Università di Roma
Tor Vergata

University of Rome Tor Vergata
ICT and Internet Engineering

Network and System Defense

Alessandro Pellegrini, Angelo Tulumello

A.A. 2023/2024

Lecture 8:
BGP basics

Angelo Tulumello

Slides by Marco Bonola

Routing Protocols at a Glance

Routing protocols

- ***Goal***: setup IP routing tables according to configurable administrative rules
- ***Routing protocols populate the Routing Tables***
  - Possibly more routes for the same destination prefix
- An internal algorithm choses ***best routes per prefix*** and inject this choice in the Forwarding Table used during packet forwarding operation

Interior and Exterior Routing Protocols

- IP routing protocols fall into one of two major categories: ***interior gateway protocols*** (***IGP***) or ***exterior gateway protocols*** (***EGP***)
  - ***IGP***: A routing protocol that was designed and intended for use inside a single autonomous system (AS)
  - ***EGP***: A routing protocol that was designed and intended for use between different autonomous systems
- An AS is a network under the administrative control of ***single organization***
- Each AS can be assigned a number called ***AS number (ASN)***
- Each AS has one (or more) public IP address range(s)

Interior Gateway Protocols

- Organizations have several options when choosing an IGP for their enterprise network (but most companies today use ***OSPF***)
- Two main branches of routing protocol algorithms exist for IGP routing protocols:
  - ***Distance Vector*** (e.g. RIP)
  - ***Link-state*** (e.g. OSPF, IS-IS)
- In a ***distance vector protocol*** the routers only exchange the routing table with their neighbors (i.e. routers on the same links)
- ***Link state protocols*** require the synchronization and the knowledge of the entire network topology on every router in the AS

Interior Gateway Protocols

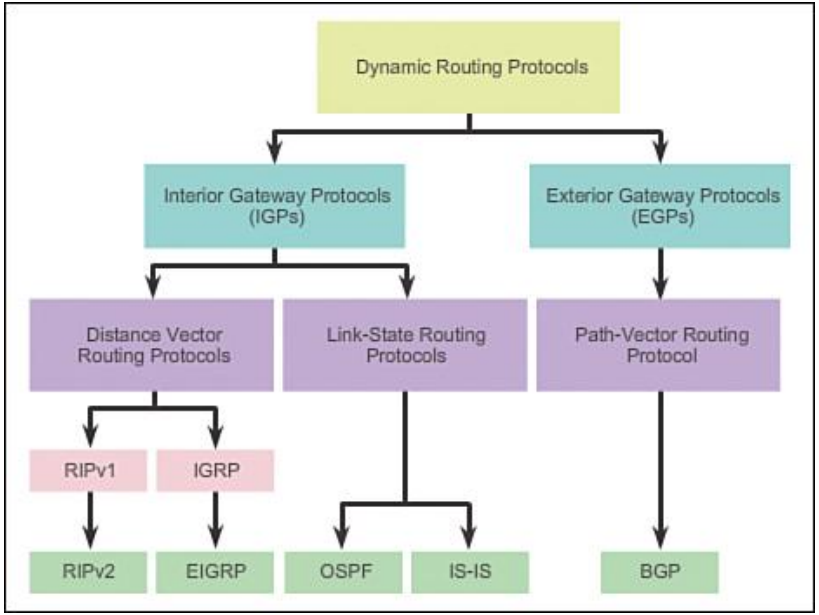
- ❑ Organizations have several optio rprise network (but most com
- ❑ Two main bra prot
  - ❑
- ❑ In routing table with the
- ❑ Link nization the knowledge of the entire netw uter in the AS

Although IGPs are a fundamental building block of IP networks and their vulnerabilities should be carefully addressed, IGPs security is outside the scope of this course

Exterior Gateway Protocols

- ❑ Typically, EGPs are used to exchange routing information between ISPs, or in some cases between a customer’s AS and the provider’s network.
- ❑ ***Border Gateway Protocol (BGP)***, version 4 (BGP4) is the most common EGP and is considered the Internet standard.
- ❑ An internetwork is a confederation of smaller, independent networks, called autonomous systems, owned and operated by a different organization: a company, university, government agency, or some other group.
- ❑ Each AS typically represents an independent organization, and applies its own unique routing and security policies.
- ❑ EGPs facilitate the sharing of routing information between autonomous systems.

The Big Picture



Dynamic Routing Protocols

Interior Gateway Protocols (IGPs)

Exterior Gateway Protocols (EGPs)

