Border Gateway Protocol

Overview

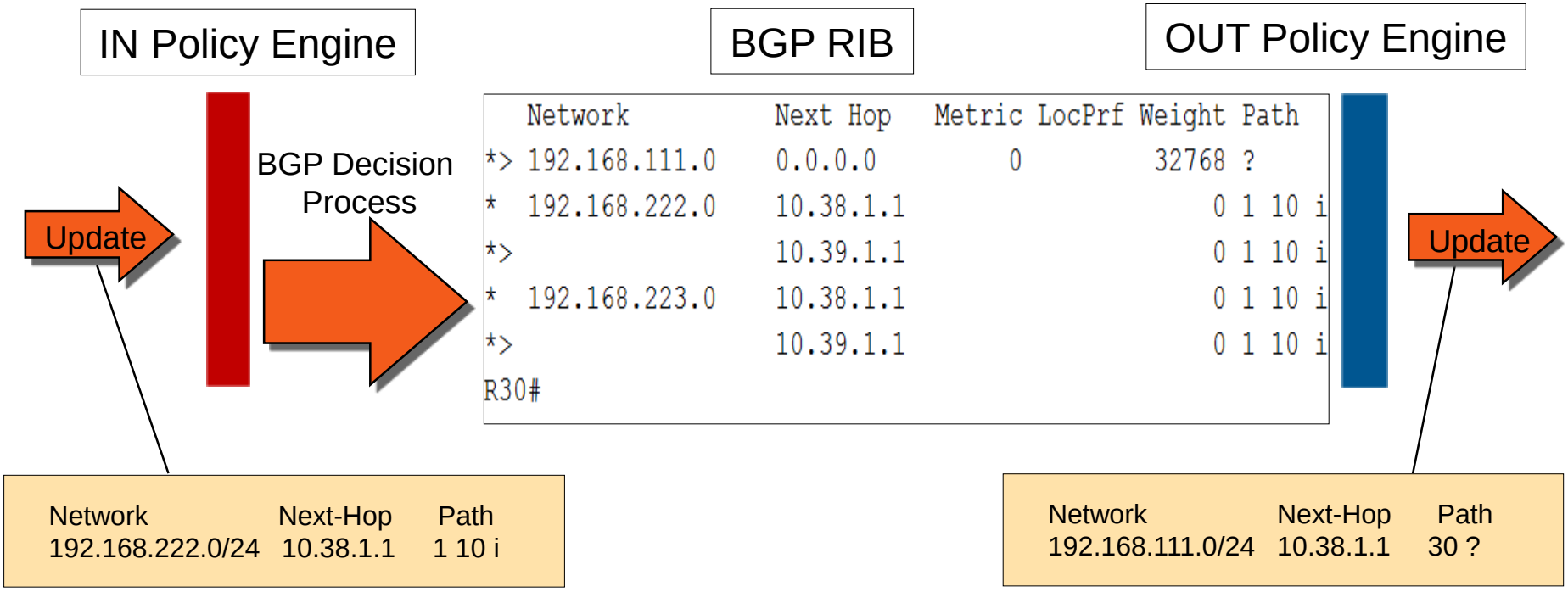
- Interior
  - Automatic discovery
  - Generally trust your IGP routers
  - Routes go to all IGP routers
- Exterior
  - Specifically configured peers
  - Connecting with outside networks
  - Set administrative boundaries

Quiz

- Why do we need BGP?
- Benefits?
- Distance vector or link state?
- BGP session establishment?
- What is iBGP and eBGP? Comparison?
- How stable iBGP peering can be achieved?
- Which three types of route injection to the BGP you know?

BGP Routing Information Base

- BGP “in” process
  - receives path information from peers,
  - results of BGP path selection placed in the BGP table, “best path” flagged (denoted by “>”)
- BGP “out” process
  - builds update using info from the table, may modify update based on config, sends update to peers



BGP Routing Information Base

- paths installed in routing table if
  - prefix and prefix length are unique
  - lowest “protocol distance”
  - Next hop is valid

```
   Network          Next Hop    Metric LocPrf Weight Path
*> 192.168.111.0    0.0.0.0          0         32768 ?
*  192.168.222.0    10.38.1.1                      0 1 10 i
*>                  10.39.1.1                      0 1 10 i
*  192.168.223.0    10.38.1.1                      0 1 10 i
*>                  10.39.1.1                      0 1 10 i
R30#
```

BGP RIB

Routing Table

```
C    192.168.111.0/24 is directly connected, FastEthernet0/1
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.39.1.0/24 is directly connected, FastEthernet0/0
C       10.38.1.0/24 is directly connected, FastEthernet1/0
C       10.1.255.30/32 is directly connected, Loopback0
B    192.168.223.0/24 [20/0] via 10.39.1.1, 01:47:16
B    192.168.222.0/24 [20/0] via 10.39.1.1, 01:47:16
```

