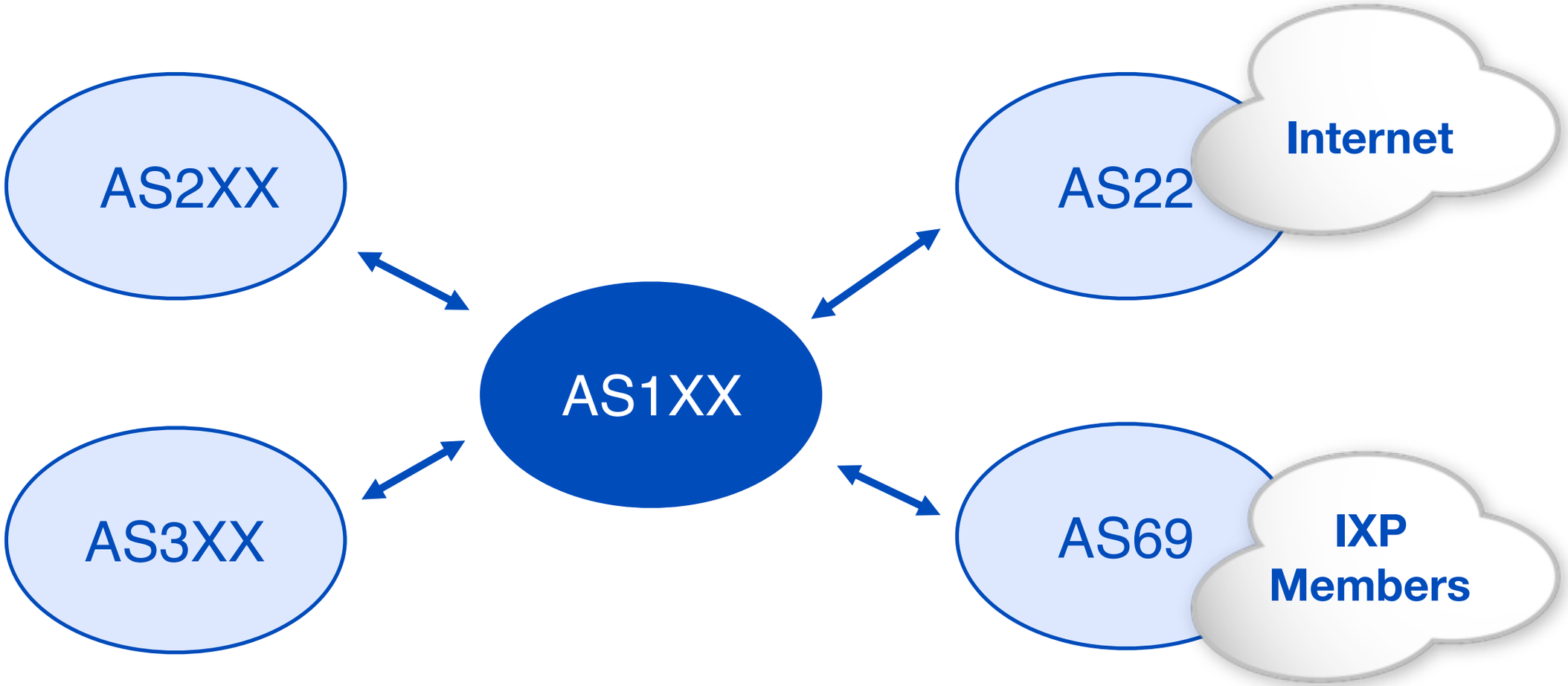
- Create a **route** object for your IPv4 allocation
- Create a **route6** object for your IPv6 allocation
- List your AS Number (**aut-num**) as the origin for both objects

Step by Step route Object

- Instructions:
  - Go to the Webupdates
  - Select “create object” and choose **route**
  - Add your IPv4 allocation prefix to your **route** object
  - Add your AS Number as the origin AS for your prefix
  - Add the correct password and submit the update
  - Go back to Webupdates and create a **route6** object

Describe You Routing Policy

- In you AS Number (**aut-num**) describe using RPSL you BGP neighbor relationships



Making Life Easier

- There are a lot of tools around that use information in the Routing Registry
- Some can generate complete router configurations like the IRRToolset
- Most are open source tools
  - You can modify them to your needs
  - Some are not very well maintained

Example Tools

- IRRToolkit (written in C++)
  - http://irrtoolset.isc.org/
- Rpsltool (perl)
  - http://www.linux.it/~md/software
- IRR Power Tools (PHP)
  - http://sourceforge.net/projects/irrpt/
- BGPQ3 (C)
  - http://snar.spb.ru/prog/bgpq3/
- Filtergen (Level 3)
  - whois -h filtergen.level3.net RIPE::ASxxx
- IRR Explorer (web)
  - http://irrexplorer.nlnog.net