Distance Vector Routing Protocols

Link-State Routing Protocols

Path-Vector Routing Protocol

RIPv1

IGRP

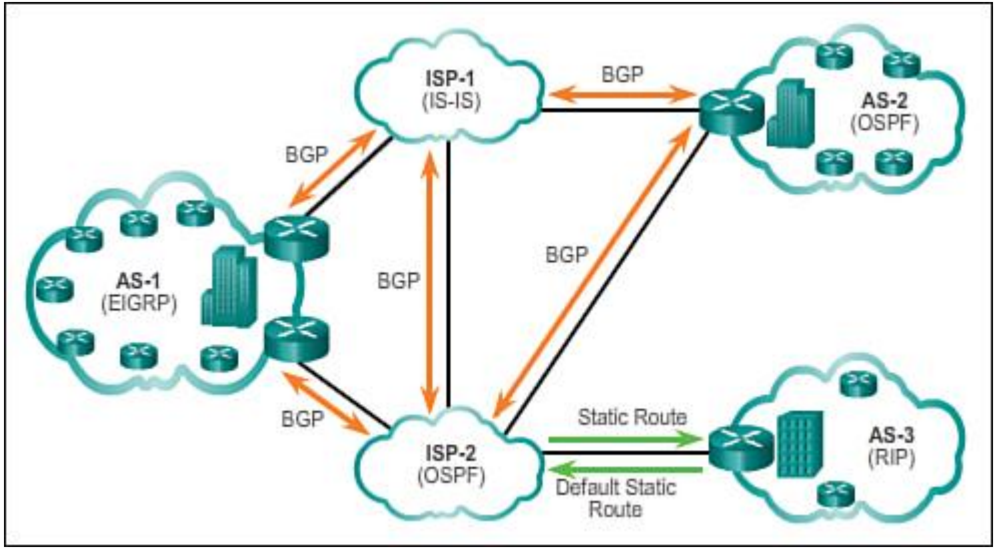
RIPv2

EIGRP

OSPF

IS-IS

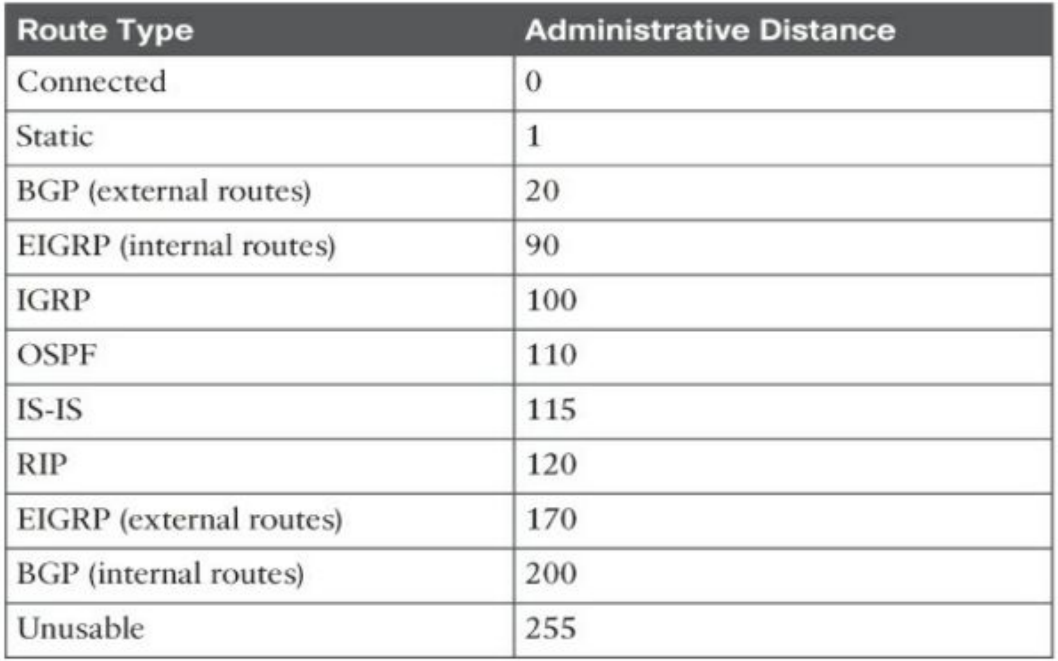
BGP



Administrative distance

- ***Multiple routing protocols in a same domain***
- Routing protocols might learn routes to the same subnets
- When a node must choose between routes learned using different routing protocols, the so-called ***administrative distance*** is compared
- Administrative distance is a number that denotes how believable an entire routing protocol is on a single router.
- The lower the number, the better, or more believable, the routing protocol.
- ***The administrative distance values are configured on a single router and are not exchanged with other routers***

Administrative distance on CISCO IOS



Route Type	Administrative Distance
Connected	0
Static	1
BGP (external routes)	20
EIGRP (internal routes)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EIGRP (external routes)	170
BGP (internal routes)	200
Unusable	255

Border Gateway Protocol (BGP)

Warning: BGP would require a lot of time to be exhaustively covered. This is going to be more of an overview required to understand the the next two topics: BGP/MPLS VPNs and BGP security

A special thanks to Andrea Detti for most of the following slides

BGP Basics

- BGP is a distance vector routing protocol (more specifically a ***path vector routing protocol***)
- It relies on downstream neighbors to pass along routes from their routing table.
- The node ***makes its route calculations*** based on the advertised routes ***and passes the results to upstream neighbors***.
- BGP uses a list of AS numbers through which a packet must pass to reach a destination as the distance metric to minimize
- In the most simple scenario, a BGP “announcement” contains the following BGP attributes (more about attributes later ...)
  - A network address
  - The path vector carrying the list of ASes traversed to reach the destination
  - The next hop to reach the destination
- Each AS has an identifying number, assigned by an Internet registry or a service provider, between 1 and 65535.
  - Private AS numbers: Between 64512 through 65535
- AS numbers, as IPv4 addresses have begun to **run out** → RFC 6793 introduced 4-bytes AS numbers
  - Private AS numbers: Between 4200000000 to 4294967294