Administrative Distance

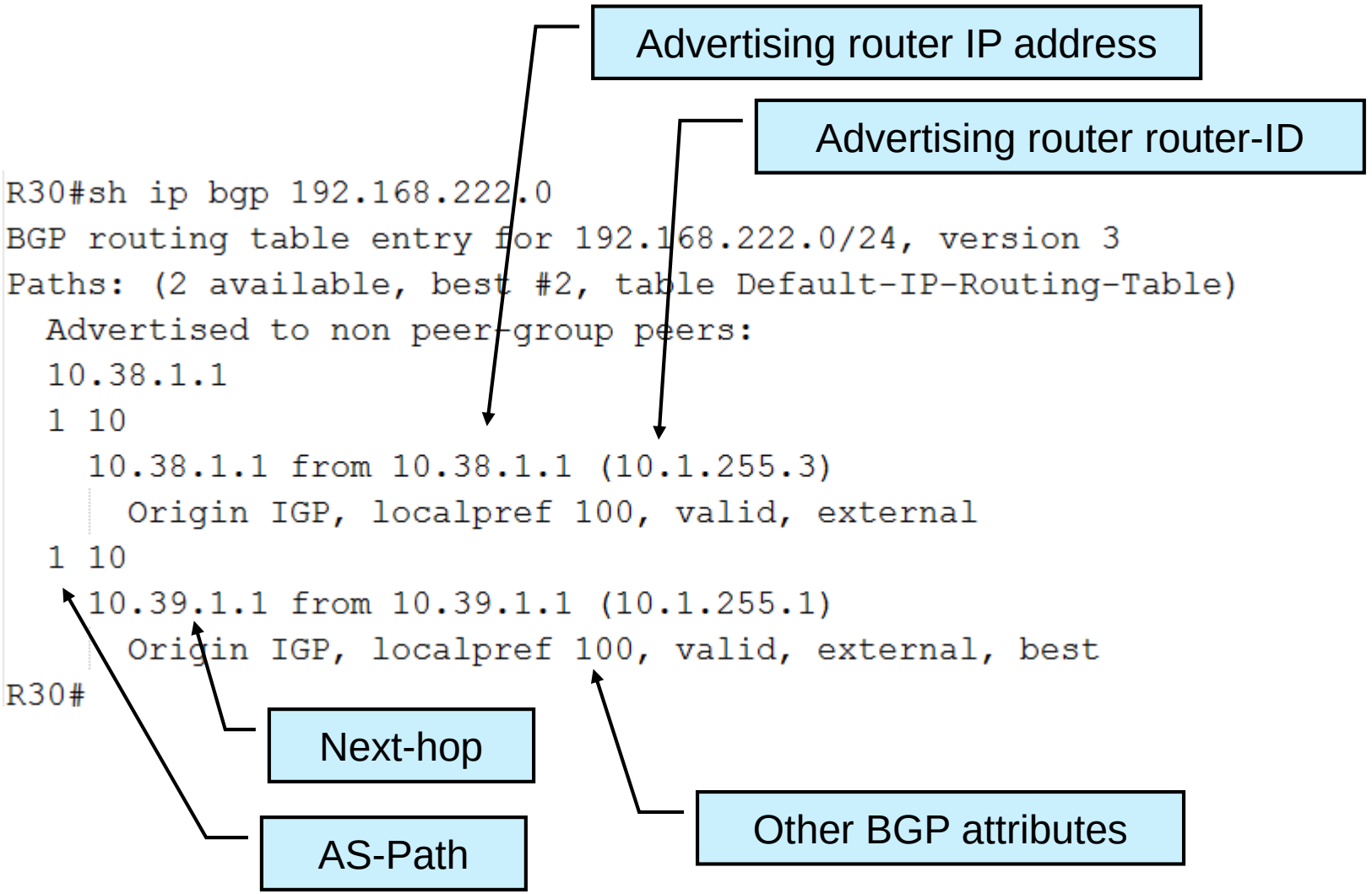
- Administrative distance defines the reliability of a routing protocol. A lower numerical value is preferred and installed in the routing table

Protocol	Administrative distance
Directly connected route	0
Static route out an interface	1°
Static route to next-hop address	1
EIGRP summary route	6
External BGP	20
Internal EIGRP	100
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
ODR	160
External EIGRP	170
Internal BGP	200
DHCP-learned	254
Unknown	255

Routing Table

```
C    192.168.111.0/24 is directly connected, FastEthernet0/1
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.39.1.0/24 is directly connected, FastEthernet0/0
C       10.38.1.0/24 is directly connected, FastEthernet1/0
C       10.1.255.30/32 is directly connected, Loopback0
B    192.168.223.0/24 [20/0] via 10.39.1.1, 01:47:16
B    192.168.222.0/24 [20/0] via 10.39.1.1, 01:47:16
```

Displaying Entries in BGP Table



The BGP Decision Algorithm - Summary

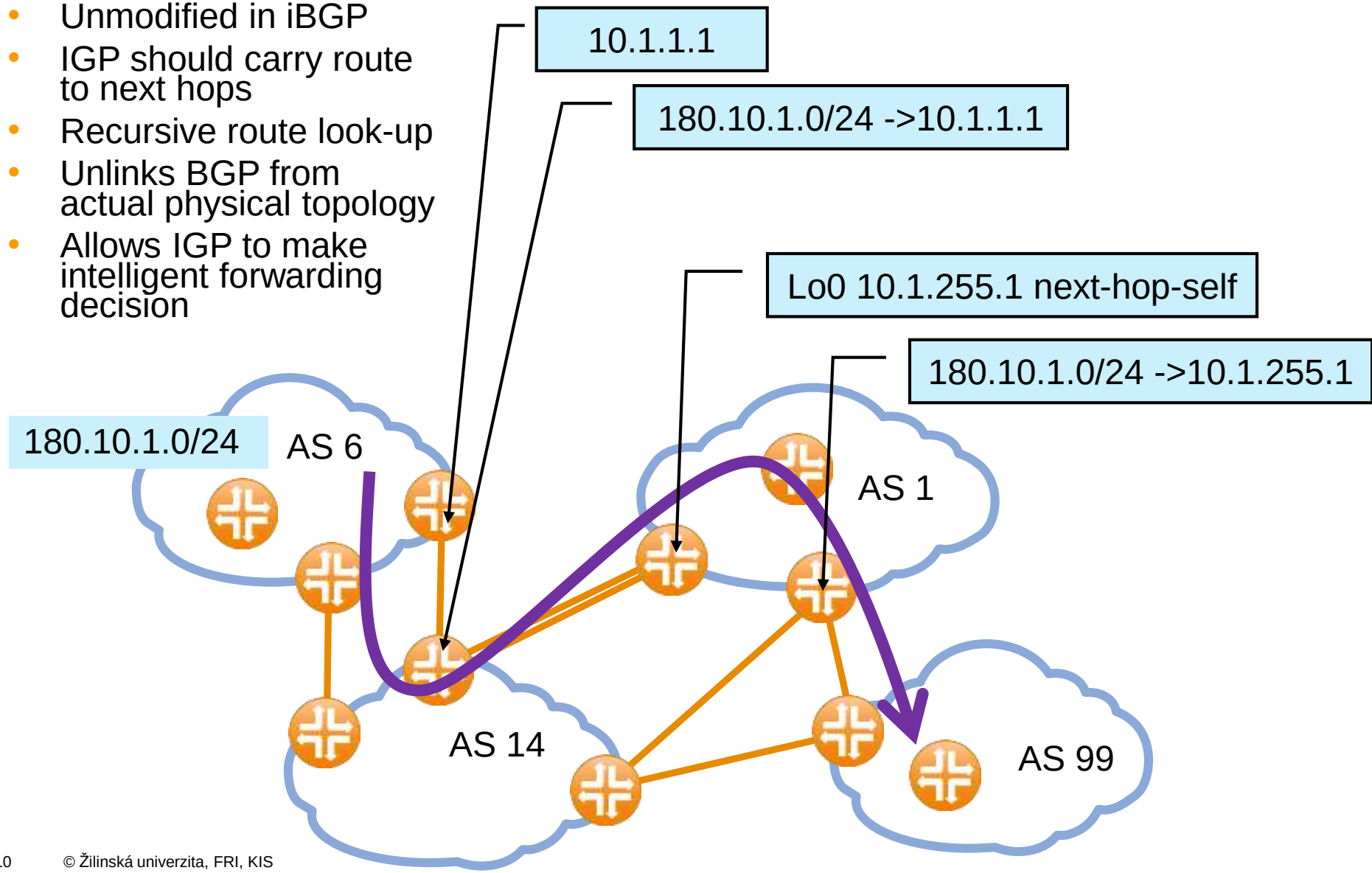
- After BGP router receives updates about destinations and paths from peers, the protocol will have to decide which path to choose in order to reach a specific destination
  - BGP by default will choose only a single path to reach a specific destination
  - The decision process is based on different attributes, such as next hop, local preference, the route origin, and so on
  - BGP will always propagate the best path to its neighbors

