The Internet Ecosystem and Evolution

Lab 6

Inter-domain routing configuration: Advanced policy routing

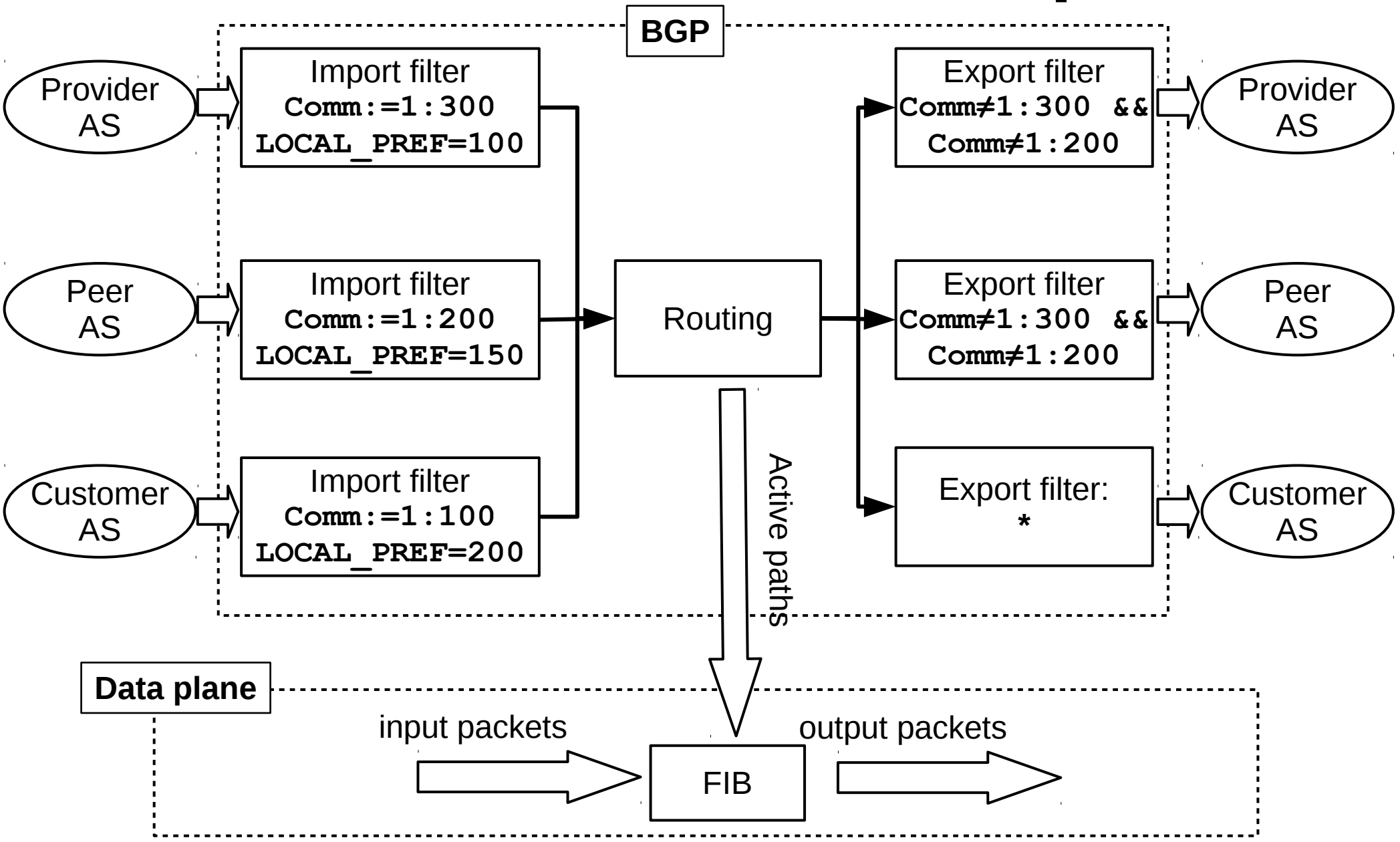
Recall: policy routing

- ASes can have either transit/peer relationship
  - **transit:** global Internet-access for transit fee
  - **peer:** mutual traffic exchange between two ASes and their customers
- 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**
- Fine-tune BGP import/export filters