BGP Basics

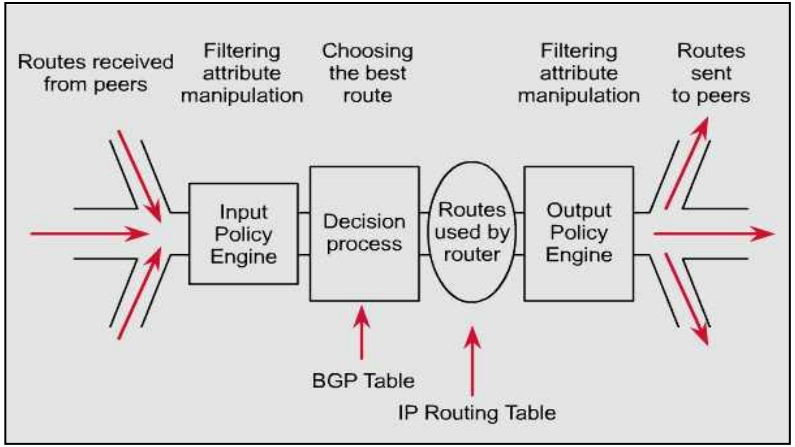
- ❏ ***BGP runs on TCP*** (so differently from OSPF, IP/TCP connectivity is required)
- ❏ When two routers establish a TCP-enabled BGP connection between each other, they are called ***neighbors*** or ***peers***.
- ❏ Each router running BGP is called a ***BGP speaker***.
- ❏ When two neighbors first establish a BGP connection, they exchange their entire BGP routing tables.
- ❏ After that, they exchange incremental, partial updates with only the information that has changed.
- ❏ Peers exchange keepalive messages to ensure the connection is maintained.
- ❏ If three keepalive intervals pass the peer declares its neighbor down and all the routes advertised by the peer are withdrawn

BGP Basics

- Before establishing a BGP peer connection the two neighbors must perform the standard TCP three-way handshake and open a TCP connection to port 179.
- After the TCP session is established, BGP peers exchanges several messages to open and confirm connection parameters and to send BGP routing information.
- All BGP messages are unicast to the one neighbor over the TCP connection.
- There are four BGP message types:
  - ***Type 1: OPEN***
  - ***Type 2: KEEPALIVE***
  - ***Type 3: UPDATE***
  - ***Type 4: NOTIFICATION***

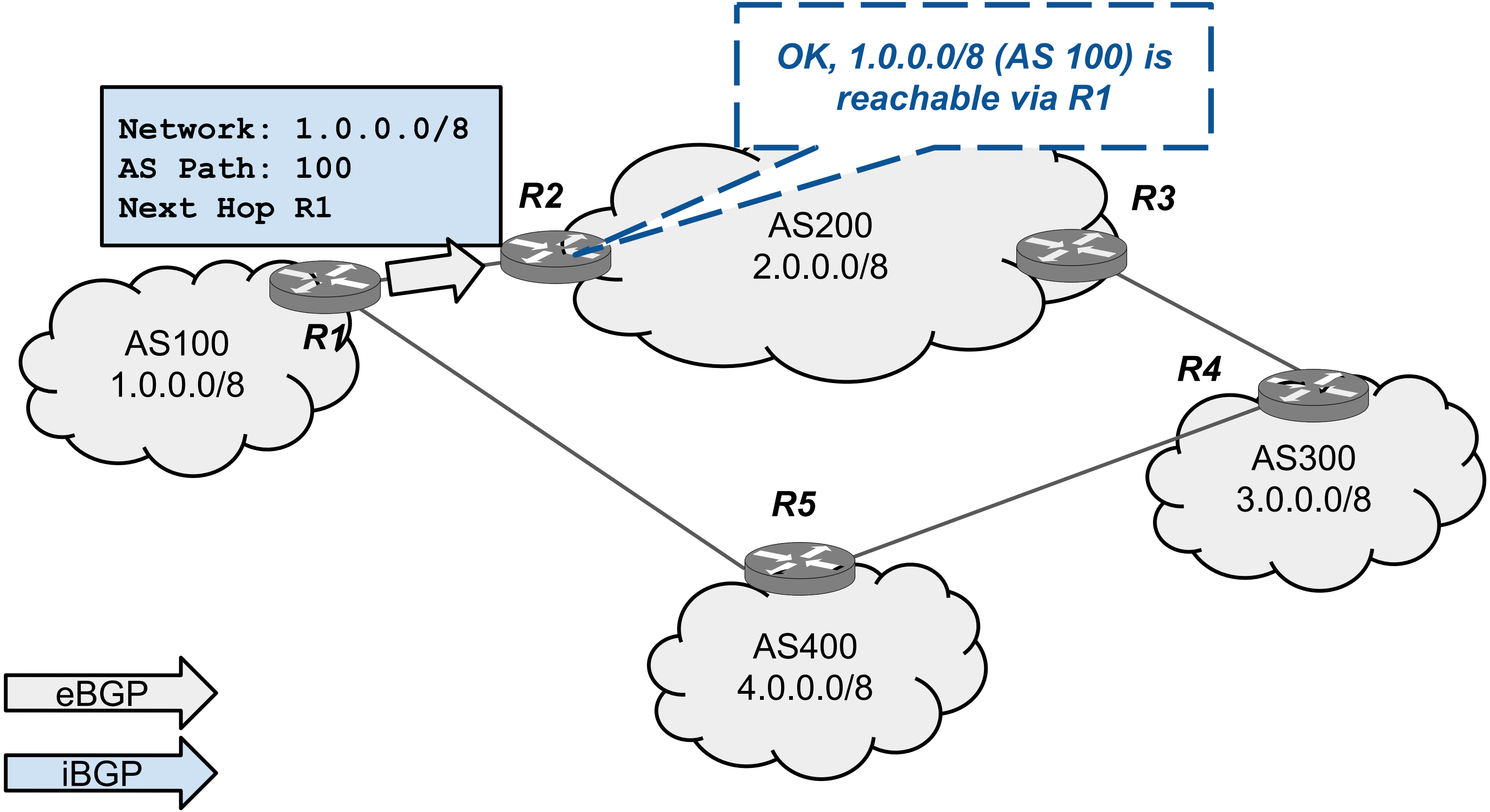
BGP Routing process

- ❑ BGP is so flexible because it is a fairly simple protocol.
- ❑ Routes are exchanged between BGP peers via UPDATE messages.
- ❑ BGP routers receive the UPDATE messages, run some policies or filters over the updates, and then pass on the routes to other BGP peers.
- ❑ Different implementations of BGP keep track of all BGP updates in a BGP table separate from the IP routing table.