How BGP selects the best path (Cisco)

1. If **Next Hop** is inaccessible do not consider it
2. Prefer the largest **Weight** (local to router)
3. If same Weight prefer largest **Local Preference** (global within AS)
4. If same Local Preference prefer the route that the specified router has **originated** (network, aggregate)
5. If no route was originated prefer the shorter **AS path**.
6. If all paths are external prefer the lowest **Origin** code (IGP<EGP<INCOMPLETE)
7. If Origin codes are the same prefer the path with the lowest **MED**
8. If path is the same length prefer the **eBGP** path over iBGP
9. If IGP synchronization is disabled and only internal path remain prefer the path through the closest **IGP neighbor** (lower metric to next hop)
10. Prefer the route with the lowest **BGP router ID**.

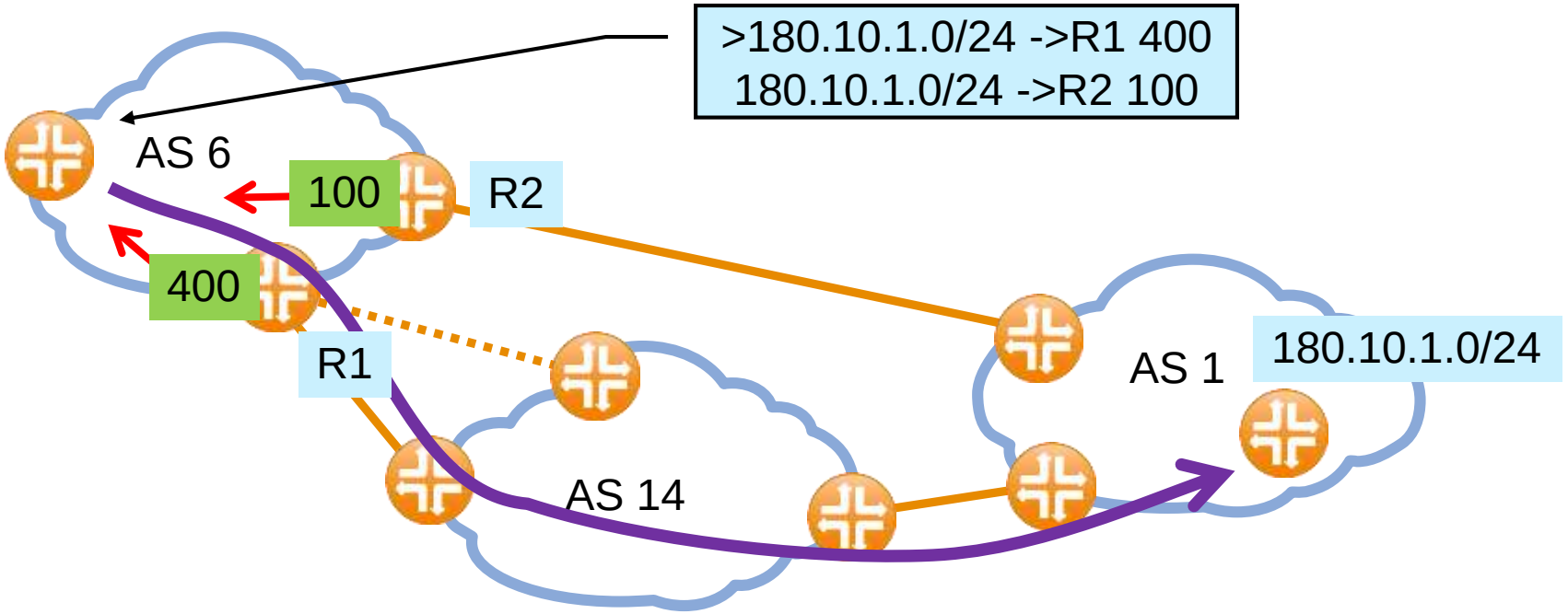
Next Hop

- Unmodified in iBGP
- IGP should carry route to next hops
- Recursive route look-up
- Unlinks BGP from actual physical topology
- Allows IGP to make intelligent forwarding decision



