The Internet Ecosystem and Evolution

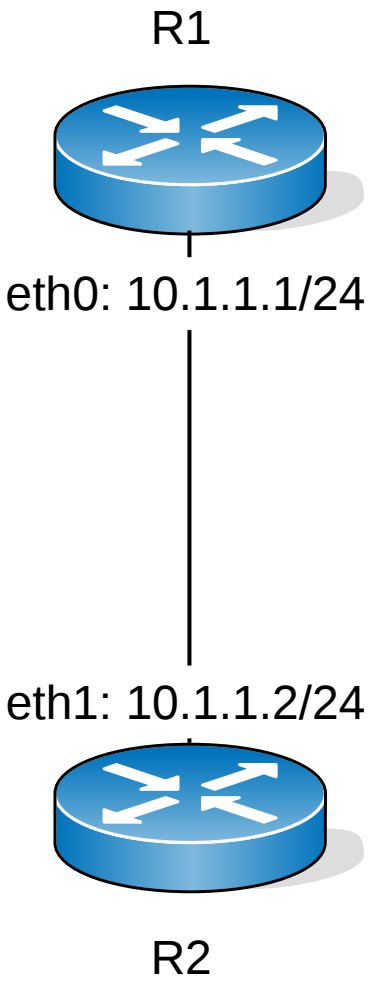
Lab 4

Inter-domain routing configuration: The basics

Reminder: Interface configuration

- Configuring link `R1–R2` at router `R1`

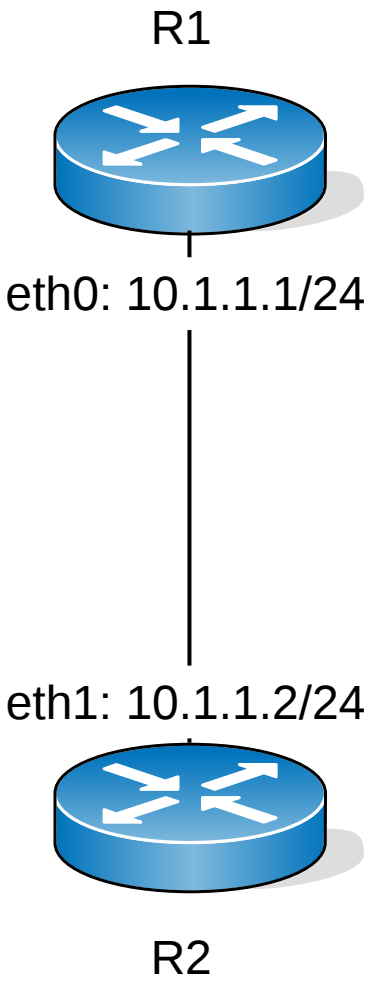
```
root@OpenWrt:/# vtysh
OpenWrt# configure terminal
OpenWrt(config)# interface eth0
OpenWrt(config-if)# ip address 10.1.1.1/24
OpenWrt(config-if)# no shutdown
OpenWrt(config-if)# exit
OpenWrt(config)# exit
OpenWrt# write terminal
[…]
interface eth0
 ip address 10.1.1.1/24
 ipv6 nd suppress-ra
[…]
OpenWrt# write file
OpenWrt# ping 10.1.1.2
```