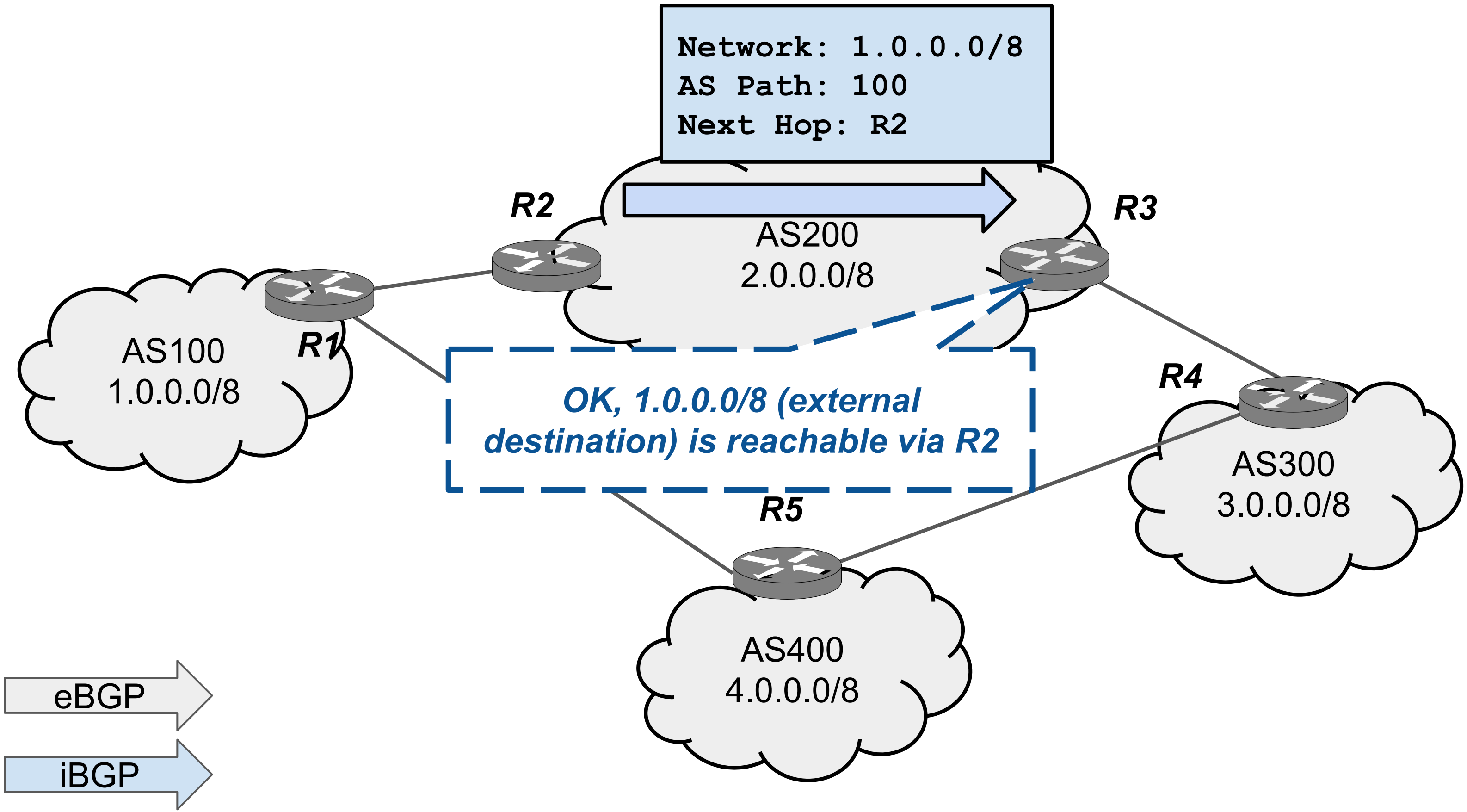
But BGP is not only used among different ASes

- BGP is also used to distribute routing information for external destination among routers in the same AS
- When BGP is running inside an AS, it is referred to as ***Internal BGP (IBGP)***.
  - If a BGP router role is to route IBGP traffic, it is called a transit router.
- When BGP runs between autonomous systems, it is called ***External BGP (EBGP)***.
  - Routers that sit on the boundary of an AS and use EBGP to exchange information with the ISP are called border routers.
- “With very few exceptions, interior BGP (IBGP) – BGP between peers in the same AS – is used only in multihomed scenarios.” – Doyle