Building Your Own

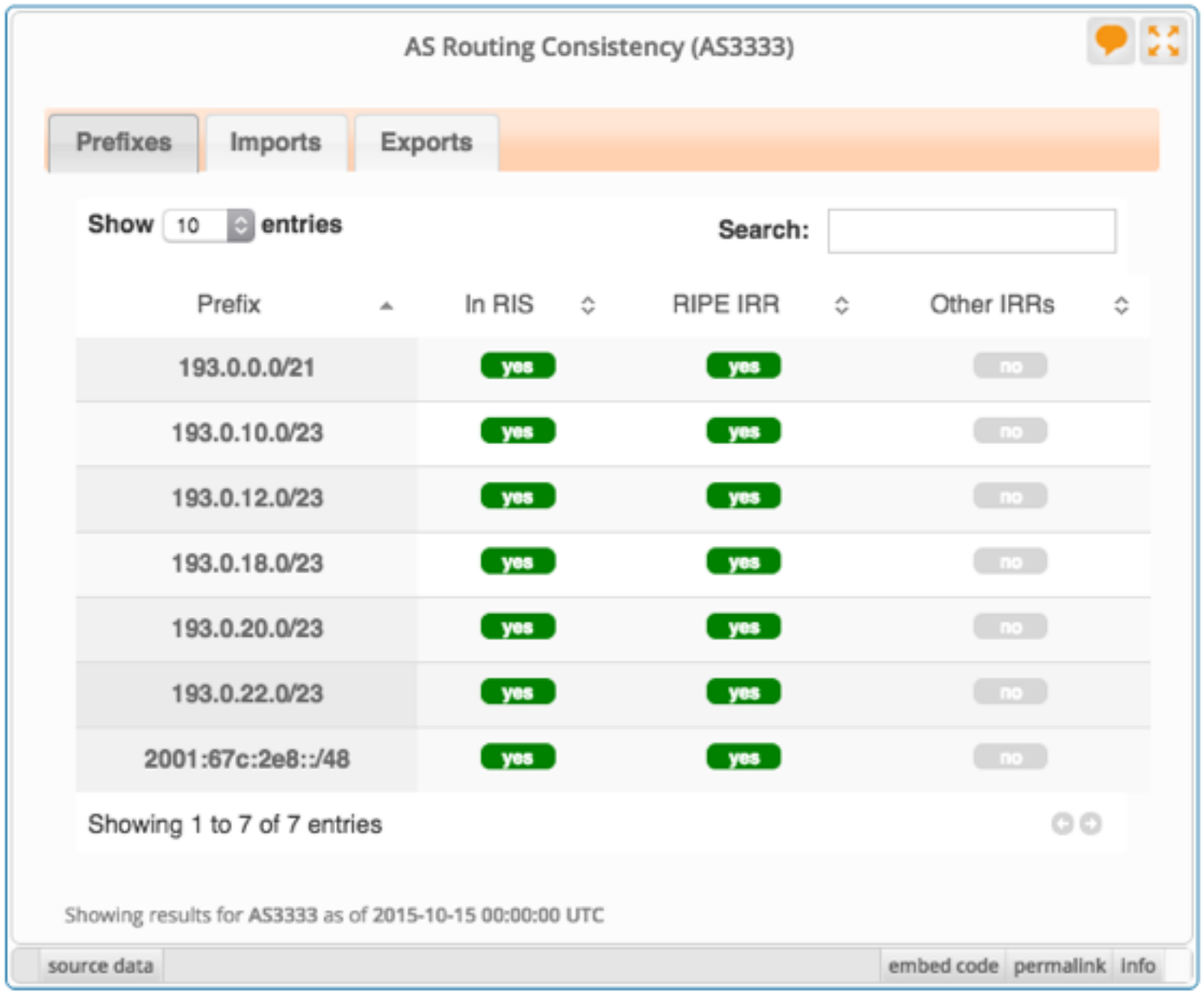
- A couple of things to keep in mind
  - The RIPE Database has limits on the number of queries you can do per day
  - Query flags or output format can change over time
- Instead of the whois interface, you can use the RESTful API for the RIPE Database
  - Uses XML or JSON for output
  - See **https://ripe.net/developer**
  - Also visit **https://labs.ripe.net** for more information

Getting the Complete Picture

- Automation relies on the IRR being complete
  - Not all resources are registered in an IRR
  - Not all information is correct
- Small mistakes can have a big impact
- Check your output before using it
  - Be prepared to make manual overrides
- Help others by documenting your policy

RIPEstat

- You can compare the Routing Registry and the Internet routing table using http://stat.ripe.net