BGP filters

- **Valley-free routing:** tag announcements received from providers by community `1:300`, from peers by community `1:200`, and from customers by `1:100` at properly configured **import filters**
- Discard announcements towards providers and peers that contain either community `1:300` OR `1:200` at **export filters**
- **Prefer-customer:** use the LOCAL_PREF attribute
  - customer: 200, peer: 150, provider: 100
  - path with highest LOCAL_PREF takes preference
- **Shortest AS-path:** automatic

Valley-free+prefer-customer+shortest-AS-path



Valley-free+prefer-customer+shortest-AS-path

- Similar filters at all BGP routers

```
!! Import filter for providers
route-map rm-provider-in permit 10
 set community 1:300
 set local-preference 100

!! Import filter for peers
route-map rm-peer-in permit 10
 set community 1:200
 set local-preference 150

!! Import filter for customers
route-map rm-customer-in permit 10
 set community 1:100
 set local-preference 200
```

Valley-free+prefer-customer+shortest-AS-path

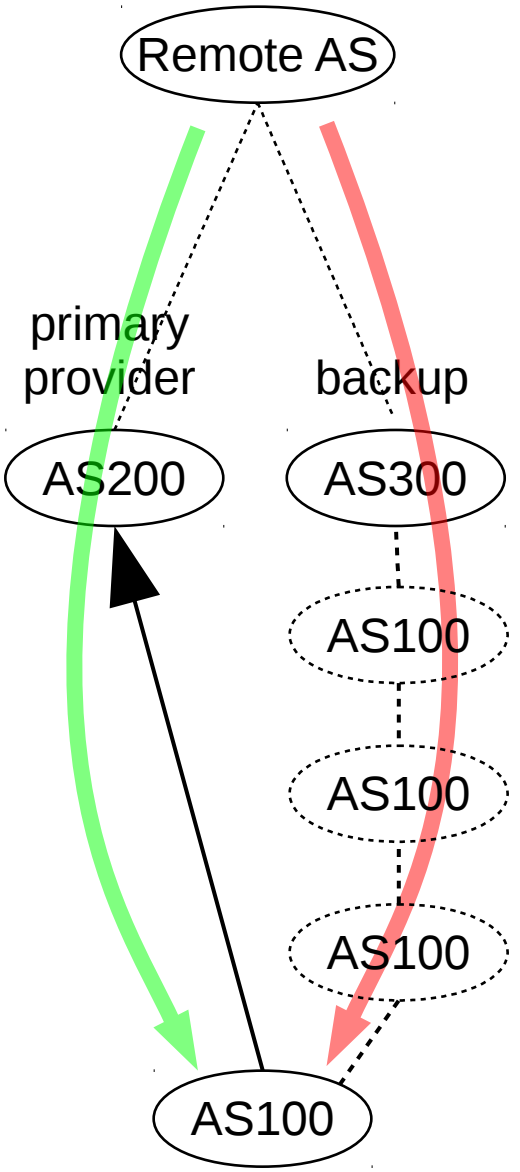
```
router bgp X
 bgp router-id ...
 network ...
!! Config for neighbors that are providers
 neighbor ... remote-as ...
 neighbor ... route-map rm-provider-in in
 neighbor ... route-map rm-no-export out
!! Config for neighbors that are peers
 neighbor ... remote-as ...
 neighbor ... route-map rm-peer-in in
 neighbor ... route-map rm-no-export out
!! Config for neighbors that are customers
 neighbor ... remote-as ...
 neighbor ... route-map rm-customer-in in
```

- Attach the right filter to the right neighbor!

Backup route: AS path prepending

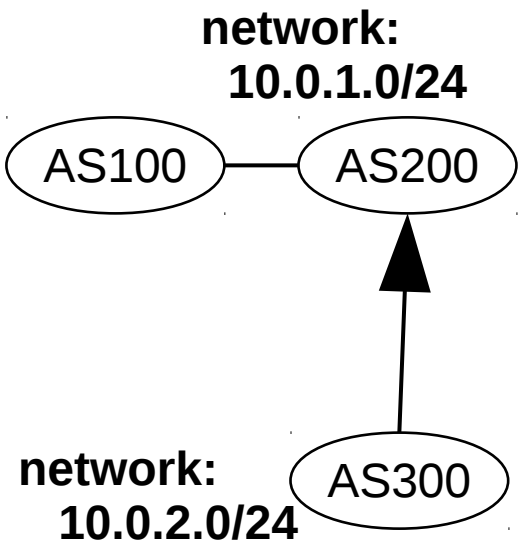
- Let ingress traffic pass the primary provider (egress: LOCAL_PREF)
- **AS-path prepending:** ingress path via the a backup “looks” longer
- Only one `route-map out` can be active for a `neighbor` at a time!

```
router bgp 100
 ...
 neighbor X remote-as 300
 neighbor X route-map rm-as-prepend out
 ...
!!! AS-path prepending filter
route-map rm-as-prepend permit 10
  set as-path prepend 100 100 100
```