Reminder: Interface configuration

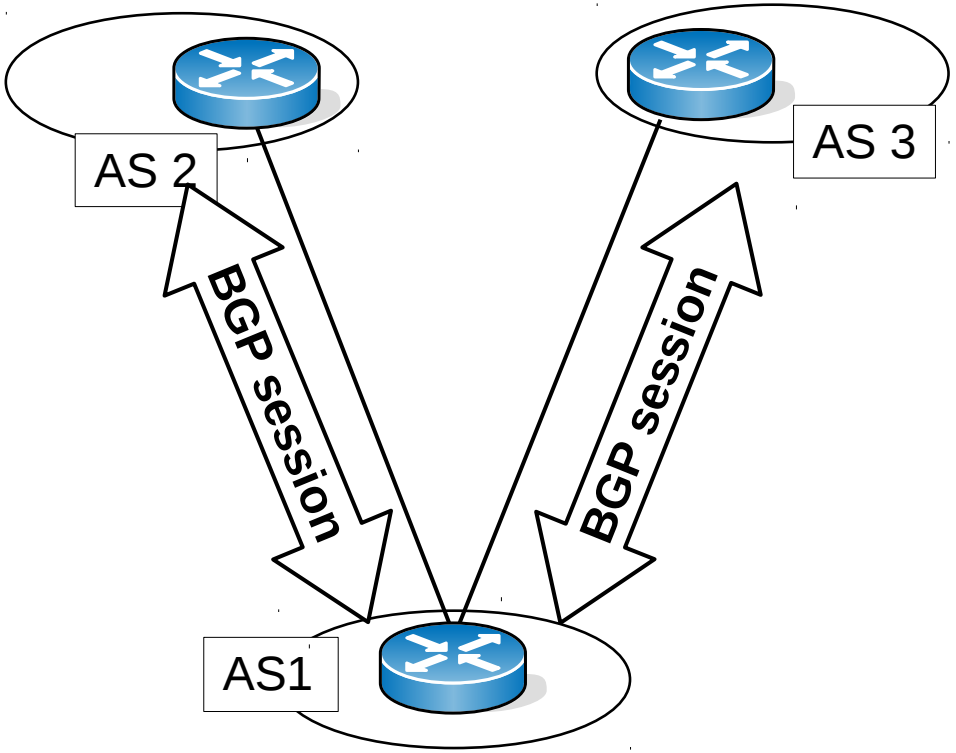
- Configuring link `R1–R2` at router `R2`

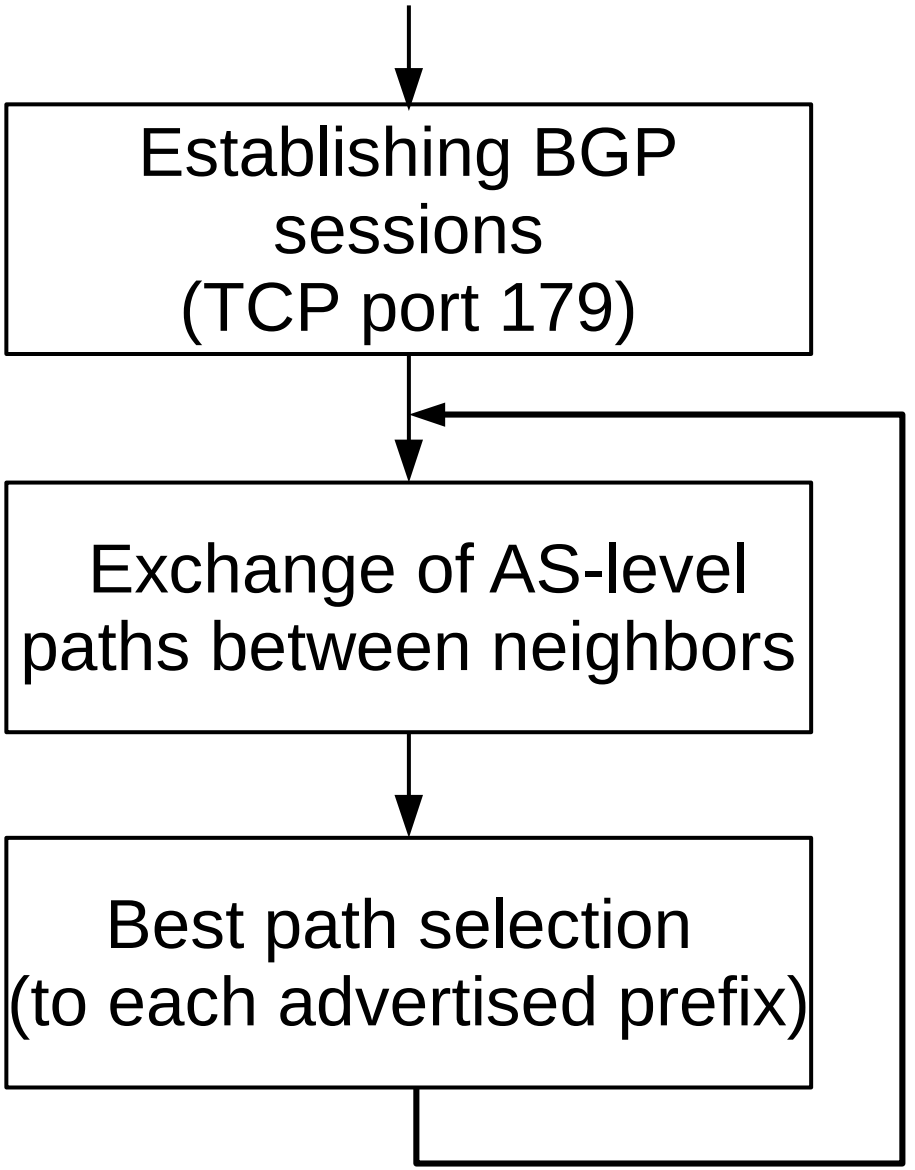
```
root@OpenWrt:/# vtysh
OpenWrt# configure terminal
OpenWrt(config)# interface eth1
OpenWrt(config-if)# ip address 10.1.1.2/24
OpenWrt(config-if)# no shutdown
OpenWrt(config-if)# exit
OpenWrt(config)# exit
OpenWrt# write terminal
[…]
interface eth1
 ip address 10.1.1.2/24
 ipv6 nd suppress-ra
[…]
OpenWrt# write file
OpenWrt# ping 10.1.1.1
```