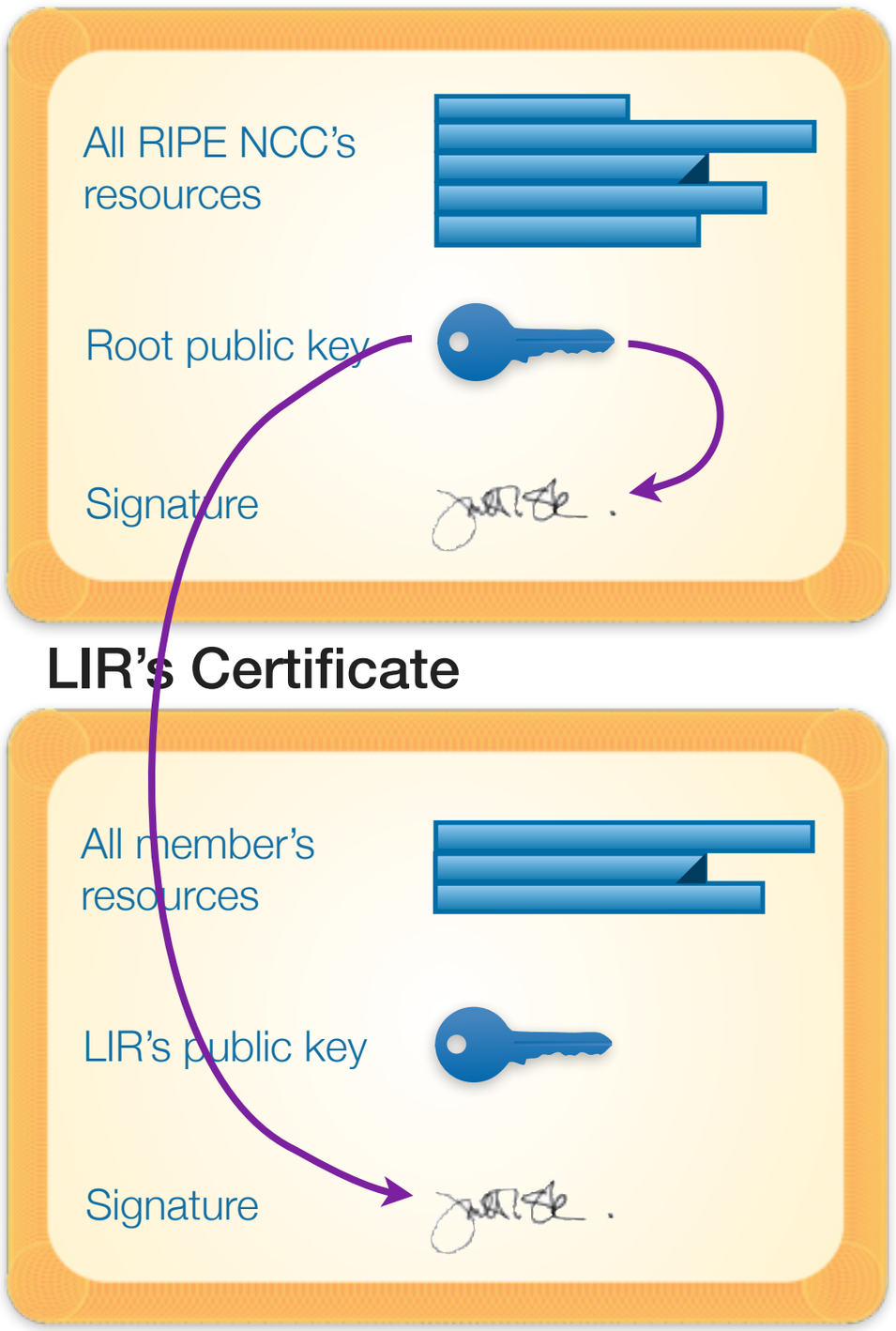
RPKI and BGPSEC

Section 10

RPKI

- A security framework for verifying the association between resource holders and their Internet resources
- Attaches digital certificates to network resources upon request that lists all resources held by the member
  - AS Numbers
  - IP Addresses
- Operators associate those two resources
  - Route Origin Authorisations (ROAs)