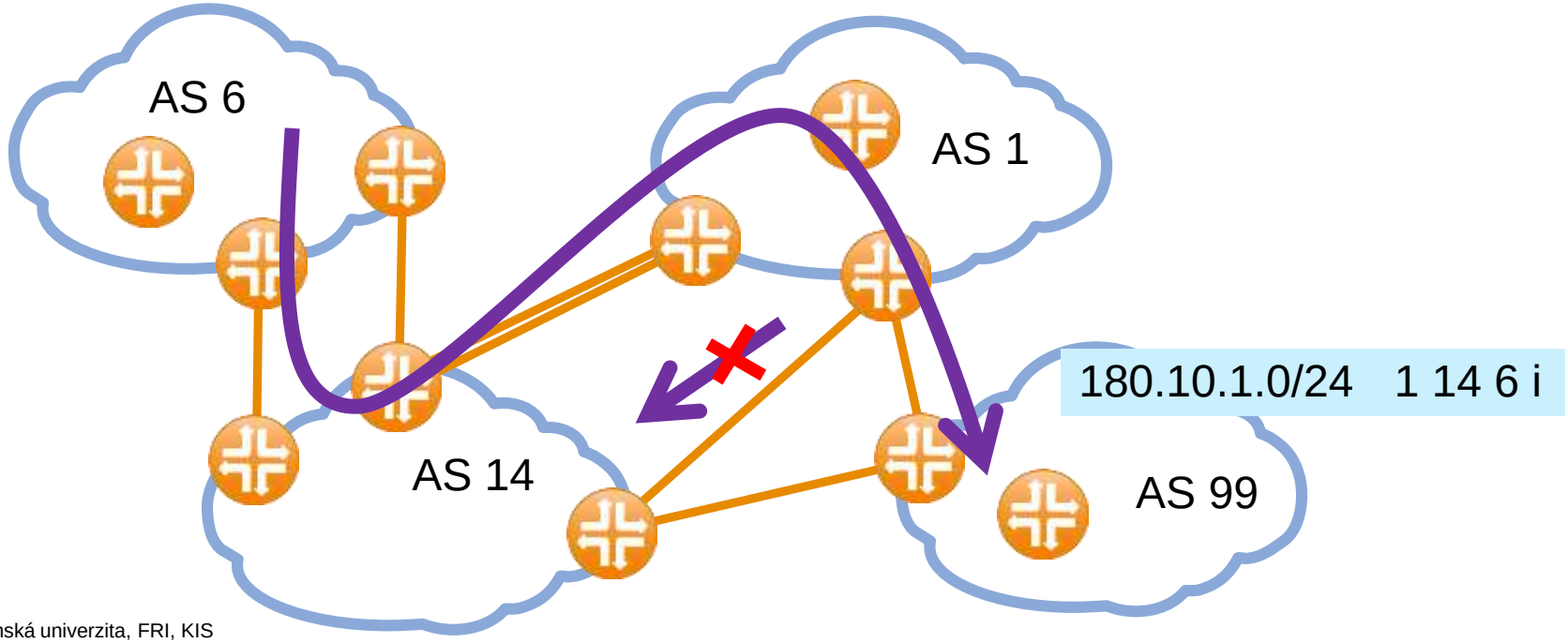
Local Preference

- I ntra-AS attribute not used for eBGP, Well-known discretionary for iBGP
- Default value is 100
- Local to an AS
- Used to prefer one exit over another
- Path with highest local preference wins



AS-Path

- Sequence of ASes a route has traversed
- Updated by the sending router with its AS number
- Contains the list of AS numbers the update traverses.
- Used to detect routing loops
  - Each time the router receives an update, if it finds its AS number, it discards the update
- Well-known mandatory, the shortest path wins



Origin

- Well-known mandatory

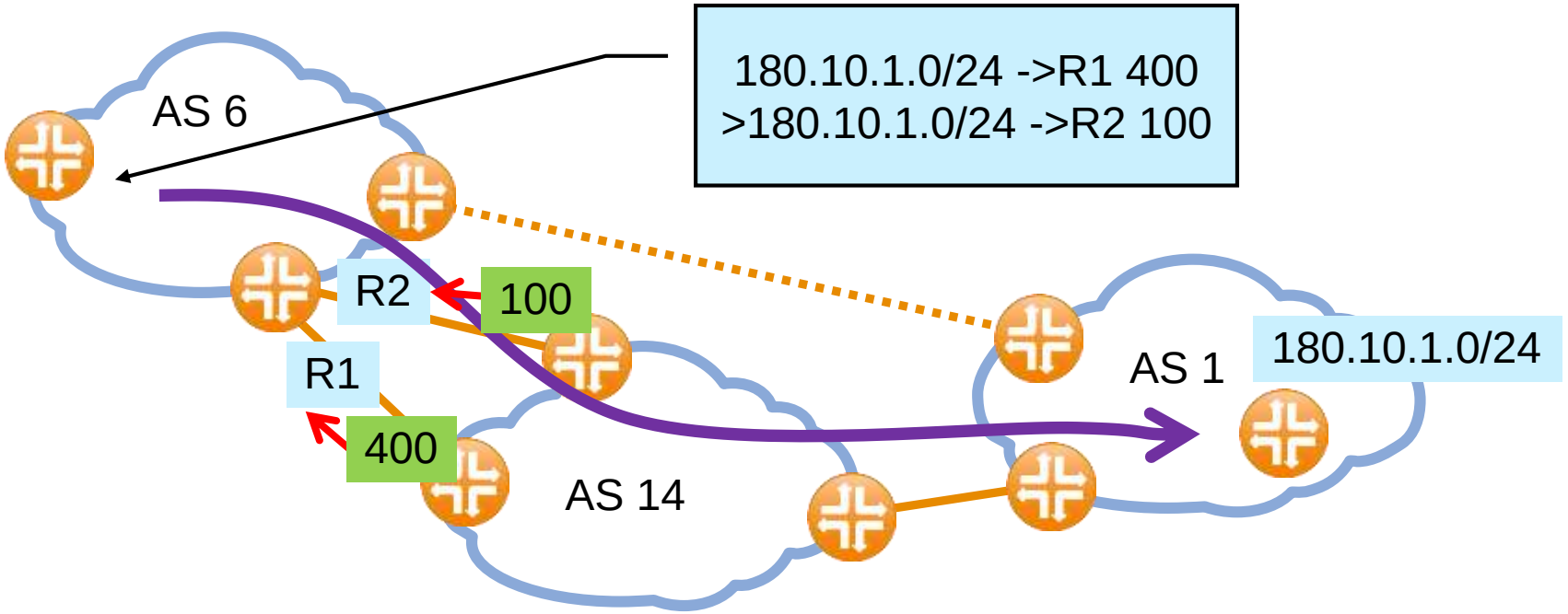
- Three values
  - IGP - Generated using “network” statement
  - EGP - Redistributed from EGP
  - Incomplete - Redistribute IGP, connected, static

- The PATH selection preference is
  - IGP(0)<EGP(1)<INCOMPLETE(2)

```
   Network          Next Hop    Metric LocPrf Weight Path
*> 192.168.111.0    0.0.0.0          0         32768 ?
*  192.168.222.0    10.38.1.1                      0 1 10 i
*>                  10.39.1.1                      0 1 10 i
*  192.168.223.0    10.38.1.1                      0 1 10 i
*>                  10.39.1.1                      0 1 10 i
R30#
```

Multi-Exit Discriminator

- Non-transitive, optional
- Represented as a numeric value (0-0xffffffff)
- Used to convey the relative preference of entry points
- Comparable if paths are from the same AS
- Path with lower MED wins
- IGP metric can be transformed and delivered as MED



Atomic Aggregate and Aggregator

- ATOMIC_AGGREGATE
  - A well-known discretionary attribute. It is used by a BGP speaker to inform other BGP speakers that the local system selected a less-specific route without selecting a more-specific route that is included in it. The "as-set" option when generating an aggregate advertisement with “aggregate-address” builds an AS path “{set}" based on the AS path information of the original prefixes that have become the aggregate prefix.

