The Internet Ecosystem and Evolution

Lab 5

Inter-domain routing configuration: Valley-free routing

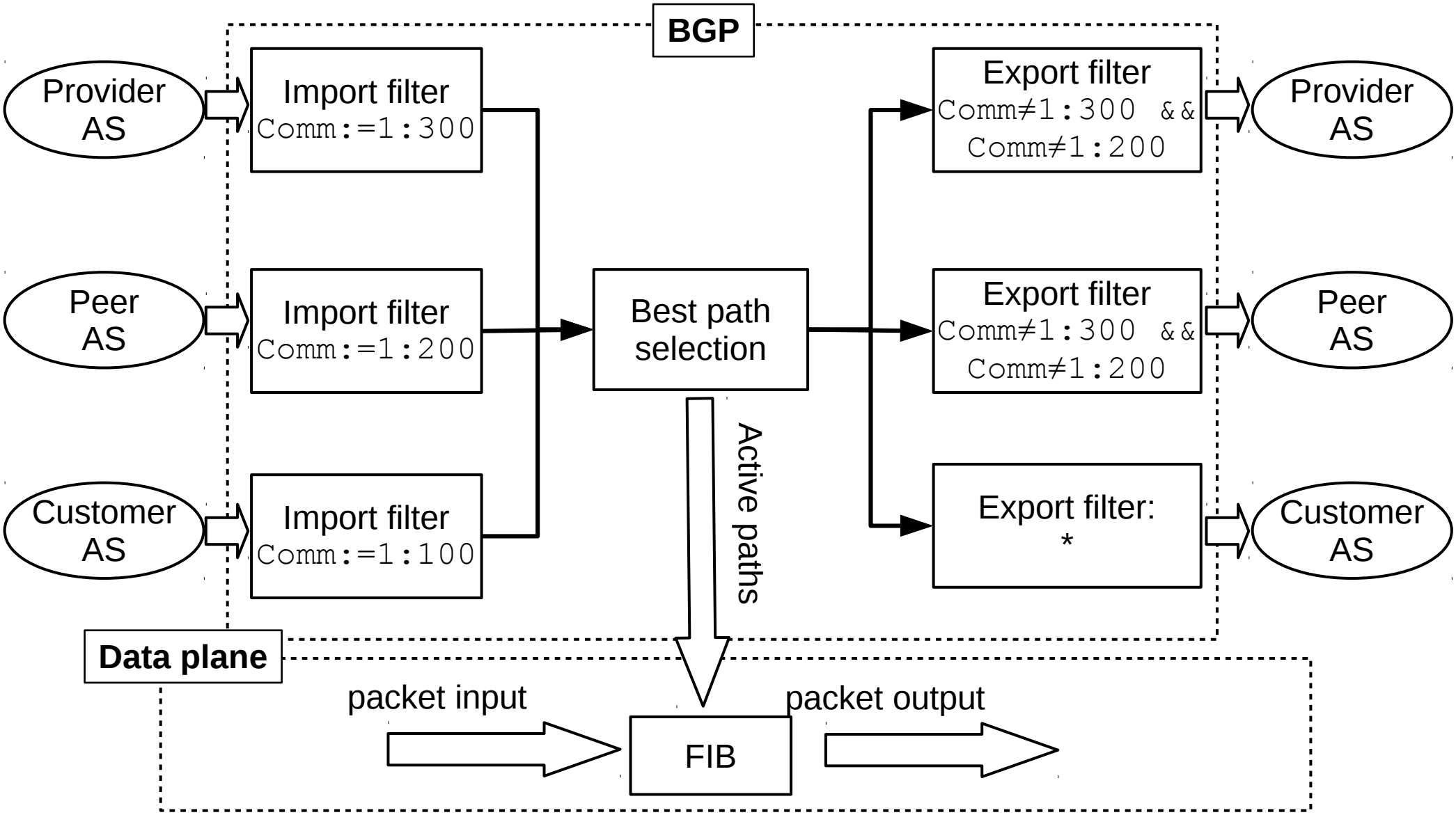
Recall: policy routing

- ASes can have either transit/peer relationship
  - **transit:** global Internet-access for transit fee
  - **peer:** mutual traffic exchange between two ASes and their customers
- Traffic follows the cash-flow
- Feasible/prohibited paths: **valley-free routing**
- If more than one valley-free paths to a prefix
  - **prefer-customer**: customer paths for free
  - then **shortest AS-path** (next lab)