RPKI Chain of Trust

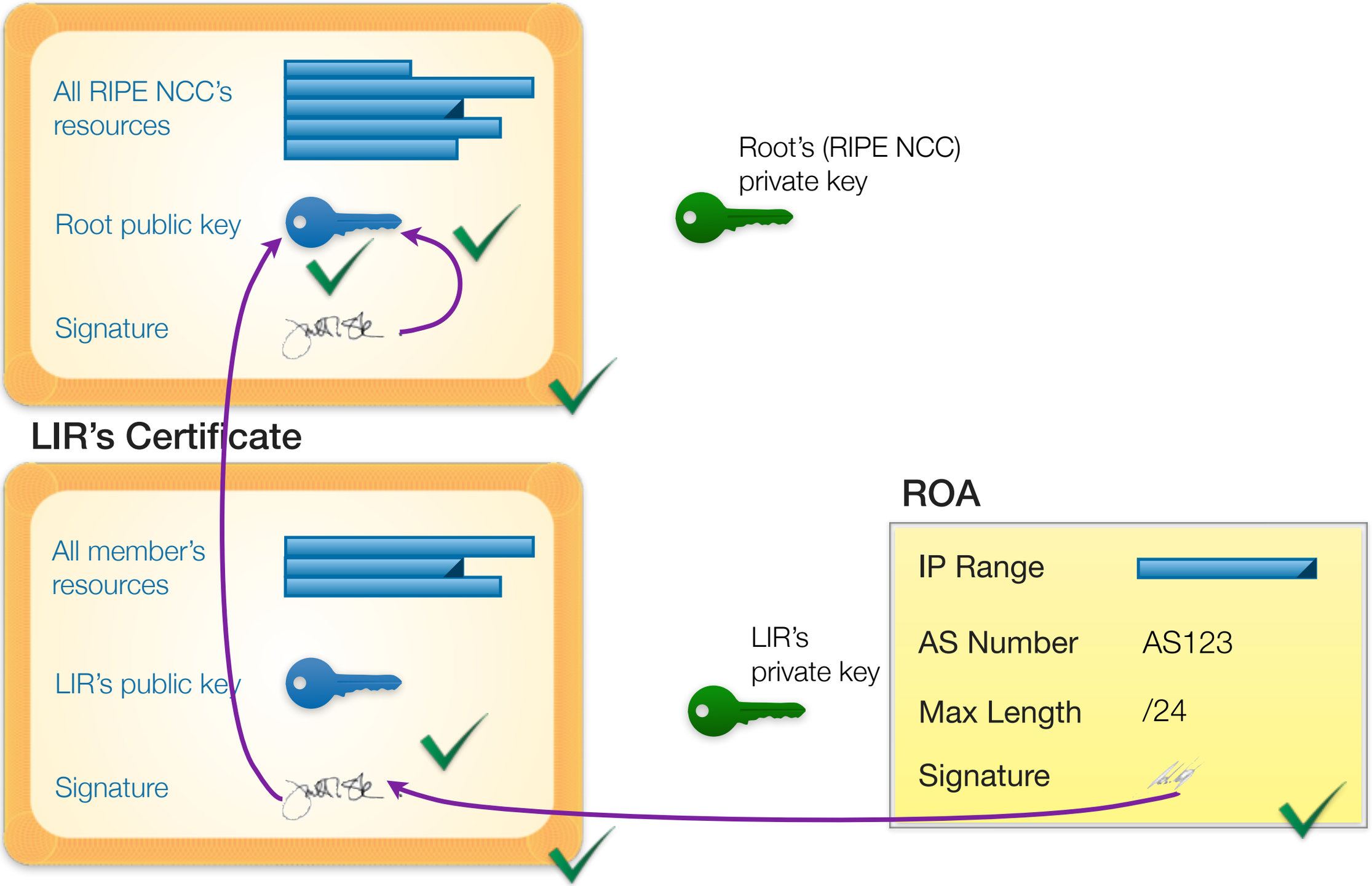


- RIPE NCC holds self-signed root certificate for all resources they have in the registry
  - Signed by the root's private key
- The root certificate is used to sign all certificates for members listing their resources
  - Signed by the root's private key

ROA (Route Origin Authorisation)

- LIRs can use their certificate to create a ROA for each of their resources (IP address ranges)
  - Signed by the root's private key
- ROA states
  - Address range
  - Which AS this is announced from (freely chosen)
  - Maximum length (freely chosen)
- You can have multiple ROAs for an IP range
- ROAs can overlap