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 160.0.0.0/8	4.4.4.1			0	300 {200,100} i

- AGGREGATOR
  - An optional transitive attribute. The attribute contains the last AS number that formed the aggregate route (encoded as two octets), followed by the IP address of the BGP speaker that formed the aggregate route (encoded as four octets)
  - When the ATOMIC_AGGREGATE attribute is set, the BGP speaker has an option of attaching the AGGREGATOR attribute

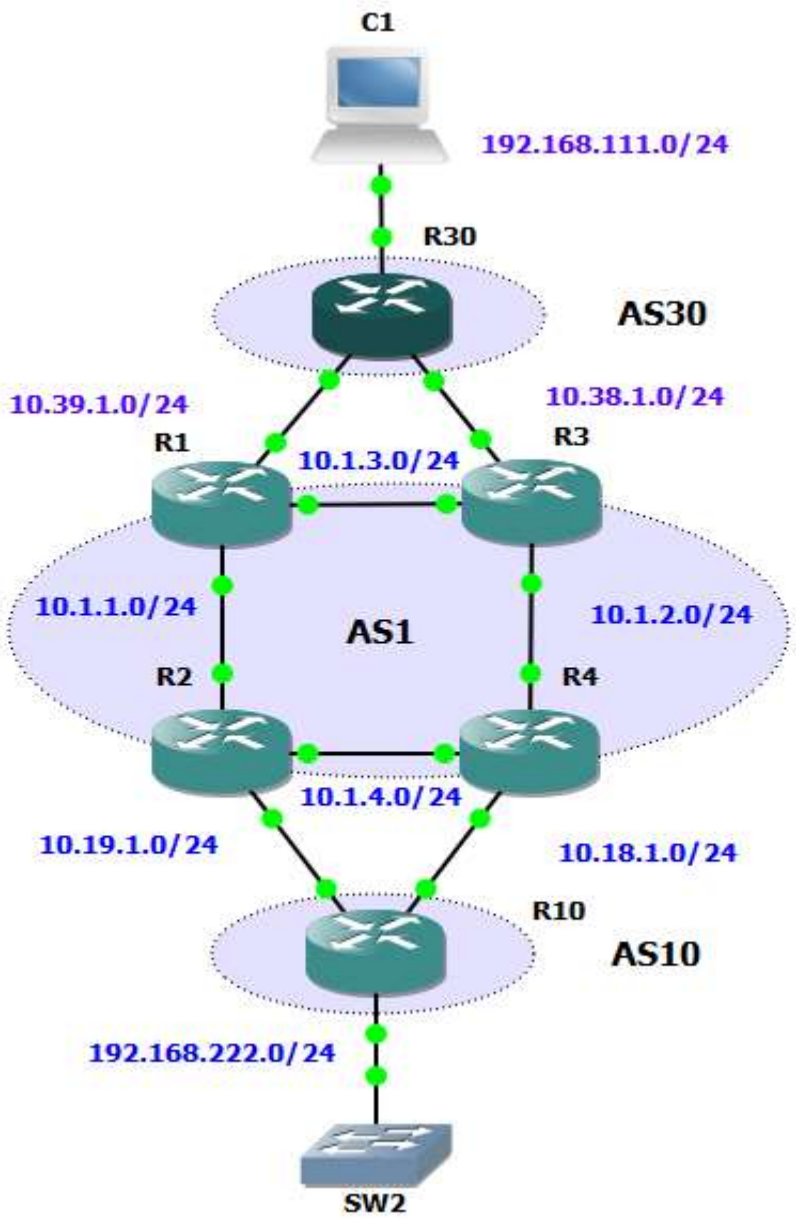
- http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5441-aggregation.html

Topology

- Why route 192.168.111.0 with nh 10.38.1.2 is selected?
- Why route 192.168.223.0 with nh 10.1.255.4 is selected?

```
R3#sh ip bgp
BGP table version is 4, local router ID is 10.1.255.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - int
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i192.168.111.0    10.1.255.1               0    100      0 30 ?
*>                  10.38.1.2                0             0 30 ?
* i192.168.222.0    10.1.255.2               0    100      0 10 i
*>i                 10.1.255.4               0    100      0 10 i
* i192.168.223.0    10.1.255.2               0    100      0 10 i
*>i                 10.1.255.4               0    100      0 10 i
R3#
```



Local Preference Configuration

```
R1#
router bgp 1
 neighbor 10.39.1.2 remote-as 30
 neighbor 10.39.1.2 route-map LP200 in
!
route-map LP200 permit 10
 set local-preference 200
!

R3#sh ip bgp  192.168.111.0
BGP routing table entry for 192.168.111.0/24, version 5
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.38.1.2
  30
    10.1.255.1 (metric 2) from 10.1.255.1 (10.1.255.1)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
  30
    10.38.1.2 from 10.38.1.2 (10.1.255.30)
      Origin incomplete, metric 0, localpref 100, valid, external
R3#
```

AS-Path Prepending

```
R30#
router bgp 30
 neighbor 10.39.1.1 remote-as 1
 neighbor 10.39.1.1 route-map PREPEND out
!
route-map PREPEND permit 10
 set as-path prepend 30 30
!

R1#sh ip bgp
BGP table version is 9, local router ID is 10.1.255.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i192.168.111.0    10.1.255.3               0    100      0 30 ?
*>                  10.39.1.2                0             0 30 ?
*>i192.168.222.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
*>i192.168.223.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
R1#
R1#
R1#sh ip bgp
BGP table version is 10, local router ID is 10.1.255.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i192.168.111.0    10.1.255.3               0    100      0 30 ?
*                   10.39.1.2                0             0 30 30 30 ?
*>i192.168.222.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
*>i192.168.223.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
R1#
```

MED Configuration

```
R30#
router bgp 30
 neighbor 10.39.1.1 remote-as 1
 neighbor 10.39.1.1 route-map MED out
!
route-map MED permit 10
 set metric 25
!


R1#sh ip bgp
BGP table version is 9, local router ID is 10.1.255.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i192.168.111.0    10.1.255.3               0    100      0 30 ?
*>                  10.39.1.2                0             0 30 ?
*>i192.168.222.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
*>i192.168.223.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
R1#

R1#sh ip bgp
BGP table version is 24, local router ID is 10.1.255.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i192.168.111.0    10.1.255.3               0    100      0 30 ?
*                   10.39.1.2               25             0 30 ?
*>i192.168.222.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
*>i192.168.223.0    10.1.255.2               0    100      0 10 i
* i                  10.1.255.4               0    100      0 10 i
R1#
```

BGP Communities

- The community attribute is a transitive optional attribute

- BGP communities are a means of tagging (coloring) routes to ensure consistent filtering or route selection policy
- Any BGP router can tag routes in incoming and outgoing routing updates or when doing redistribution
- Any BGP router can filter routes in incoming or outgoing updates or select preferred routes based on communities
- By default, communities are stripped out in outgoing BGP updates (neighbor <IP> send-community is needed)
- There can be more than one BGP community attached to a single route

- Routers that do not support communities pass them along unchanged

BGP Communities

- The standards define several filtering-oriented communities
  - no-export – do not advertise routes to external eBGP peers, but is sent to iBGP peers and confederation neighbors, this is the most commonly used attribute
  - no-advertise – instructs neighbor do not advertise routes to any peer
  - local-as – do not advertise routes to any eBGP peer, or even confederation neighbor
  - Internet – advertise this route normally, default.