BGP: Process model

- Neighboring routers establish a **BGP session** between each other



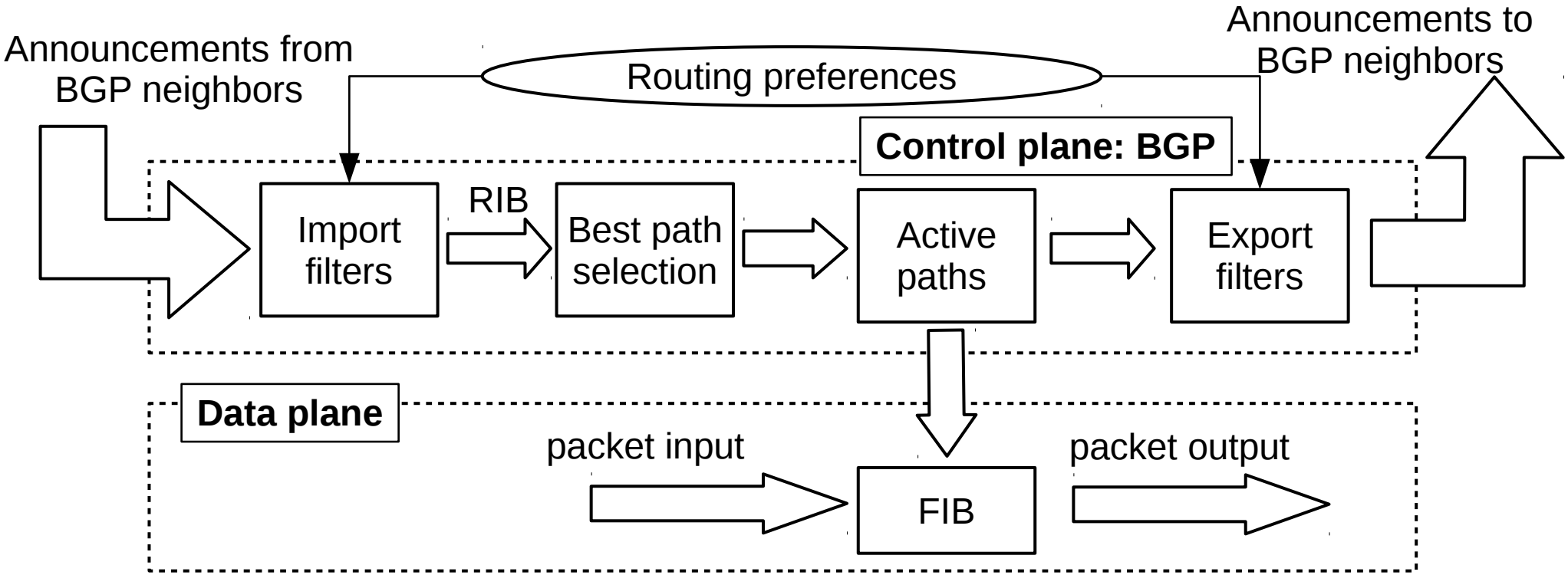


BGP: a path-vector EGP

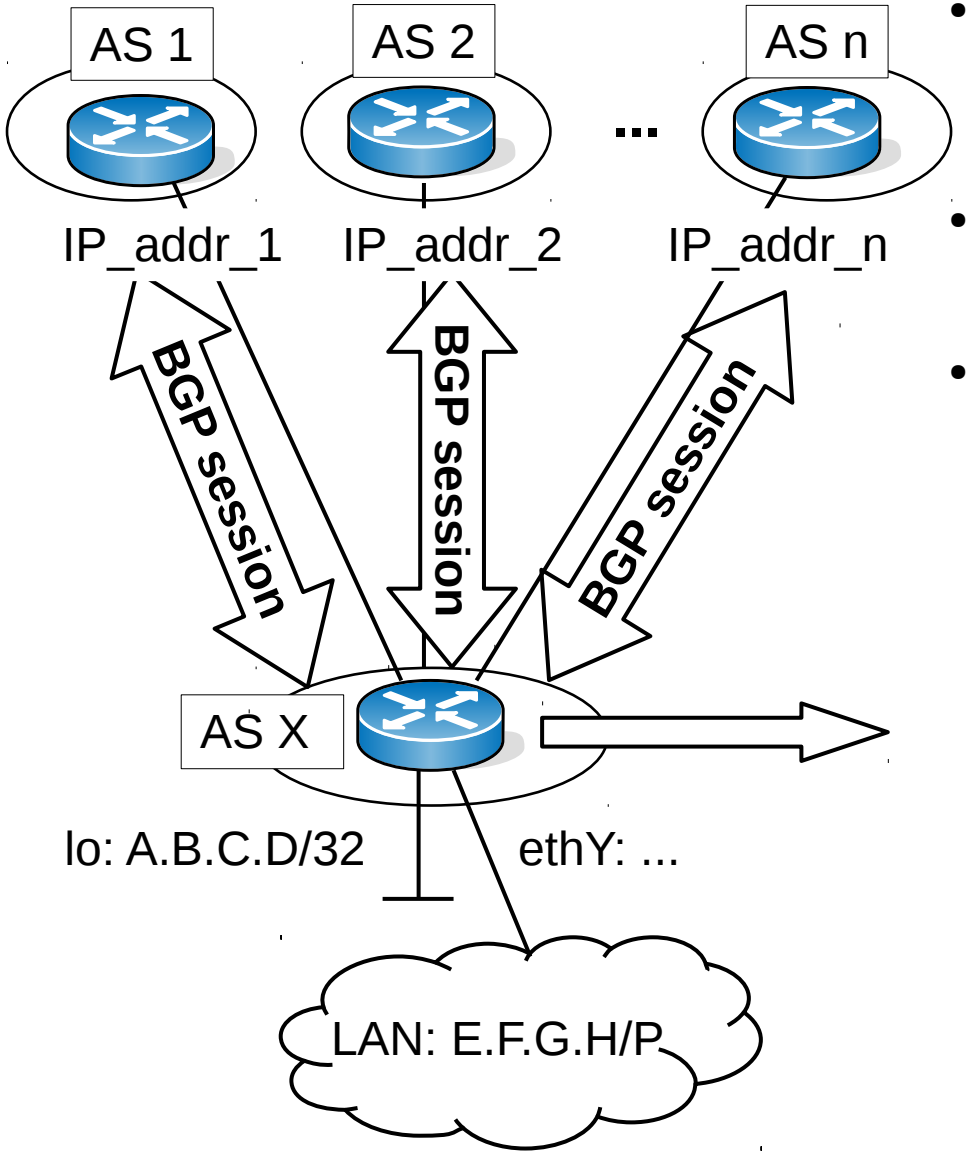
- The destinations in BGP are IP subnet prefixes
- Routing based on **AS-level paths**
- Routers advertise the best AS-path to each prefix to neighbors: **BGP announcement=prefix + attributes**
- Paths received from/advertised to neighbor ASes are subjected to **import/export filters:** withdraw/rewrite/ suppress paths
- Filtered BGP announcements go into an AS-path database: **BGP RIB**
- From this database **the best path is selected** to each advertised prefix: **active path**