Prefix filtering

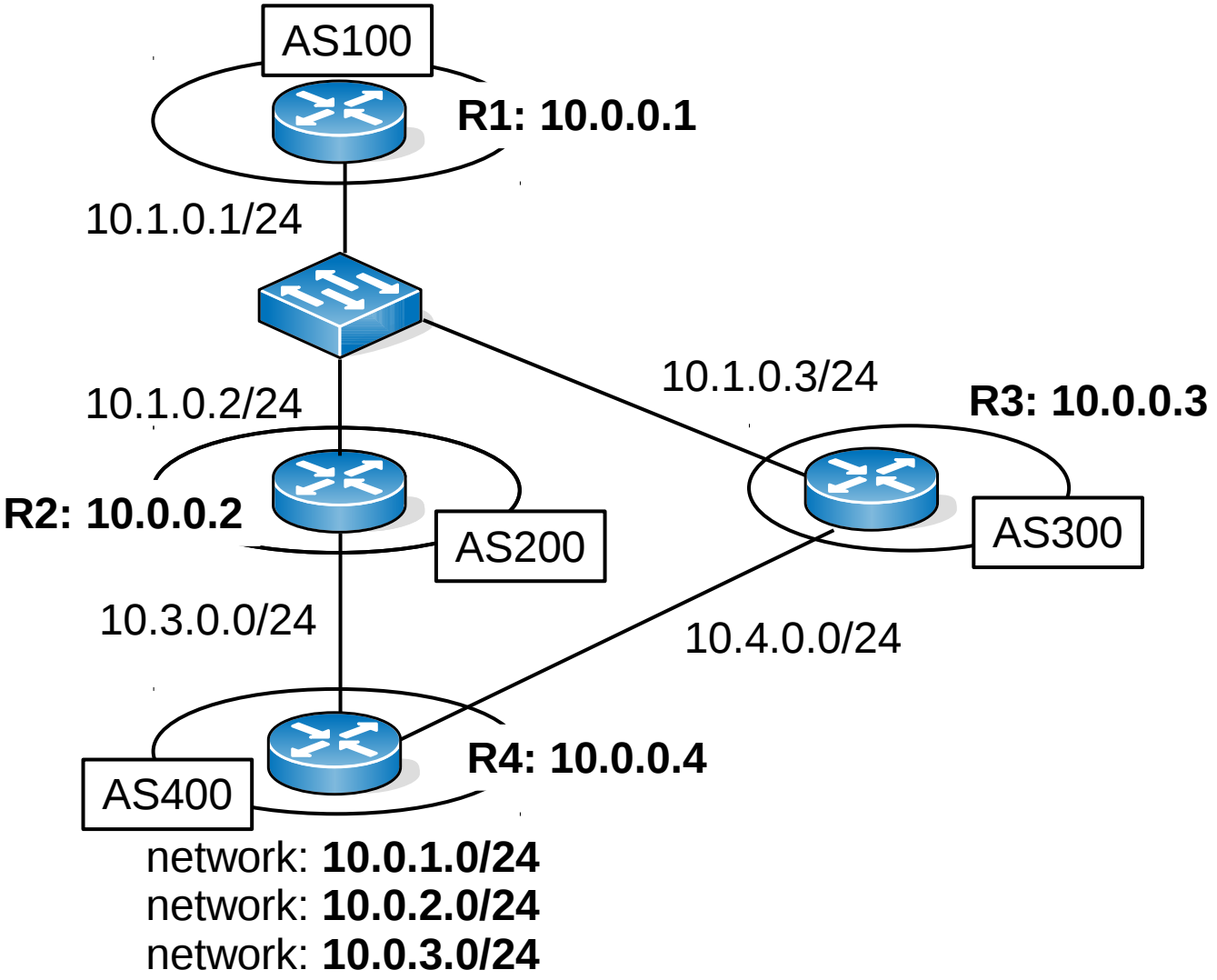
- Declare prefixes accepted from a neighbor
- Accepted prefixes: `permit`, rest: `deny`
- A `route-map` and a `prefix-list` *can* be active for a `neighbor` in the same direction (`in`/`out`)