- Defining own communities
  - 32-bit value is split into two parts
    - High order 16-bit contains AS number of the AS that defines the community meaning
    - Low order 16-bit have local significance
    - [ASN]:[low-order-bits]

Administrative Policy Example

Communities to control the traffic

- 1:9001 Routes from UP ISP
- 1:9002 Routes from Peering
- 1:9003 Customer routes
- 1:9004 Default

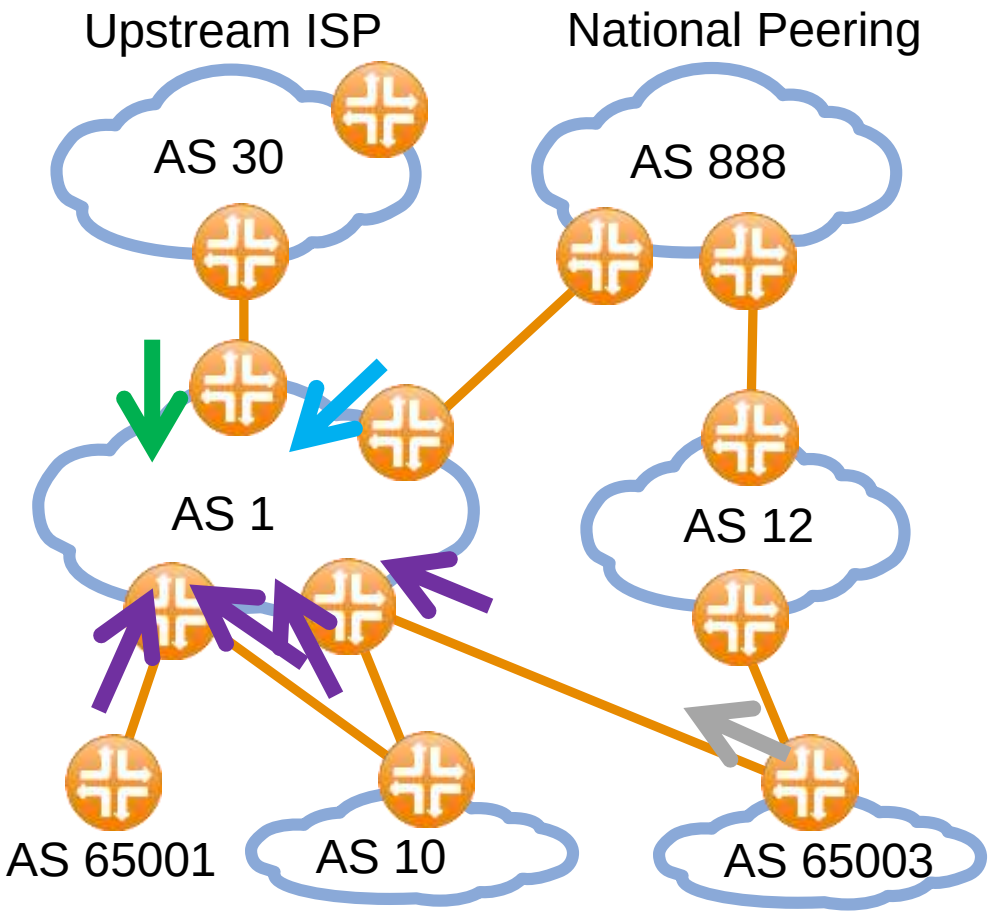
- 1:1101 Do not announce to peering
- 1:1102 Do not announce to UP ISP

- 1:1201 Prepend 1 1 to UP ISP
- 1:1202 Prepend 1 1 to Peering

Local Preference settings

- 1:50 Cust routes – glob backup

- 60 UP ISP
- 70 Peering
- 110 Customer routes – primary



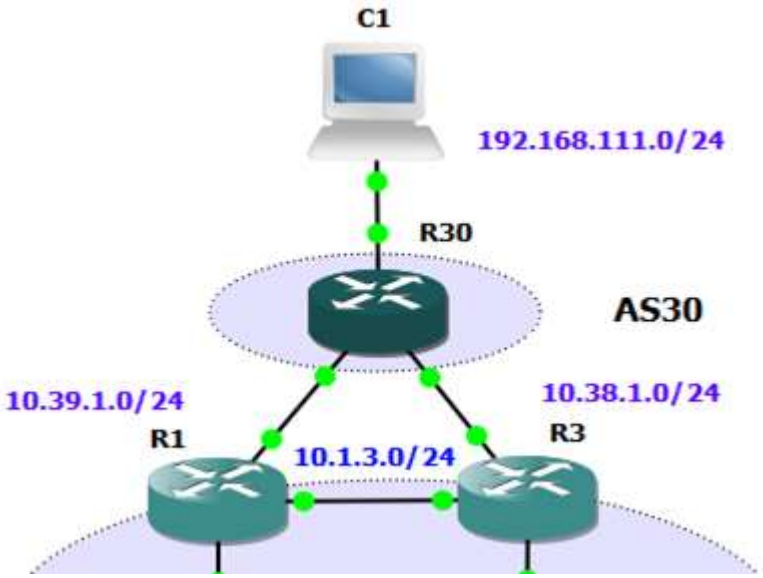
How to Design Communities

- Define administrative policy goals that needs to be implemented
- Define filters and route map selection policy that will achieve the required goals
- Assign community value to each goal
- Apply communities on incoming updates from neighboring ASes or ask the neighbors to set the communities
- Enable community propagation in your AS
- Match communities with route-maps and route filters - filter, change BGP attributes or influence route selection process