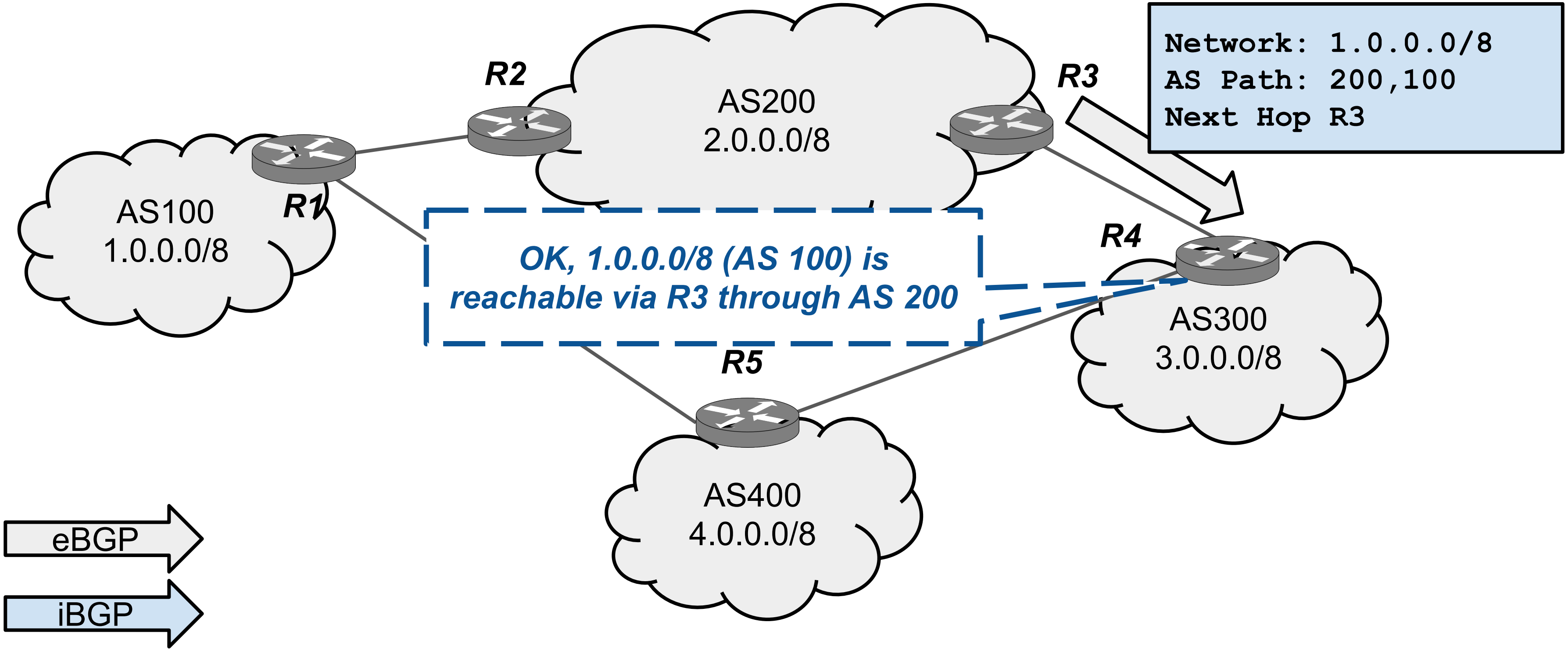
```
Network: 1.0.0.0/8
AS Path: 100
Next Hop R1
```

OK, 1.0.0.0/8 (AS 100) is reachable via R1



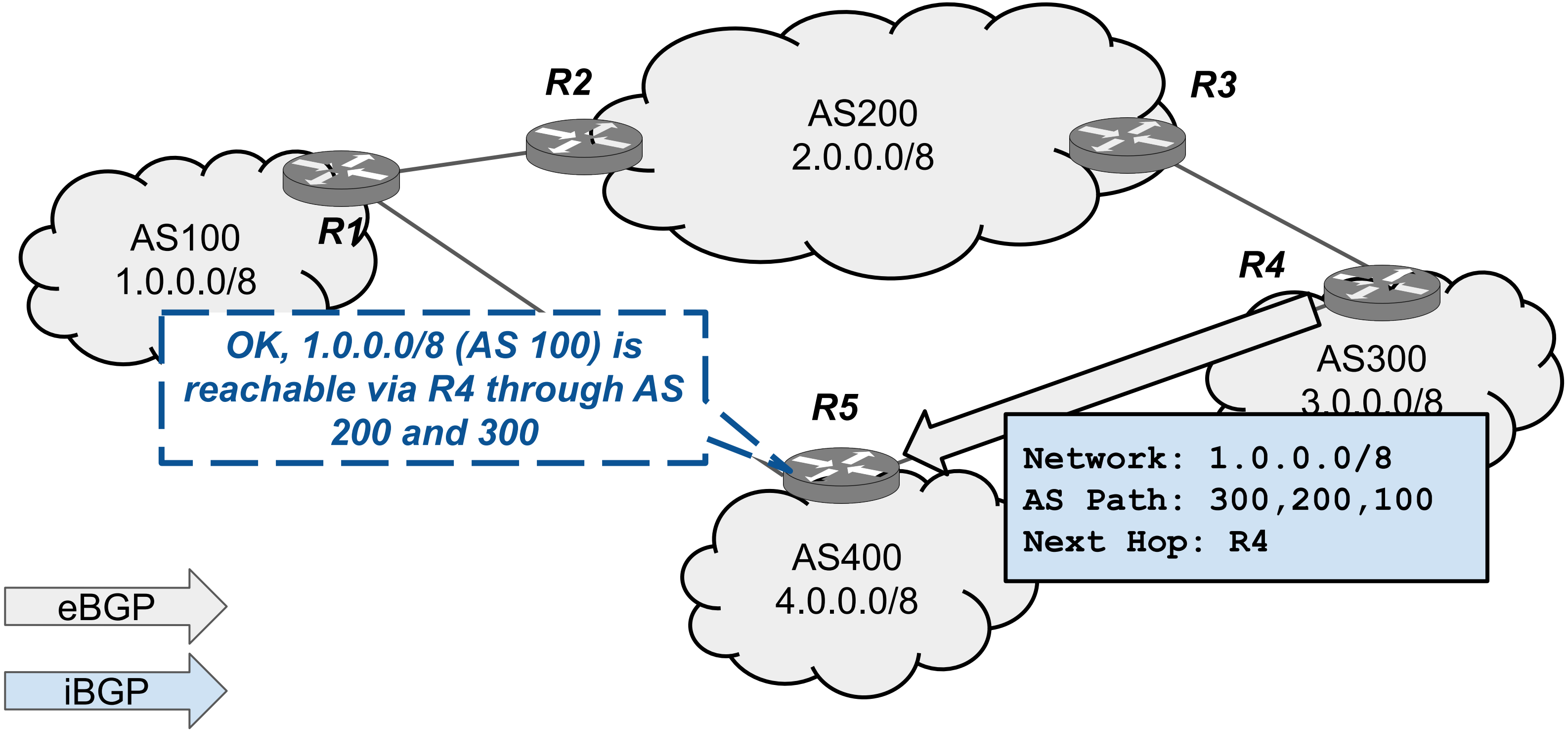
```
Network: 1.0.0.0/8
AS Path: 100
Next Hop: R2
```