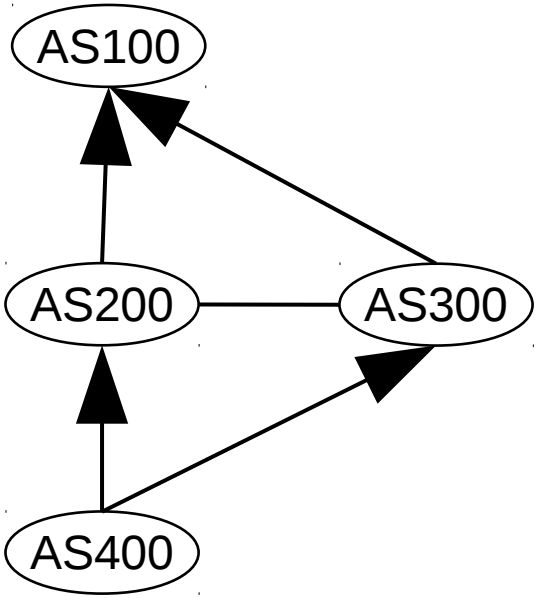



```
router bgp 100
 ...
 neighbor X remote-as 200
 neighbor X route-map rm-provider in
 neighbor X prefix-list AS200 in
 ...
!!! Filter for prefixes announced by AS200
ip prefix-list AS200 seq 5 permit 10.0.1.0/24
ip prefix-list AS200 seq 10 permit 10.0.2.0/24
ip prefix-list AS200 seq 15 deny 0.0.0.0/0 le 32
```