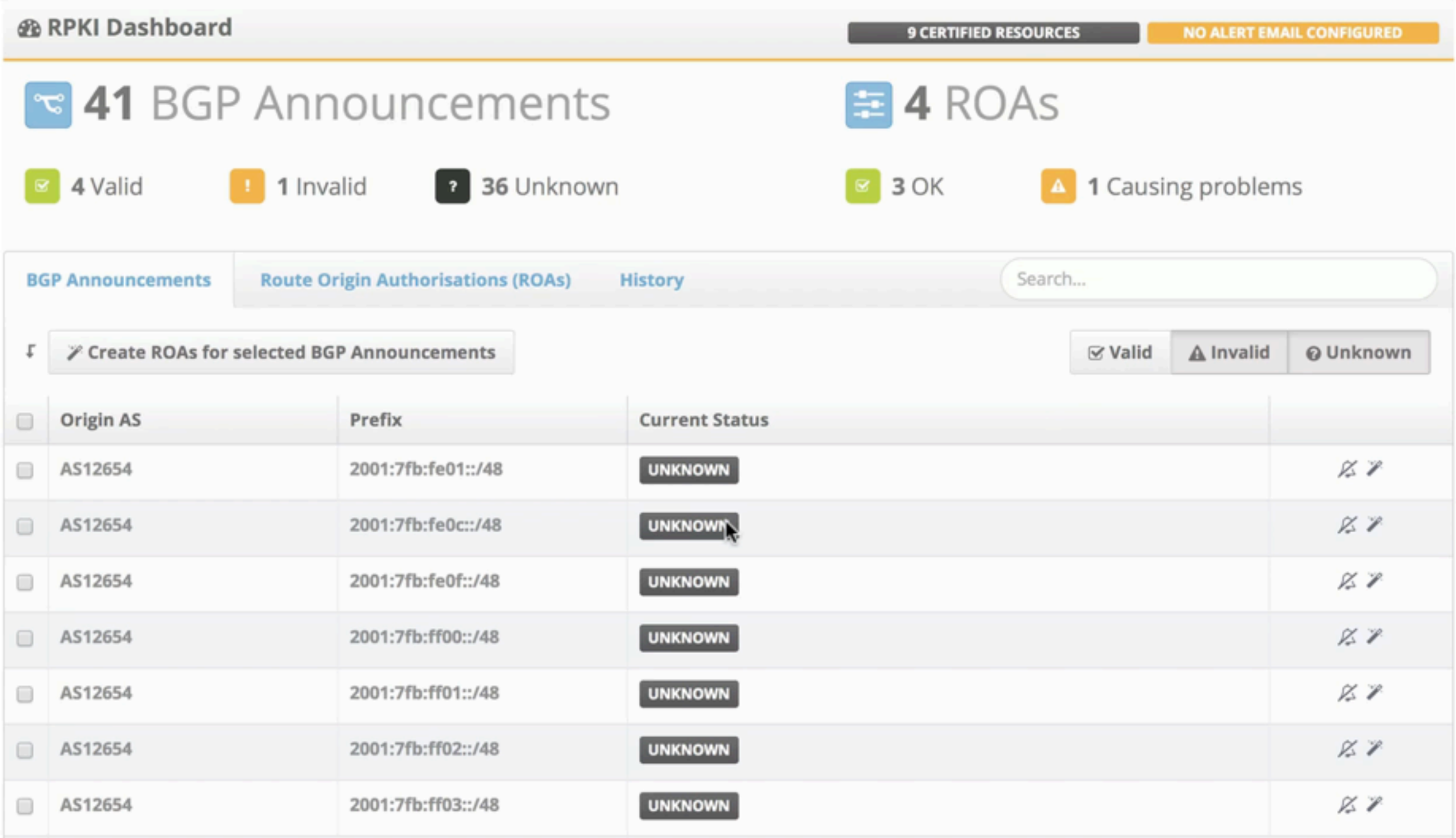
ROA Chain of Trust



Hosted RPKI

- Automate signing and key roll overs
  - One click setup of resource certificate
  - User has a valid and published certificate for as long as they are the holder of the resources
  - Changes in resource holdership are handled automatically
- Hide all the crypto complexity from the UI
  - Hashes, SIA and AIA pointers, etc.
- Just focus on creating and publishing ROAs
  - Match your intended BGP configuration

Creating ROA



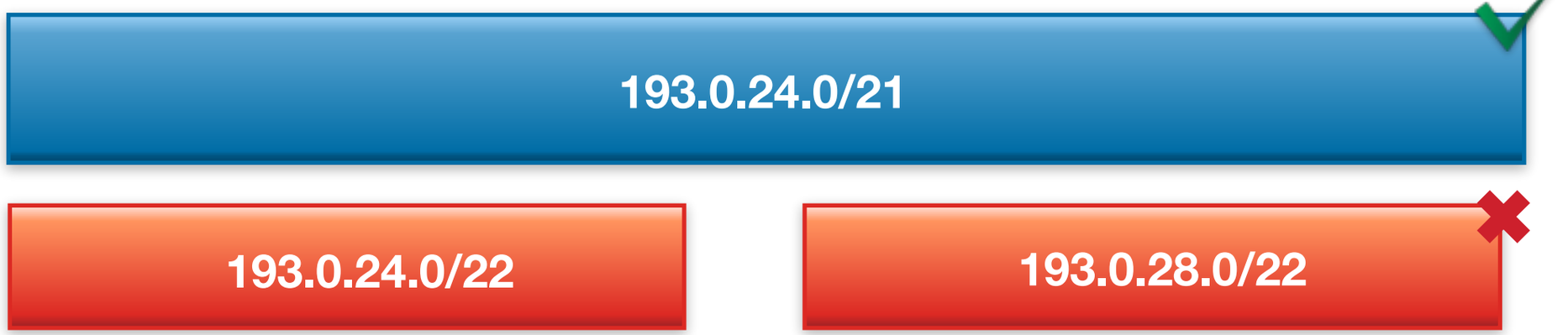
ROA (Route Origin Authorisation) Example