OK, 1.0.0.0/8 (external destination) is reachable via R2



```
Network: 1.0.0.0/8
AS Path: 200,100
Next Hop R3
```

OK, 1.0.0.0/8 (AS 100) is reachable via R3 through AS 200

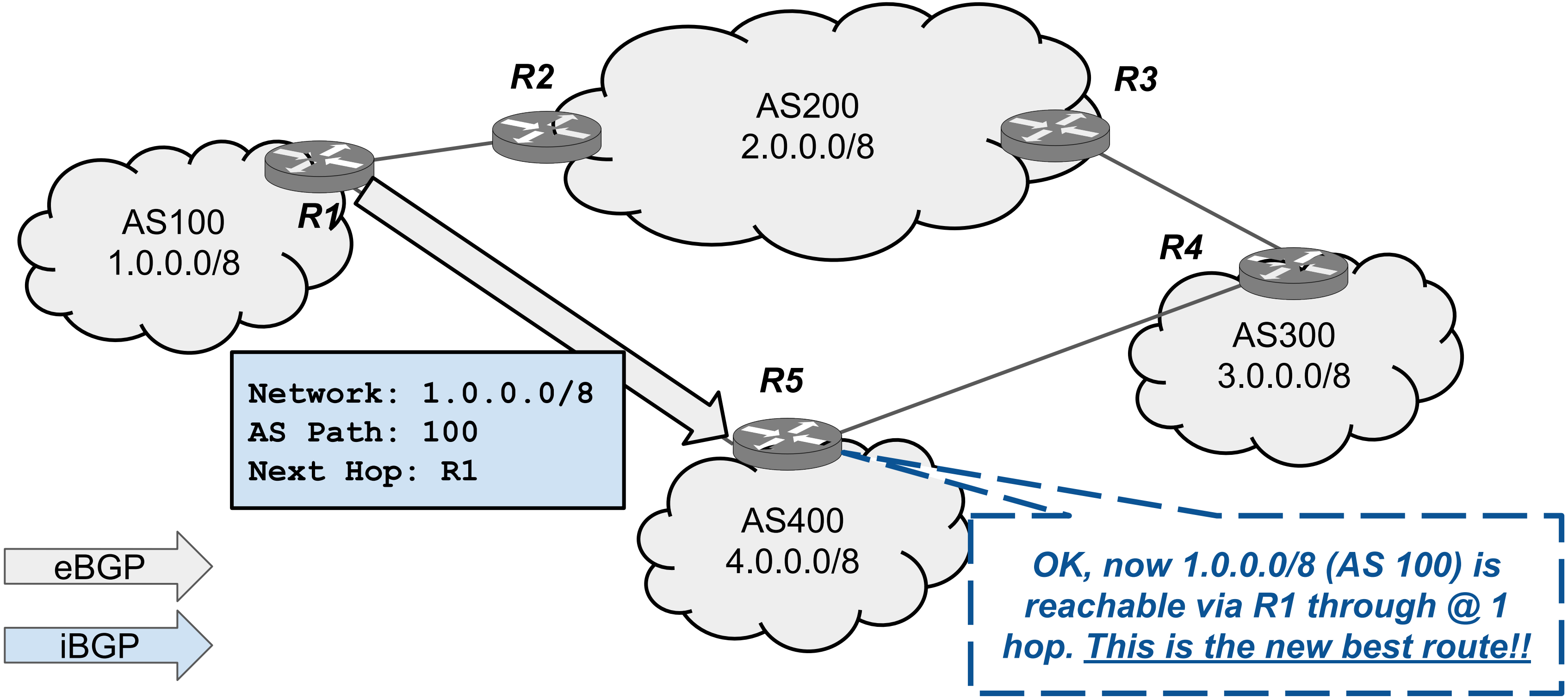


OK, 1.0.0.0/8 (AS 100) is reachable via R4 through AS 200 and 300

```
Network: 1.0.0.0/8
AS Path: 300,200,100
Next Hop: R4
```

How BGP chooses the best path

- Network topology has nothing to do with geographical topology
  - of course, it is more probable that a AS at 1 hop is most likely geographically closer AS
  - but ASes can be distributed over the entire world..
  - and peering also depends on commercial agreement
- Let’s imagine that the BGP topology changes
  - For example, R1 and R5 become neighbours
- In this case R5 would receive an advertisement for 1.0.0.0/8 with AS_PATH=100 and next_hop R1
- The current routing entry to 1.0.0.0/8 is next_hop R4, AS_PATH=100,200,300
- ***The new entry is preferred (because it has a shorter AS_PATH attribute) and installed in the global IP routing table***



```
Network: 1.0.0.0/8
AS Path: 100
Next Hop: R1
```

OK, now 1.0.0.0/8 (AS 100) is reachable via R1 through @ 1 hop. This is the new best route!!