Exercise

- Configure the below AS hierarchy

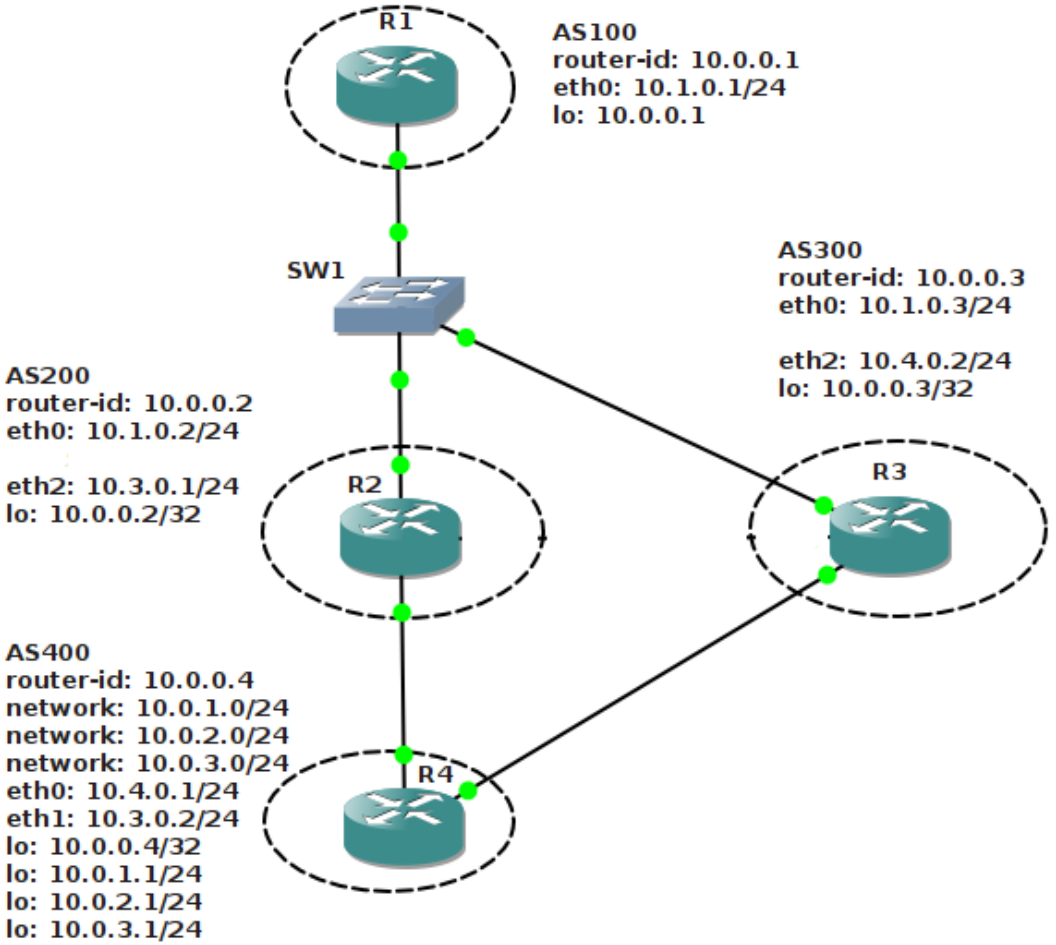


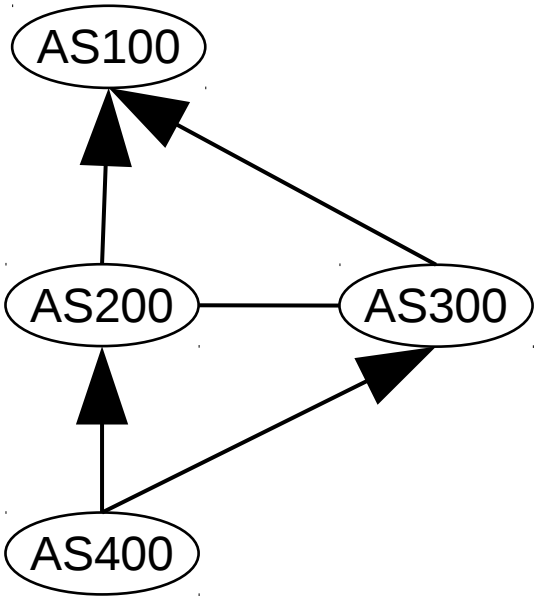


10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

Exercise

- Let `AS200` be the primary provider of `AS400`
- Configure the valley-free+prefer-customer rule





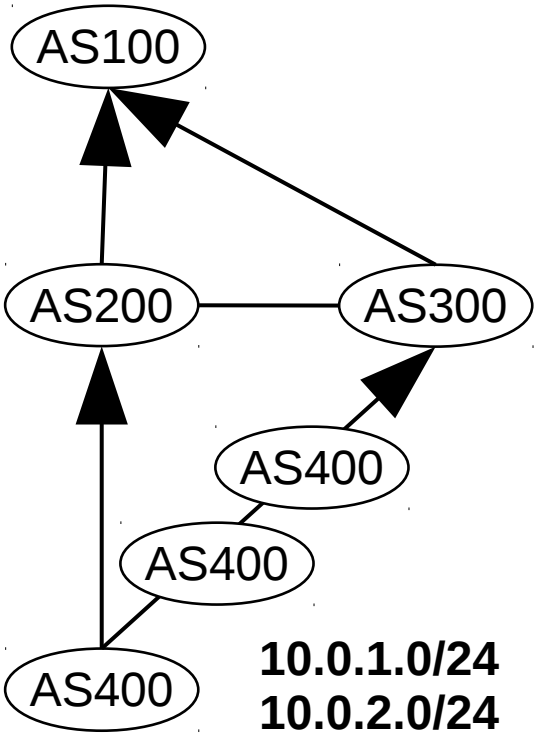
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

Exercise

- Let the primary provider of `AS400` be `AS200`: AS path prepending towards `AS300` at `AS400`
- Should somehow combine two `route-maps`

```
!!! INVALID CONFIGURATION
router bgp 400
 ...
 neighbor 10.4.0.2 remote-as 300
 neighbor 10.4.0.2 route-map rm-provider-in in
 neighbor 10.4.0.2 route-map rm-as-prepend out
 neighbor 10.4.0.2 route-map rm-no-export out
 ...

route-map rm-as-prepend permit 10
  set as-path prepend 400 400
route-map rm-no-export deny 10
 match community cm-no-export
route-map rm-no-export permit 20
```