The BGP routing process

- **BGP configuration:** BGP session config + announced prefixes + import/export filters
- Today, we concentrate on setting up BGP sessions exclusively, next time we learn BGP filters



Configuring BGP sessions



- Usually the a `router-id` and the advertised prefixes are configured to a (virtual) loopback interface on the router
- `lo`: pingable loopback interface
  `lo*`: virtual interface, cannot ping
- Can set multiple addresses on an interface

```
!! router-id
interface lo
 ip address A.B.C.D/32
!! advertised prefix/prefixes
interface ethY
 ip address E.F.G.H/P
router bgp X
 bgp router-id A.B.C.D
 network E.F.G.H/P
 neighbor IP_addr_1 remote-as 1
 neighbor IP_addr_2 remote-as 2
[…]
 neighbor IP_addr_n remote-as n
```

BGP configuration: check&debug

- Basic BGP information (BGP RIB): `show ip bgp`
  - local prefixes and prefixes learnt from neighbors
  - all AS-paths plus the active path marked by “>”

```
OpenWrt# show ip bgp
BGP table version is 0, local router ID is 10.0.0.3
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      10.5.0.2                       0 500 200 100 i
*>                  10.3.0.2                       0 200 100 i
*                   10.4.0.3                       0 400 200 100 i
*> 10.0.3.0/24      0.0.0.0          0         32768 i

Total number of prefixes 2
```

BGP configuration: check&debug

- All AS-paths known for a prefix or IP address:
  `show ip bgp A.B.C.D/P`

```
OpenWrt# show ip bgp 10.0.1.1
BGP routing table entry for 10.0.1.0/24
Paths: (3 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.4.0.1 10.5.0.2
  500 200 100
    10.5.0.2 from 10.5.0.2 (10.0.0.5)
      Origin IGP, localpref 100, valid, external
      Last update: Thu Mar 26 13:14:18 2015

  200 100
    10.3.0.2 from 10.3.0.2 (10.0.0.2)
      Origin IGP, localpref 100, valid, external, best
      Last update: Thu Mar 26 13:14:16 2015
  […]
```

BGP configuration: check&debug

- BGP neighbors: `show ip bgp neighbor`
- Concrete neighbor: `show ip bgp neigh IP_addr`

```
OpenWrt# show ip bgp neighbor 10.5.0.2
BGP neighbor is 10.5.0.2, remote AS 500,
                          local AS 300, external link
  BGP version 4, remote router ID 10.0.0.5
  BGP state = Established, up for 00:15:01
[…]
For address family: IPv4 Unicast
  Community attribute sent to this neighbor(both)
  1 accepted prefixes

  Connections established 1; dropped 0
  Last reset never
Local host: 10.5.0.1, Local port: 39110
Foreign host: 10.5.0.2, Foreign port: 179
[…]
```

BGP configuration: check&debug

- Summary on BGP neighbors' status: `show ip bgp summary`
- Reset all BGP sessions or session to a concrete neighbor (e.g., when `router-id` changes): `clear ip bgp IP_addr/clear ip bgp *`
- Dump IPv4 FIB: `show ip route`