BGP loop avoidance

- AS number is added to the AS path only when a BGP UPDATE goes out from the AS
  - **Possible loops can be easily detected**
  - AS path check can not be used to avoid loops within a AS
- BGP Split Horizon Rule used to avoid loop inside the AS
  - A route learned via IBGP is never propagated to other IBGP peers.
  - A route is not advertised back to the EBGP peer from which the route was received
- Consequence: if all internal routers need the full routing table, there should be a full-mesh of IBGP sessions
  - unless we configure a so-called ***BGP Route Reflector***

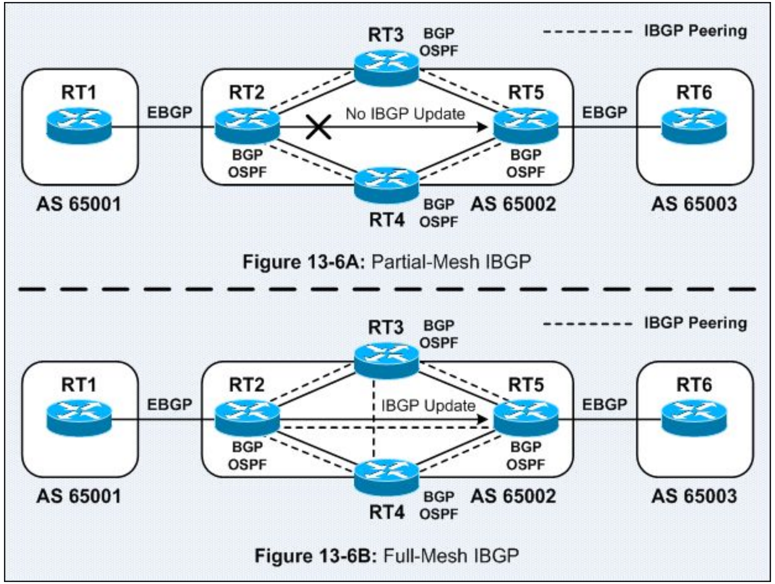


Figure 13-6A: Partial-Mesh IBGP

Figure 13-6B: Full-Mesh IBGP

BGP Attributes

- Part of a BGP Update
  - Describes the characteristics of prefix
  - Each route has its own set of defined attributes, which can include path information, route preference, next-hop, and aggregation information.
  - The destination network, the AS path and the next hop are mandatory BGP attributes
- ***Administrators use these attributes to enforce routing policy.***
- Based on attribute values, you can configure BGP to filter routing information, prefer certain paths, or otherwise customize its behavior.
- Every UPDATE message has a variable-length sequence of path attributes in the form **`<attribute type, attribute length, attribute value>`**.
- Since you will use path attributes extensively when configuring routing policy, you should note that not all vendor implementations of BGP recognize the same attributes. In fact, path attributes come in four different types: (1) ***Well-known mandatory***, (2) ***Well-known discretionary***; (3) ***Optional transitive***; (4) ***Optional non-transitive***

BGP Attributes and Path Selection

- BGP selects only one path as the best path.
- When the path is selected, BGP puts the selected path in its (BGP) routing table and propagates the path to its neighbors.
- The AS Path attribute is not the only one used in the selection
- BGP uses the following criteria, in the order presented(*), to select a path for a destination:
  - ***"We Love Oranges AS Oranges Mean Pure Refreshment"***
  - **W** Weight (Highest)
  - **L** LOCAL_PREF (Highest)
  - **O** Originate (local)
  - **AS** AS_PATH (shortest)
  - **O** ORIGIN Code (IGP > EGP > Incomplete)
  - **M** MED (lowest)
  - **P** Paths (External > Internal)
  - **R** RID (lowest)
- ***Each attribute has its own semantic and usage***