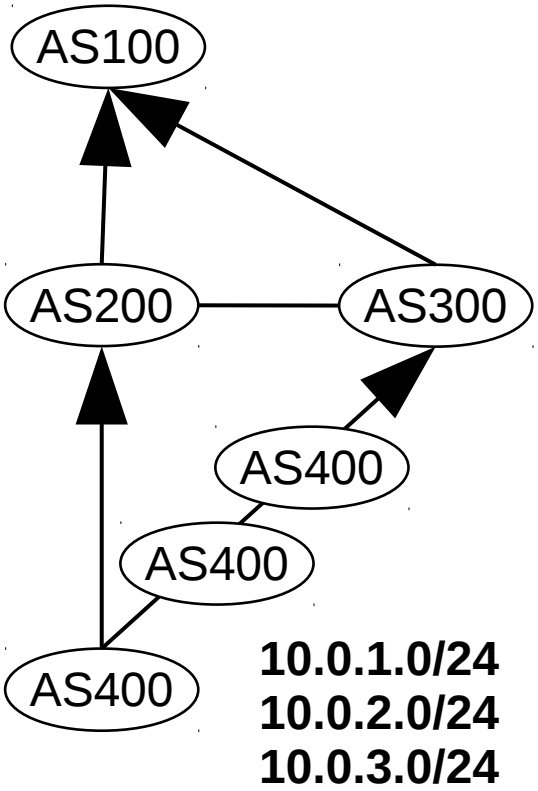


Exercise

- Create a specific `route-map` for each neighbor
- Can use the usual `route-map rm-no-export` for the neighbor `AS200` (nothing special needed)

```
!!! VALID BGP CONFIGURATION
router bgp 400
 ...
 neighbor 10.4.0.2 remote-as 300
 neighbor 10.4.0.2 route-map rm-provider-in in
 neighbor 10.4.0.2 route-map rm-AS300-out out
 ...

route-map rm-AS300-out deny 10
 match community cm-no-export
route-map rm-AS300-out permit 20
  set as-path prepend 400 400
```



Exercise

- AS300 now prefers the peer AS-path (via AS200) over the customer path (due to the prepending!)

```
R3# show ip bgp
   Network      Next Hop Metric LocPrf Weight Path
*  10.0.1.0/24 10.1.0.2                   0 100 200 400 i
*>             10.2.0.2                   0 200 400 i
*              10.4.0.1      0            0 400 400 400 I
```

- We haven't configured prefer-customer at R3 yet!

```
router bgp 300
 ...
 neighbor 10.4.0.1 remote-as 400
 neighbor 10.4.0.1 route-map rm-customer-in in

!!! Set LOCAL_PREF to 200: customer path are now preferred
route-map rm-customer-in permit 10
 set community 1:100
 set local-preference 200
```

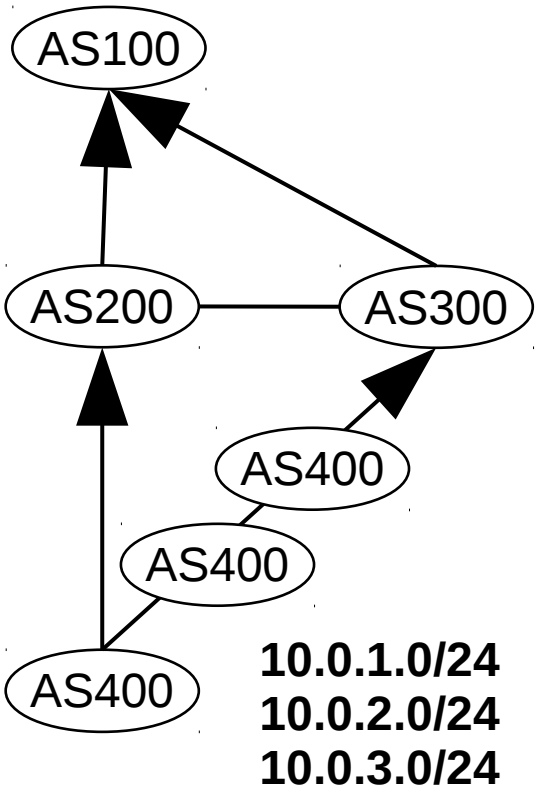
Exercise

- After configuring “prefer-customer”
- AS300 prefers the customer path indeed

```
R3# show ip bgp
   Network          Next Hop  Path
*  10.0.1.0/24 10.1.0.2  100 200 400 i
*              10.2.0.2  200 400 i
*>             10.4.0.1  400 400 400 i
```

- AS100 now uses the primary provider to reach AS400

```
R1# show ip bgp
   Network          Next Hop  Path
*  10.0.1.0/24 10.1.0.3  300 400 400 400 i
*>             10.1.0.2  200 400 i
...
```