Configuring BGP filters

- **Valley-free routing:** announcements received from providers are marked by community `1:300`, those from peers with community `1:200`, and from customers with `1:100` on **import filters**
- Drop announcements to providers and peers that contain the community `1:300` or `1:200` on **export filters**

BGP: valley-free



BGP import filters

- Same filters at all BGP routers
- Enough to configure only once at each router

```
!! Import filter for providers
route-map rm-provider-in permit 10
 set community 1:300

!! Import filter for peers
route-map rm-peer-in permit 10
 set community 1:200

!! Import filter for customers
route-map rm-customer-in permit 10
 set community 1:100
```

BGP export filters

- The below works on Cisco and Quagga as well

```
ip community-list standard cm-no-export permit 1:200
ip community-list standard cm-no-export permit 1:300

route-map rm-no-export deny 10
 match community cm-no-export

route-map rm-no-export permit 20
```

- List the `community` values to be dropped, let the rest through
- **The art of BGP configuration**: attach the right filter to the right neighbor

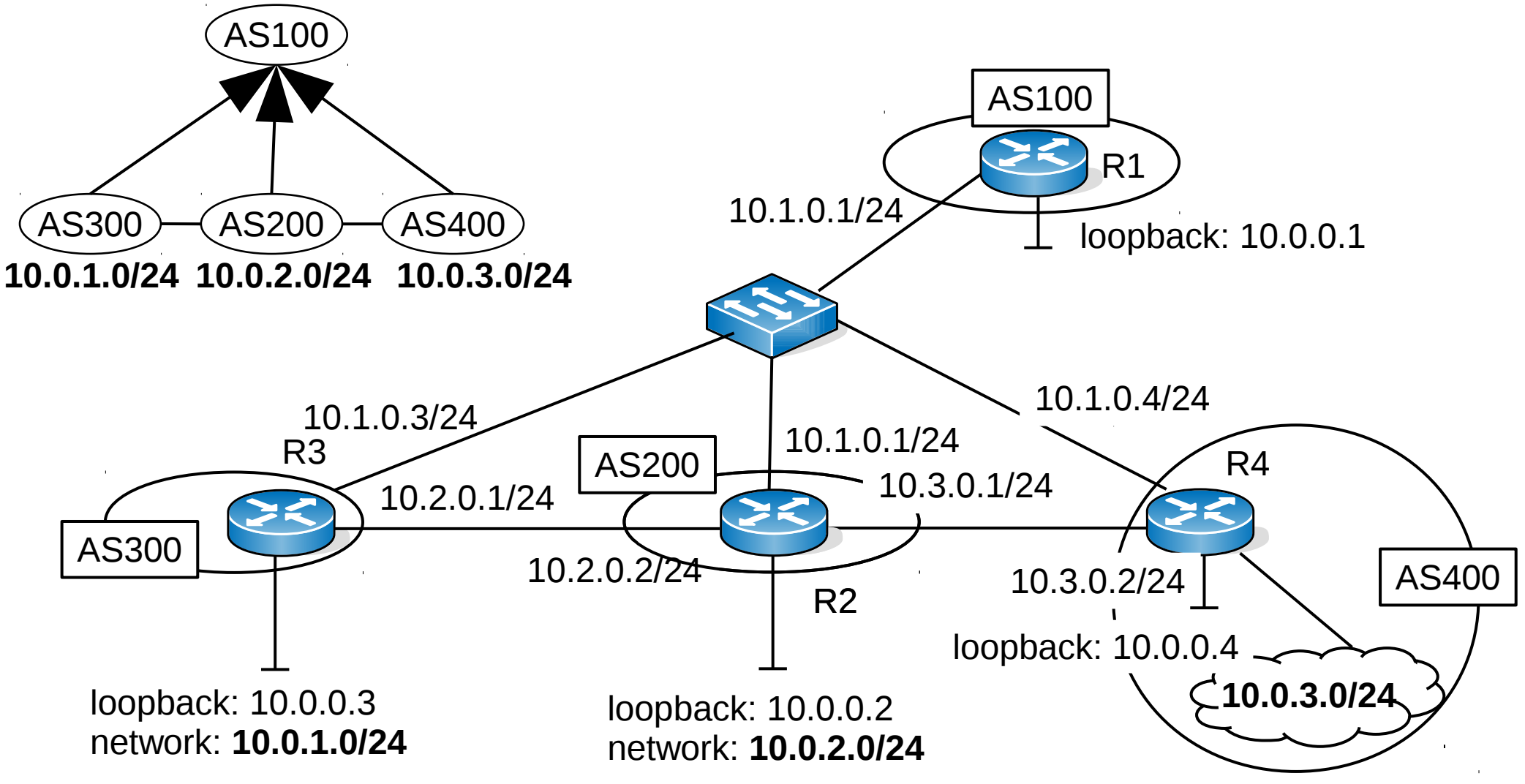
Configure BGP neighbors

```
router bgp X
 bgp router-id ...
 network ...
!! Provider neighbor config
 neighbor ... remote-as ...
 neighbor ... route-map rm-provider-in in
 neighbor ... route-map rm-no-export out
!! Peer neighbor config
 neighbor ... remote-as ...
 neighbor ... route-map rm-peer-in in
 neighbor ... route-map rm-no-export out
!! Customer neighbor config
 neighbor ... remote-as ...
 neighbor ... route-map rm-customer-in in
```