ROA

193.0.24.0/21

AS2121

Max Length: _



ROA

193.0.24.0/23

AS2121

Max Length: /24

ROA

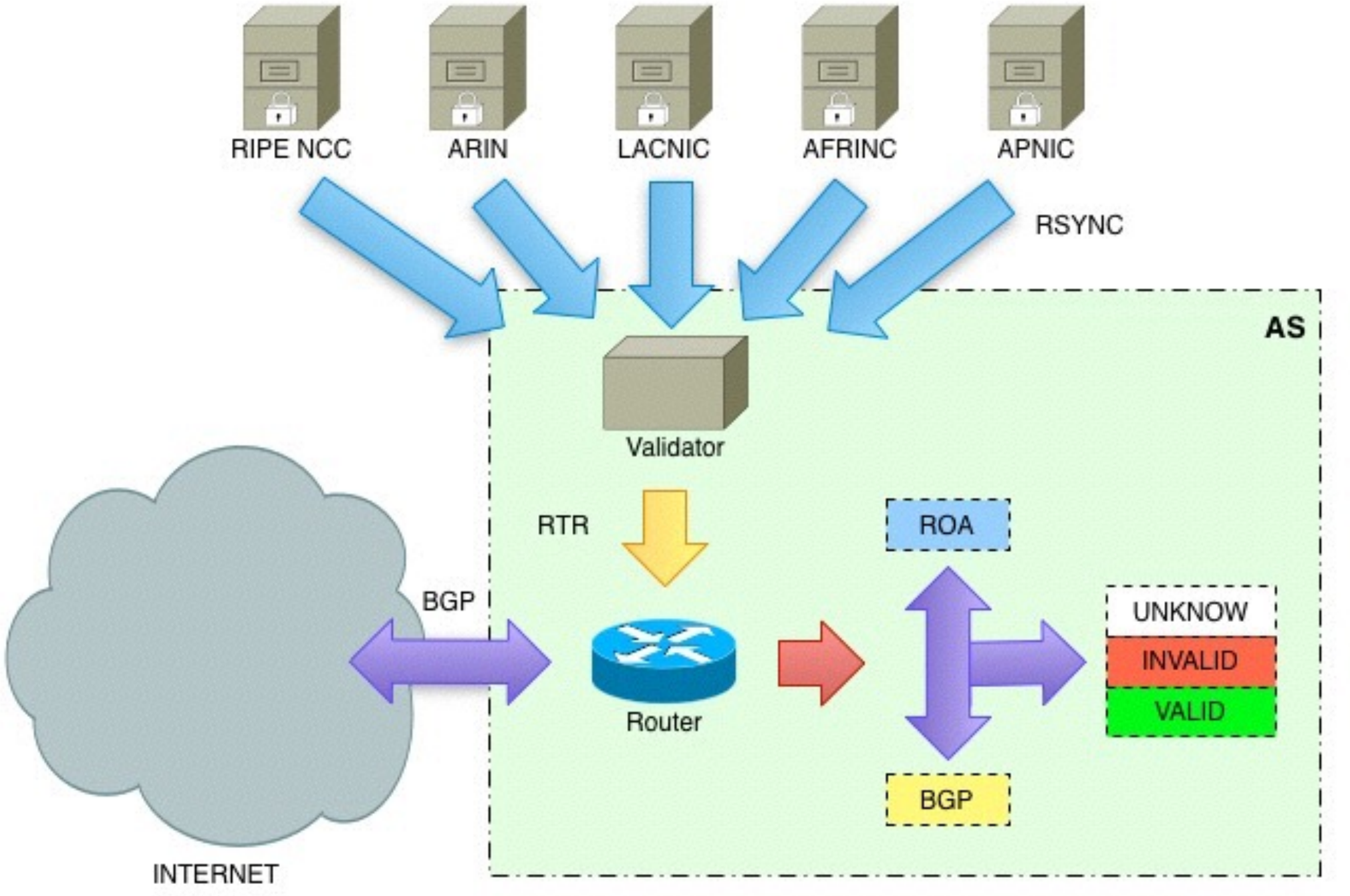
193.0.30.0/23

AS2121

Max Length: _

RIPE NCC

Relying Party



Invalid ROA

- Invalid ROA
  - The ROA in the repository cannot be validated by the client (ISP) so it is not included in the validated cache
- Invalid BGP announcement
  - There is a ROA in validated cache for that prefix but for a different AS.
  - Or the max length doesn’t match.
- If no ROA in the cache then announcement is “unknown”

RPKI Implementations

- RPKI and RPKI-RTR Protocol are an IETF standard
- All router vendors can implement it
- Cisco support:
  - XR 4.2.1 (CRS-x, ASR9000, c12K) / XR 5.1.1 (NCS6000, XRv)
  - XE 3.5 (C7200, c7600, ASR1K, CSR1Kv, ASR90x, ME3600…)
  - IOS15.2(1)S
- Juniper has support since version 12.2
- Quagga has support through BGP-SRX
- BIRD has support for ROA but does not do RPKI-RTR

BGPSEC - The Next Step

- The RPKI prevents configuration errors by an ISP from hijacking address space
  - The RPKI does not protect against attacks on BGP, e.g., bogus routes terminating in a valid origin
- To protect against attacks, one needs to enable every AS to verify that the route received via a BGP UPDATE message is accurate

BGPSEC Operations

- New, optional, transitive attribute, to carry digitally signed route info
- Support is negotiated between routers, non BGPSEC router will not be burdened by big UPDATE messages
- Data is never sent through non BGPSEC ASes, so secure paths exist only for contiguous sequences of ASes
- Incremental deployment is possible

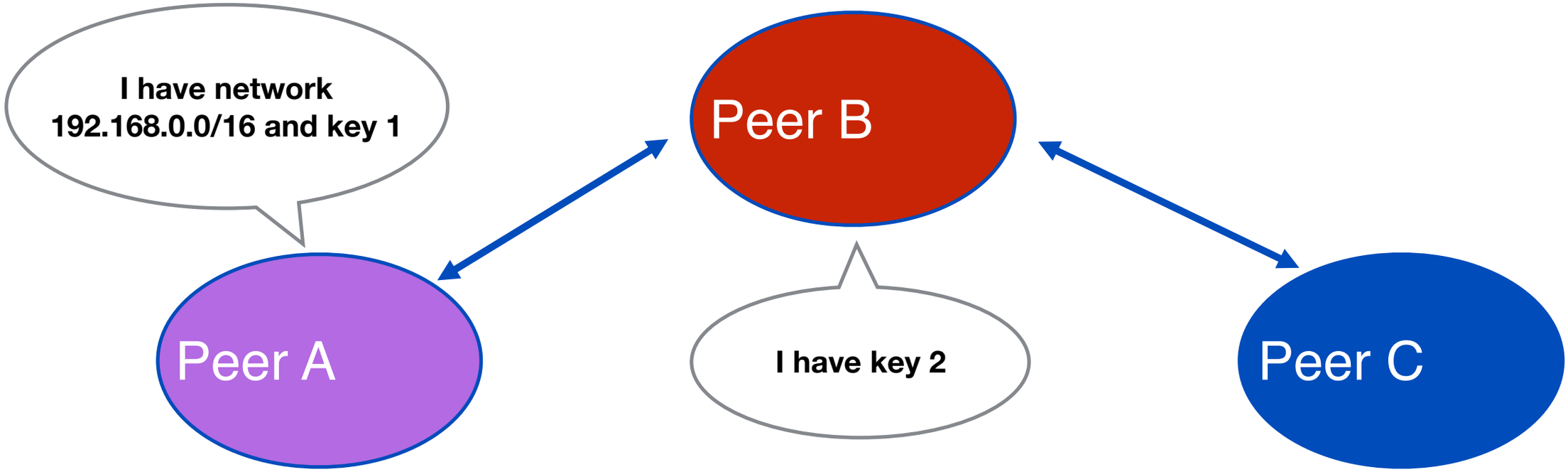
How Does BGPSEC Work?