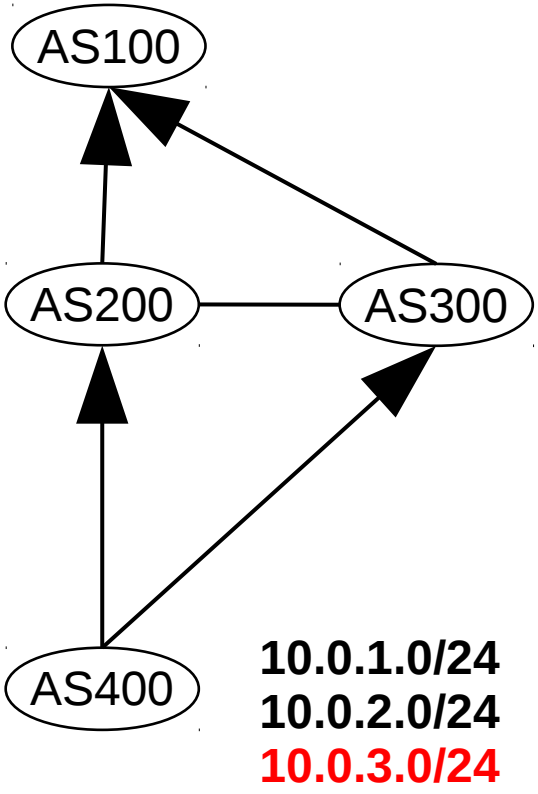


Exercise

- Suppose now that AS400 "owns" 10.0.1.0/24 and 10.0.2.0/24 but not 10.0.3.0/24
- Filter prefixes at R3 to reject 10.0.3.0/24 from AS400

```
router bgp 300
 ...
 neighbor 10.4.0.1 remote-as 400
 neighbor 10.4.0.1 prefix-list AS400 in
 neighbor 10.4.0.1 route-map rm-customer-in in

ip prefix-list AS400 seq 5 permit 10.0.1.0/24
ip prefix-list AS400 seq 10 permit 10.0.2.0/24
ip prefix-list AS400 seq 15 deny 0.0.0.0/0 le 32
```

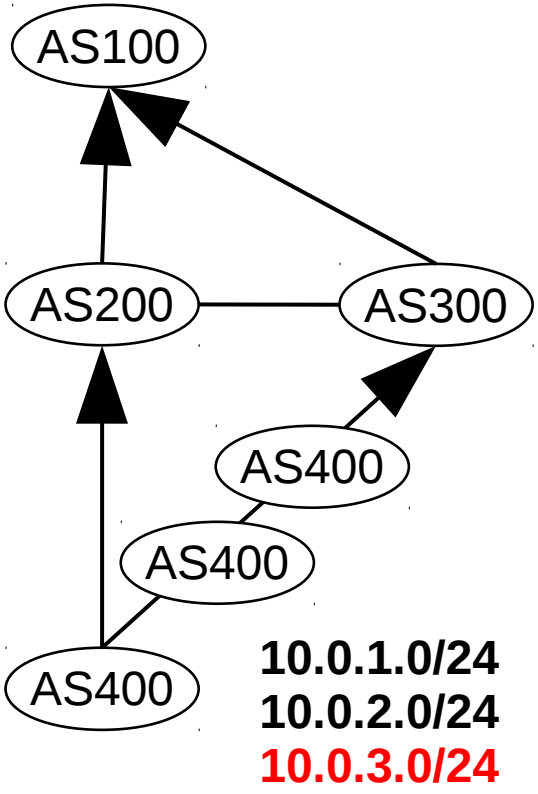


Exercise

- Of course, `AS300` needs to filter prefixes for other neighbors too
- Otherwise, hijacked prefixes still received from other neighbors

```
R3# show  ip bgp
   Network      Next Hop   Path
[…]
*  10.0.3.0/24 10.1.0.2   100 200 400 i
*>             10.2.0.2   200 400 i
```

- If an AS does not apply prefix filtering correctly: leaks hijacked prefixes to the rest of the Internet
- Collateral damage



Exercise

1. Configure the below AS-hierarchy and configure valley-free routing! Which paths appear in the BGP RIB at `AS300`?
2. Set the AS-path prepending as depicted in the figure! Which one is the preferred path at `AS300` towards prefix `10.0.2.0/24`?
3. Set the prefer-customer+shortest AS-path policy and revisit the AS paths `AS300`! What changed?
4. Set prefix filtering to reject `10.0.3.0/24` at `AS300`! Could `AS300` completely filter the prefix? How to remove `10.0.3.0/24` from the AS hierarchy entirely?