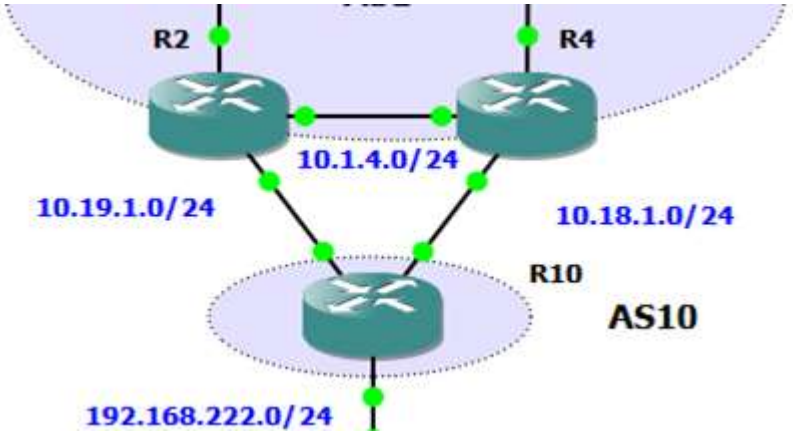
Community configuration

```
R1#
router bgp 1
 neighbor 10.39.1.2 route-map UP-ISP in
 neighbor 10.39.1.2 route-map PREPEND out
!
ip community-list 101 permit 1:1201
!
route-map PREPEND permit 10
 match community 101
 set as-path prepend 1 1
!
route-map PREPEND permit 20
!
route-map UP-ISP permit 10
 set local-preference 60
 set community 1:9001 additive

!
R1#sh ip bgp  192.168.111.0
BGP routing table entry for 192.168.111.0/24, version 8
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.39.1.2
  30
    10.39.1.2 from 10.39.1.2 (10.1.255.30)
      Origin incomplete, metric 0, localpref 60, valid, external
      Community: 1:9001
  30
    10.1.255.3 (metric 2) from 10.1.255.3 (10.1.255.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
```



Community configuration




```
R4#
router bgp 1
 neighbor 10.18.1.2 route-map Customer1 in
!
route-map Customer1 permit 10
 set local-preference 50
 set community 1:1201 1:9003 additive
!

R4#sh ip bgp 192.168.222.0
BGP routing table entry for 192.168.222.0/24, version 5
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.18.1.2
  10
    10.1.255.2 (metric 2) from 10.1.255.2 (10.1.255.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Community: 141:1201 141:9003
  10
    10.18.1.2 from 10.18.1.2 (10.1.255.10)
      Origin IGP, metric 0, localpref 50, valid, external
      Community: 1:1201 1:9003

R30#sh ip bgp
   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.111.0    0.0.0.0                  0         32768 ?
*  192.168.222.0    10.39.1.1                              0 1 1 1 10 i
*>                  10.38.1.1                              0 1 10 i
*  192.168.223.0    10.39.1.1                              0 1 1 1 10 i
*>                  10.38.1.1                              0 1 10 i
R30#
```

Community setting to Alter Local Preference

- 141:50 Customer routes – global backup
- When a customer wants to remove all traffic from a link
  - Why? Circuit Maintenance
- Two of attributes used to determine the path: local preference and AS path
  - The customer can prepend its AS number
  - Will not help if ISP has set up higher Local preference on the routes learned on the link
- The solution
  - ISP allows to the customer to send community string in order to setup proper Local Preference value

```
CUST3#
router bgp 65003
 neighbor 10.0.1.1 send-community
 neighbor 10.0.1.1 route-map Adv_to_AS141 out
!
 ip prefix-list PL_to_AS141 seq 10 perm 192.168.1.0/24
!
route-map Adv_to_AS141 permit 10
 match ip address prefix-list PL_to_AS141
 set as-path prepend 65003 65003 65003
 set community 1:50
```

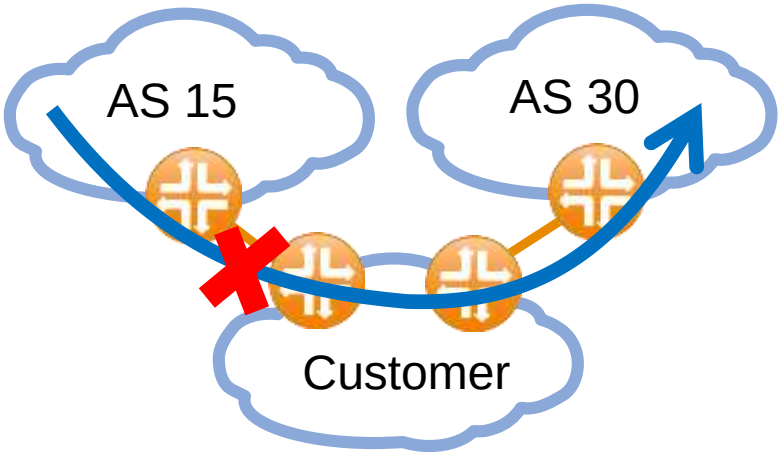
```
ISP1#
router bgp 141
 neighbor 10.0.1.2 route-map CUSTOMERS in
!
ip community-list 1 permit 141:50
ip community-list 1 deny
!
route-map CUSTOMERS permit 10
 match community 1
 set local-preference 50
!
route-map CUSTOMERS permit 20
 set local-preference 110
```

BGP Filters

Why filtering?

- Router performance
  - Propagation of limited group of prefixes (typically customer & peering) + default route
- Transit traffic issue
  - The multihomed customer / ISP could become transit AS for other service providers
  - Requirement - do not propagate provider routes to other providers
  - Solution - filter outgoing information to both ISPs