BGP UPDATE

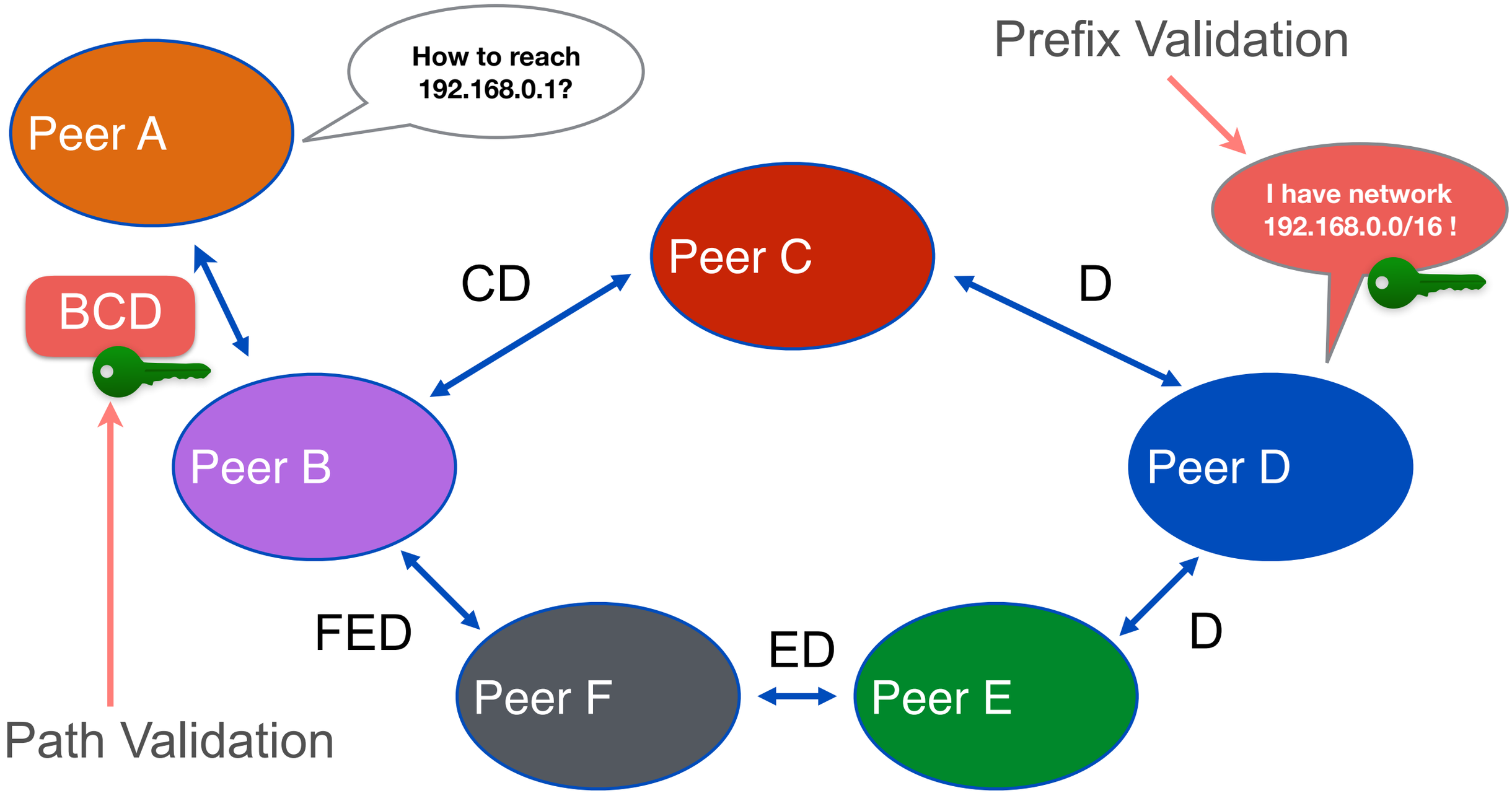
Network: 192.168.0.0/16
AS Path: A
BGPSEC: (key1, signature1)

BGP UPDATE

Network: 192.168.0.0/16
AS Path: B, A
BGPSEC: (key1, signature1)
(key2, signature2)



RPKI with BGPSEC



BGP Security Status

- RPKI
  - RPKI and RPKI-RTR are an IETF standards (RFC5280, RFC3779, RFC6481-6493)
  - RIRs are in production since 2011
    - http://certification-stats.ripe.net/
  - Most vendors already implemented it (testbeds are available)
- BGPSEC (IETF Draft)
  - Threat and requirements document published (RFC7132, RFC 7353)
  - Router vendors are working on designs for real implementations

BGP Software

Section 11

BGP Software

- Many different BGP implementations exist
- Many are open source
- Running mainly on Unix/Linux
- You can run your own BGP router on your PC

Quagga

- Successor of Zebra, which was the first BGP daemon for UNIX/Linux
- Supports also rip, ripng, ospf, ospfv3, is-is, mpls
  - Cisco-style CLI
- http://www.nongnu.org/quagga/

Bird

- Developed by CZ.NIC
  - Works on any UNIX/Linux
- A full suite of routing protocols
  - BGP, RIP, OSPF, BFD
- Most popular system for Route Servers now
- http://bird.network.cz/

OpenBGPd

- Developed as part of OpenBSD
  - Ported to other UNIX/Linux platforms
  - Supports also ospf through ospfd, and MPLS
- Written with security in mind
- Tightly integrated with pf, the packet filter
- http://www.openbgpd.org/