- Customers don't need export filters

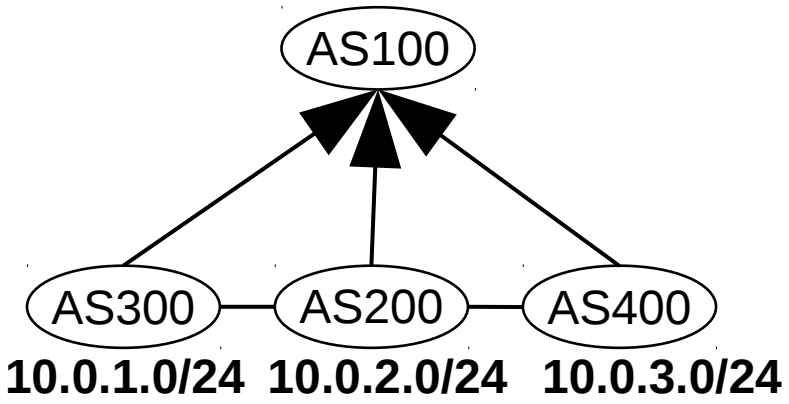
Exercise

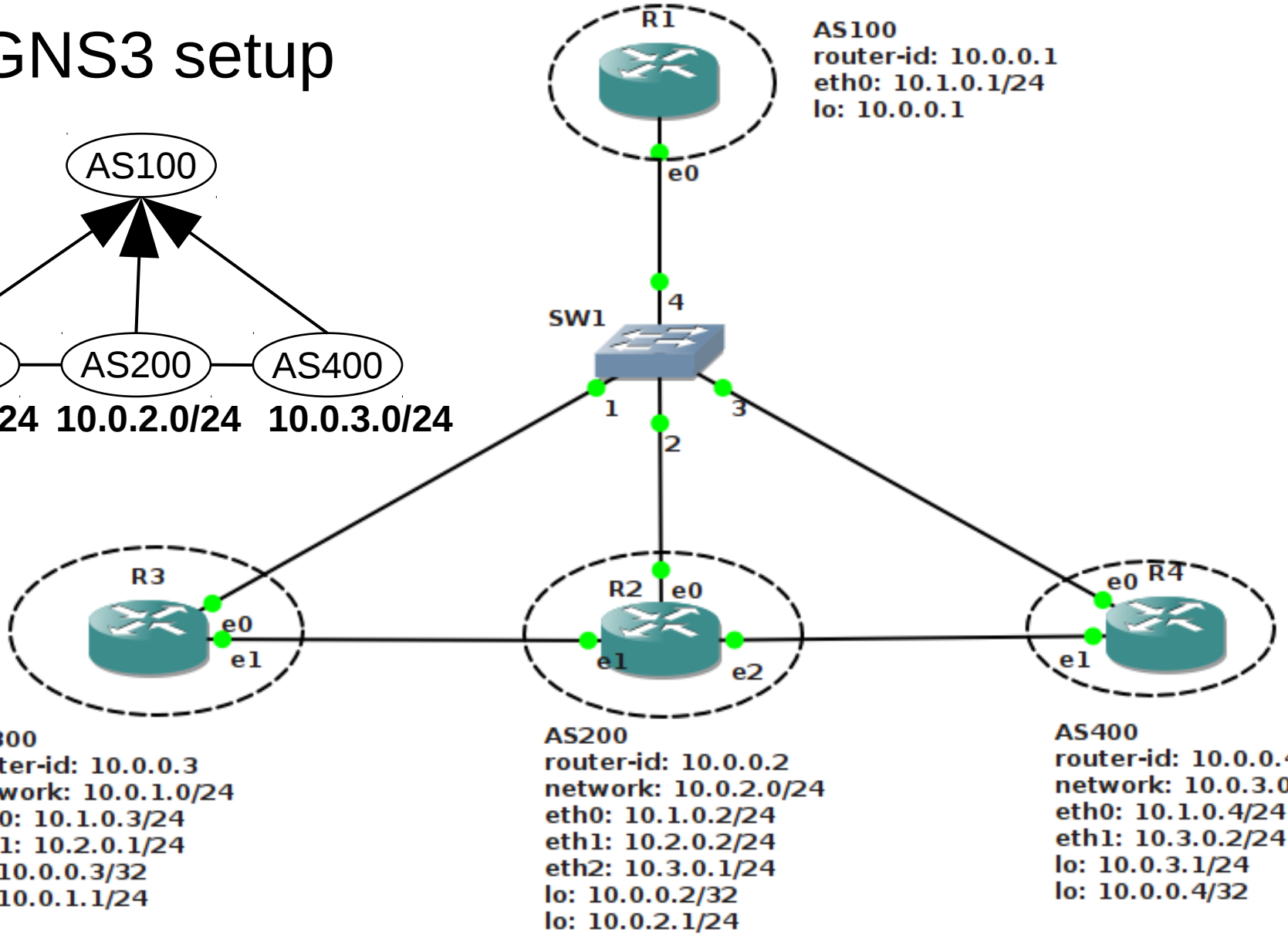
- Configure the below AS-hierarchy



Exercise

- GNS3 setup





Exercise

- `R2` base configuration (w/o import/export filters)

```
interface eth0
 ip address 10.1.0.2/24

interface eth1
 ip address 10.2.0.2/24
!
interface eth2
 ip address 10.3.0.1/24

interface lo
 ip address 10.0.0.2/32
 ip address 10.0.2.1/24

router bgp 200
 bgp router-id 10.0.0.2
 network 10.0.2.0/24
 neighbor 10.1.0.1 remote-as 100
 neighbor 10.2.0.1 remote-as 300
 neighbor 10.3.0.2 remote-as 400
```

Exercise

- Checking BGP neighbor status on `R2`

```
OpenWrt# show ip bgp summary
BGP router identifier 10.0.0.2, local AS number 200
RIB entries 5, using 360 bytes of memory
Peers 3, using 7584 bytes of memory

Neighbor V  AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down  State/PfxRcd
10.1.0.1 4 100       7      11      0   0    0 00:04:48          2
10.2.0.1 4 300       7      10      0   0    0 00:04:45          2
10.3.0.2 4 400       8      10      0   0    0 00:04:43          1

Total number of neighbors 3
```

- If we want to check a specific neighbor

```
OpenWrt# show ip bgp neighbors 10.1.0.1
BGP neighbor is 10.1.0.1, remote AS 100, local AS 200, external link
  BGP version 4, remote router ID 10.0.0.1
  BGP state = Established, up for 00:07:40
...
```

Exercise

- BGP RIB on `R4`

```
OpenWrt# show ip bgp
BGP table version is 0, local router ID is 10.0.0.4
Status codes: s suppressed, d damped, h history,
              * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network      Next Hop Metric LocPrf Weight Path
*  10.0.1.0/24 10.1.0.3                    0 100 300 i
*>             10.3.0.1                    0 200 300 i
*> 10.0.2.0/24 10.3.0.1      0             0 200 i
*              10.1.0.2                    0 100 200 i
*> 10.0.3.0/24 0.0.0.0       0         32768 i

Total number of prefixes 3
```

- Only to advertised prefixes: no route to `AS100`
- The active path `400` → `200` → `300` is **not valley-free!**

Exercise

- Configure valley-free routing:
  - define import filters as a `route-map`
  - define `community-lists` and export filters
  - attach the right import/export filters to the adequate BGP neighbors
  - warning: if you mistype filter parameters or IP addresses, BGP may silently drop all announcements → empty BGP RIB
- Do the filter config first only at `R2`

Exercise

- All prohibited paths have disappeared at `R4`

```
OpenWrt# show ip bgp
BGP table version is 0, local router ID is 10.0.0.4
Status codes: s suppressed, d damped, h history,
              * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network      Next Hop Metric LocPrf Weight Path
*> 10.0.1.0/24 10.1.0.3                    0 100 300 i
*  10.0.2.0/24 10.1.0.2                    0 100 200 i
*>             10.3.0.1      0            0 200 i
*> 10.0.3.0/24 0.0.0.0       0        32768 i

Total number of prefixes 3
```

- Worth doing filter configuration at the rest of the routers as well