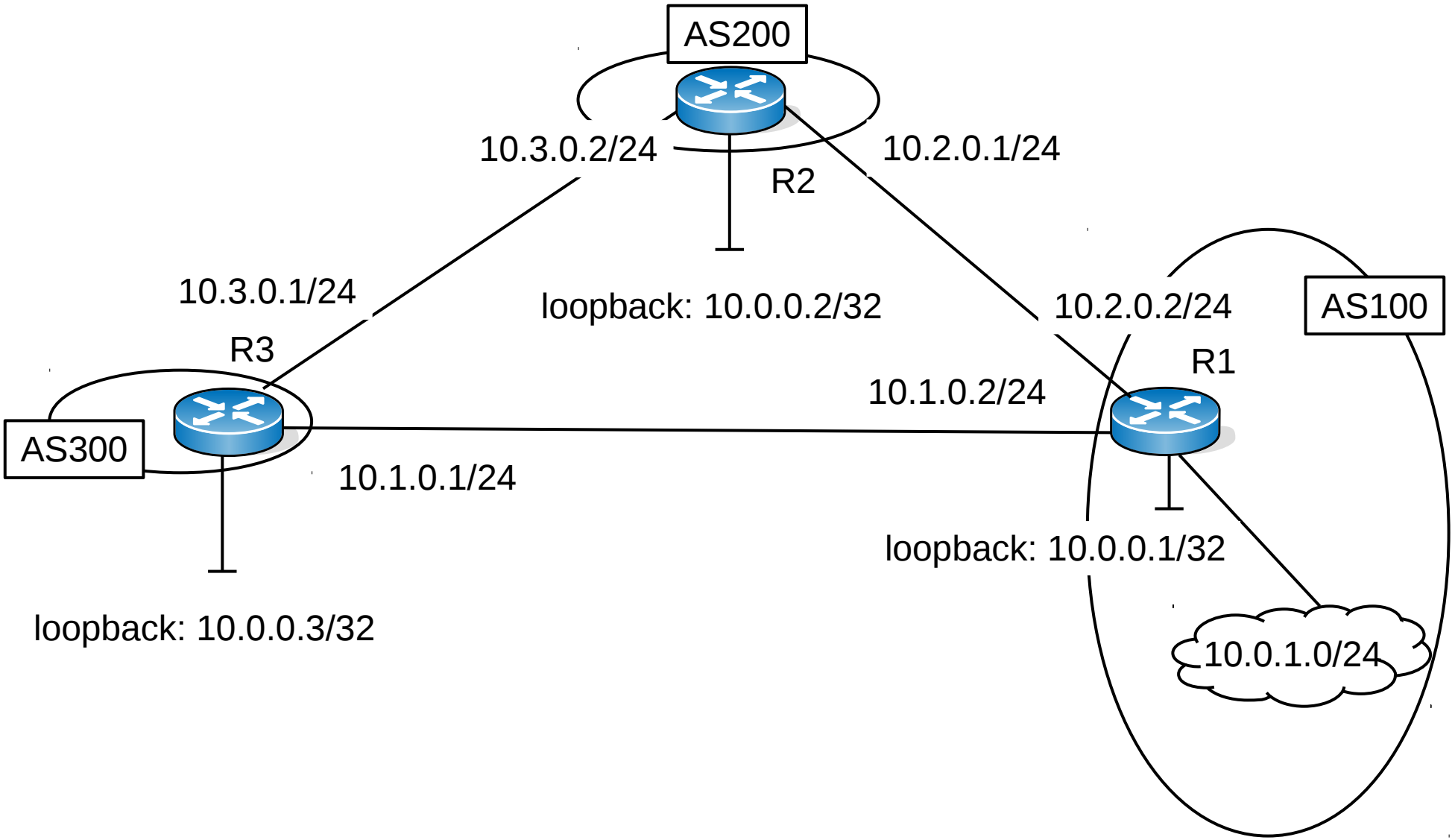
```
OpenWrt# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, H - HSLS, o - OLSR,
B>* 10.0.1.0/24 [20/0] via 10.3.0.2, eth1, 00:30:23
C>* 10.3.0.0/24 is directly connected, eth1
C>* 10.4.0.0/24 is directly connected, eth0
C>* 10.5.0.0/24 is directly connected, eth2
[…]
```

Exercises

Set up the below topology and create BGP sessions between neighboring routers using the IP addresses and prefixes as given in the figure. Choose the `router-ids` as `10.0.0.<AS_num/100>`!

1) Identify the BGP neighbors as seen by `R2` (`show ip bgp neigh`)? Check that none of the neighbors is in `Idle` state!
2) Note the AS-paths of router `R3` to the prefix `10.0.1.0/24` (`show ip bgp/show ip bgp A.B.C.D/X`). Which one is the active path?
3) Check that packets are indeed forwarded along the expected paths (`traceroute`)!
4) Withdraw the announcement of `10.0.1.0/24` on router `R1` (`no network 10.0.1.0/24`) and note the BGP messages traveling on the link `R2-R3` (`tcpdump/wireshark`)! Re-annonce the prefix at `R1` and again note the BGP messages captured on the interface!

Exercises


AS200
10.3.0.2/24
R2
10.2.0.1/24
10.3.0.1/24
loopback: 10.0.0.2/32
10.2.0.2/24
AS100
R3
R1
10.1.0.2/24
AS300
10.1.0.1/24
loopback: 10.0.0.1/32
loopback: 10.0.0.3/32
10.0.1.0/24