GoBGP

- Developed for performance
  - Full use of todays multicore processor
- Developed for automation
  - Supports RPC APIs
  - Many data formats (toml / json / yaml / hcl)
- http://osrg.github.io/gobgp/

ExaBGP

- A toolkit to “speak BGP”
  - Not a real BGP daemon
- You can hook up scripts, software, functions to any route update
- https://github.com/Exa-Networks/exabgp

OpenBMP

- BGP Monitoring Protocol
  - IETF Draft
  - Full view into operation of BGP Speaker (RAW data)
  - Implemented by Cisco IOS XE, XR and JunOS
- OpenBMP
  - BMP devices send BMP messages to a OpenBMP collector/daemon
  - RAW BGP data can be read via API
- http://openbmp.org/

PMACCT

- Set of small multi-purpose passive network monitoring tools

- Many import and export capabilities
  - libpcap, NFLOG, NetFlow, sFlow, IPFIX
  - SQL and no SQL databases, AMQP and Kafka message brokers and flat files

- http://pmacct.net

Looking glass

- A software that lets you query a BGP speaker
- Many built for Cisco, Juniper, Quagga
  - Now also for Bird
- Helpful for diagnostics and to check your configuration with other peers
  - Gives you an insight into other networks

Retrieving Information from the IRR

Exercise

A Look at the Real World

- Have a look at AS 3333 in the RIPE Database
  - Which prefixes would you accept from AS 3333 if it was your customer?
- Remember to use the real database!
- Optionally verify the results using the tools at http://stat.ripe.net

BGP Tips & Tricks

Section 12

EGP vs IGP

- Never redistribute routes from the IGP into BGP
- Never redistribute routes from BGP into the IGP

- The default admin distance for eBGP updates on IOS and XR is lower than for IGPs
  - eBGP updates take preference to IGP updates

MD5

- Not as useful
- Primary use at IXP is to stop session hijacking on address re-use
  - Some companies have security policies which require it
- Formally obsoleted by TCP-AO since June 2010
  - Still no production TCP-AO implementations

Flow Collection

- Export information about packets routed through your network
  - Traffic sampled is send to a collector
  - A variety of commercial and open-source tools to collect and display these flow records
  - Profile your traffic
- Many Flow protocols:
  - NetFlow (v5,v9)
  - sFlow
  - jFlow

Managing Multiple Protocols

- Independent operation
  - One RIB per protocol
  - Distinct policies per protocol (IP address specific route maps and prefix lists must be adjusted)
    - Make separate route maps for IPv4 and IPv6
    - Prefix lists are always separate
    - It is common to use a **-v4** and a **-v6** suffix to names

IXP Hygiene

- Unicast Internet Exchange only:
  - unicast packets between member networks
  - broadcast ARP for IXP IPv4 addresses
  - multicast IPv6 NS/NA for IXP IPv6 addresses
- Do not:
  - DHCP, IPv6 SLAAC, STP, bridging, VTP, proxy ARP
  - multicast: PIM, IGMP, MLD
  - network discovery: CDP, LLDP, EDP

Next Hop and IXP

- eBGP allows you to set the BGP next hop address to be any address on the link LAN
- At an IXP, you can configure next hop IP to be any address on peering LAN
  - Recommend checking peer address = next hop address
  - Documented in draft-ietf-grow-ix-bgp-routeserveroperations

Getting Transit

- Find well peered transit providers
  - Can improve quality and shorten AS paths
  - No capacity problems

- Find your top traffic destinations:
  - Can improve quality
  - Peer with them or find closer upstream
  - Traffic profile from flow collectors can be useful

Common Mistakes

- No diversity
  - All reached over same cable
  - All connect to the same transit
  - All have poor onward transit and peering arrangements
- Signing up with too many transit providers
  - Lots of small circuits
  - These cost more per Mbps than larger ones

Check your visibility

- RIPEstat
  - https://stat.ripe.net/
- VizAS
  - https://labs.apnic.net/vizas/
- Bgpmon
  - http://routeviews.org/
  - http://bgplay.routeviews.org
  - http://traceroute.org
- RIPE Atlas
  - http://atlas.ripe.net/
- NLNOG Ring
  - http://ring.nlnog.net/
- HE BGP Toolkit
  - http://bgp.he.net/
- Sonar
  - http://www.v6sonar.com

Best practises

- Apply BCP38
  - Use inbound and outbound packet filters to protect network
  - Example
    - Outbound: only allow my network source addresses out
    - Inbound: only allow specific ports to specific destinations in
- Update your Routing Registry information!
- The Routing Resilience Manifesto initiative
  - https://www.routingmanifesto.org/

Questions

RIPE NCC
Academy

Graduate to the next level!

http://academy.ripe.net

Feedback

www.ripe.net/training/bgp/survey

Follow us!

@TrainingRIPENCC

The End!

Край

Y Diwedd

Fí

Finis

النهاية

Соңы

Վերջ

Liðugt

Ende

Finvezh

Кінець

Konec

Kraj

Ënn

Fund

پایان

Kpaj

Lõpp

Beigas

Vége

Son

An Críoch

הסוף

Endir

Fine

Sfârşit

Fin

Τέλος

Einde

Конец

Slut

Slutt

დასასრული

Pabaiga

Tmiem

Koniec

Fim

Amaia

Loppu