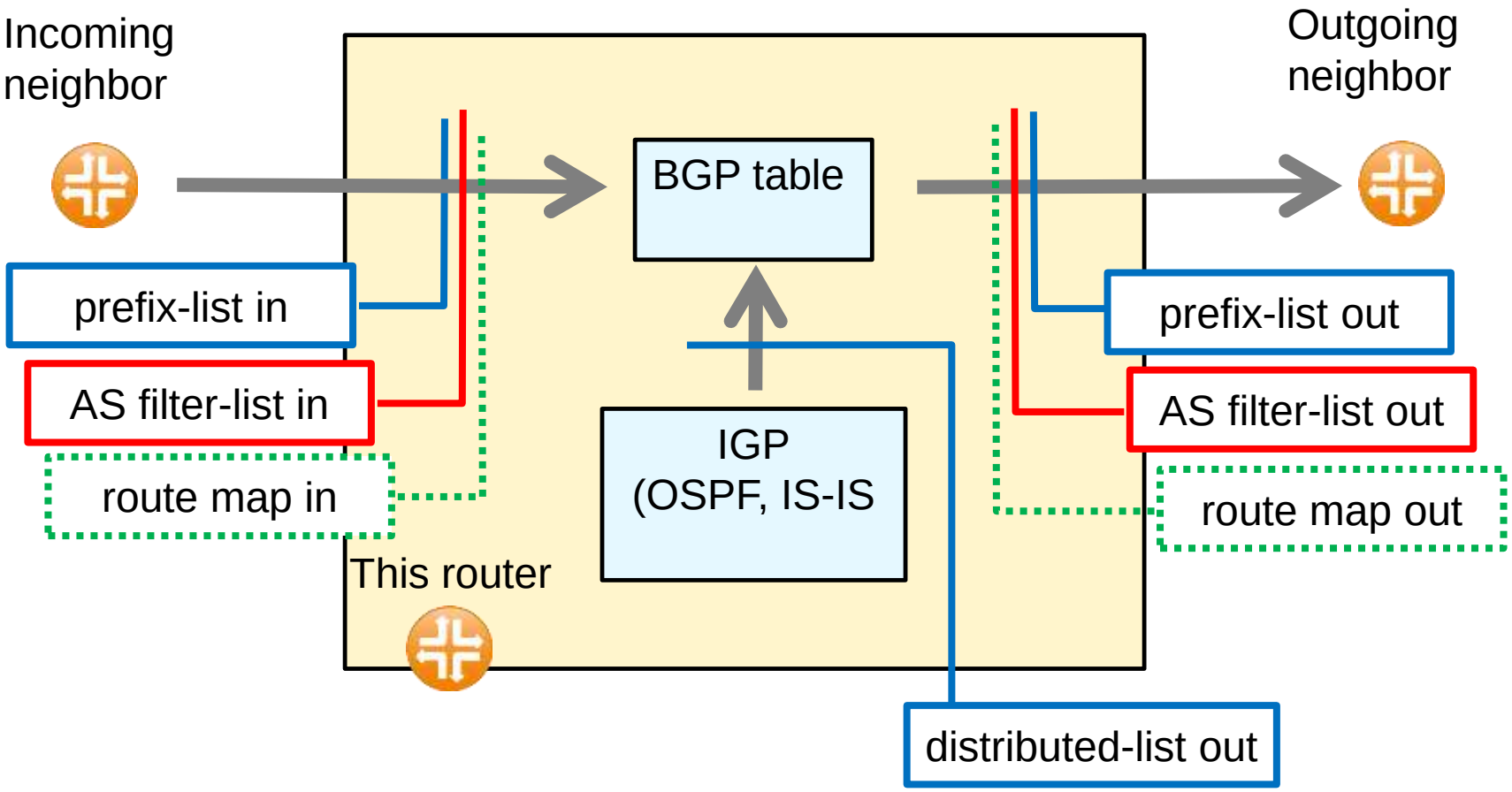
- **Prefix-list filters**
- **AS-PATH filters**
- **Community based filters**



Attribute modification & filtering

- **Route Maps**
  - Allow complex filters and attribute modifications

BGP Filters & Route map Implementation



Prefix-list Filters Examples

- Exact match
  - prefix-list xx seq 10 permit 10.134.12.0/22
  - prefix-list xx seq 20 permit 10.134.16.0/21
  - prefix-list xx seq 30 permit 10.134.24.0/24

- Prefix list entry with “le” and “ge” statement matches prefix within specified space where the subnet mask falls within specified limits

- Any route in the range of 63.1.0.0 - 63.1.255.255 having the mask longer or equal to 24
  - prefix-list xx seq 10 permit 63.1.0.0/16 **ge** 24
  - 63.1.10.0/24, 63.20.10.4/30, ..

- Question?
  - prefix-list xx seq 10 permit 63.1.0.0/16 **le** 25

neighbor {*ip-address* } **prefix-list** *prefix-list-name* {**in** | **out**}

AS Path filters

- Route filtering based on AS-Path
  - Announce only local routes to the ISP – AS-Path empty
  - Select routes based on a specific AS Number
  - Accept for specific AS only from some BGP neighbors

- AS-Path use **regular expressions**
  - A regular expression is a pattern to match against an input string
    - **`ip as-path access-list 1 permit 215`**
    - **`show ip bgp regexp 215`**
  - the regular expression, 215, will match not only AS 215 but also 12150, 21502, 21503, etc

  - neighbor <IP> filter-list <xy> in/out

AS-Path - Regular Expressions

*****	matches zero or more atoms
?	Matches zero or one atom
+	matches one or more atoms
_	matches any delimiter (white space, tab, comma, etc)
$	matches end of AS-Path
^	matches beginning of AS-Path
.	Any single character (except ^ and $)
[ ]	A range of character
|	expr1|expr2 logical OR
( )	to group smaller expression into large ones
****	to remove special characters

^21, 31$, _31_	|213 317 218 31 731|
[1-3].[34]	|213 317 2316 31| |213 317 2316 31|
21|31	|213 317 2316 31|
(213|218)_31	|213 317 1218 316 31|
_23(_78)?_45_	|23 78 45 56 23 45 333|
^\(213_	|(213 317) 1218 316 31|

AS-Path - Regular Expressions Quiz

A	_100_
B	^100$
C	_100$
D	^100_.
E	^[0-9]+$
F	^$
G	.*

1	Originated in AS 100
2	Matches everything
3	Networks behind AS 100
4	Networks originated in local AS
5	Going through AS 100
6	AS path one AS long
7	Our network is directly connected into the AS100

Route Map

- Default statement action is “permit”
- Route not matched by any statement is dropped
- “Permit all” is achieved by specifying “permit” without “match”
- More match conditions in one statements are ANDed
- First matching statement permits or denies the route
- Both prefix and AS lists can be used within Route map

```
CUST3#
router bgp 65003
 neighbor 10.0.1.1 send-community
 neighbor 10.0.1.1 route-map Adv_to_AS141 out
!
 ip prefix-list PL_to_AS141 seq 10 perm 192.168.1.0/24
!
route-map Adv_to_AS141 permit 10
 match ip address prefix-list PL_to_AS141
 set as-path prepend 65003 65003 65003
 set community 1:50
```

```
ISP1#
router bgp 141
 neighbor 10.0.1.2 route-map CUSTOMERS in
!
ip community-list 1 permit 141:50
ip community-list 1 deny
!
route-map CUSTOMERS permit 10
 match community 1
 set local-preference 50
!
route-map CUSTOMERS permit 20
 set local-preference 110
```

Ďakujem za pozornosť

roman dot kaloc at gmail dot com