(*) actually depends on the specific router BGP implementation

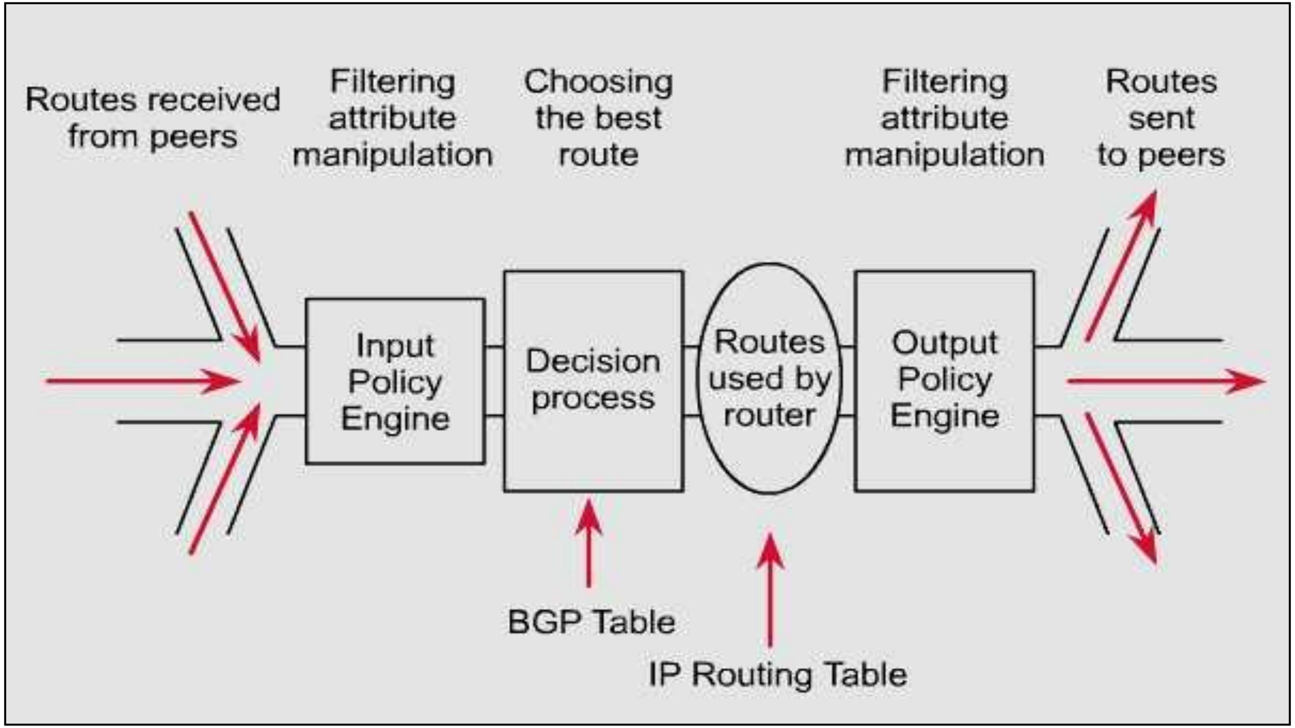
BGP Attributes and Path Selection

- ❑ BGP selects only one path as the best path.
- ❑ When the path is selected, BGP puts the selecte gates the path to its neighbors.
- ❑ The AS Path attribute is not th
- ❑ BGP uses the follo ination:
  - ❑ *"We*
  - ❑
  - ❑
  - ❑ mplete)
  - ❑
  - ❑ al > Internal)
  - ❑ D (lowest)
- ❑ ***Each attribute has its own semantic and usage***

A comprehensive study of the BGP attributes is outside the scope of this course

(*) actually depends on the specific router BGP implementation

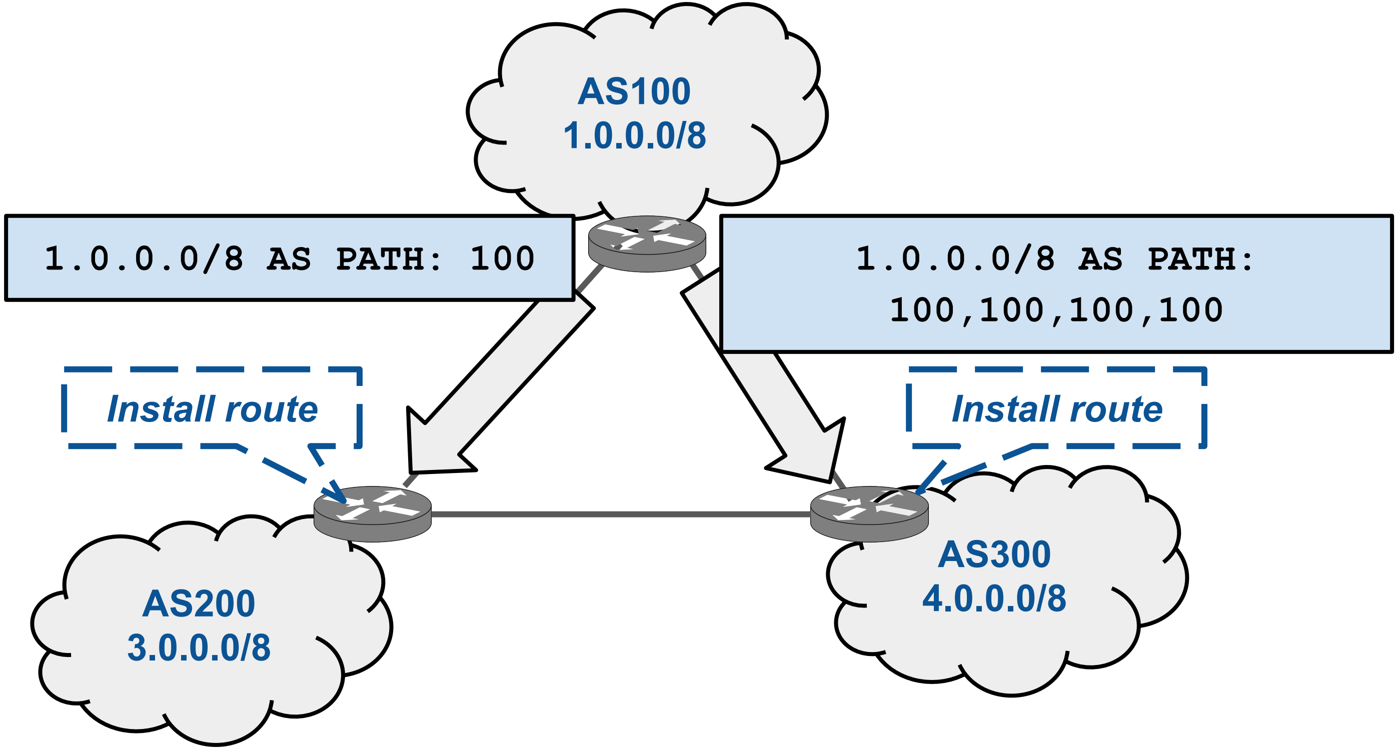
The BGP processing pipeline



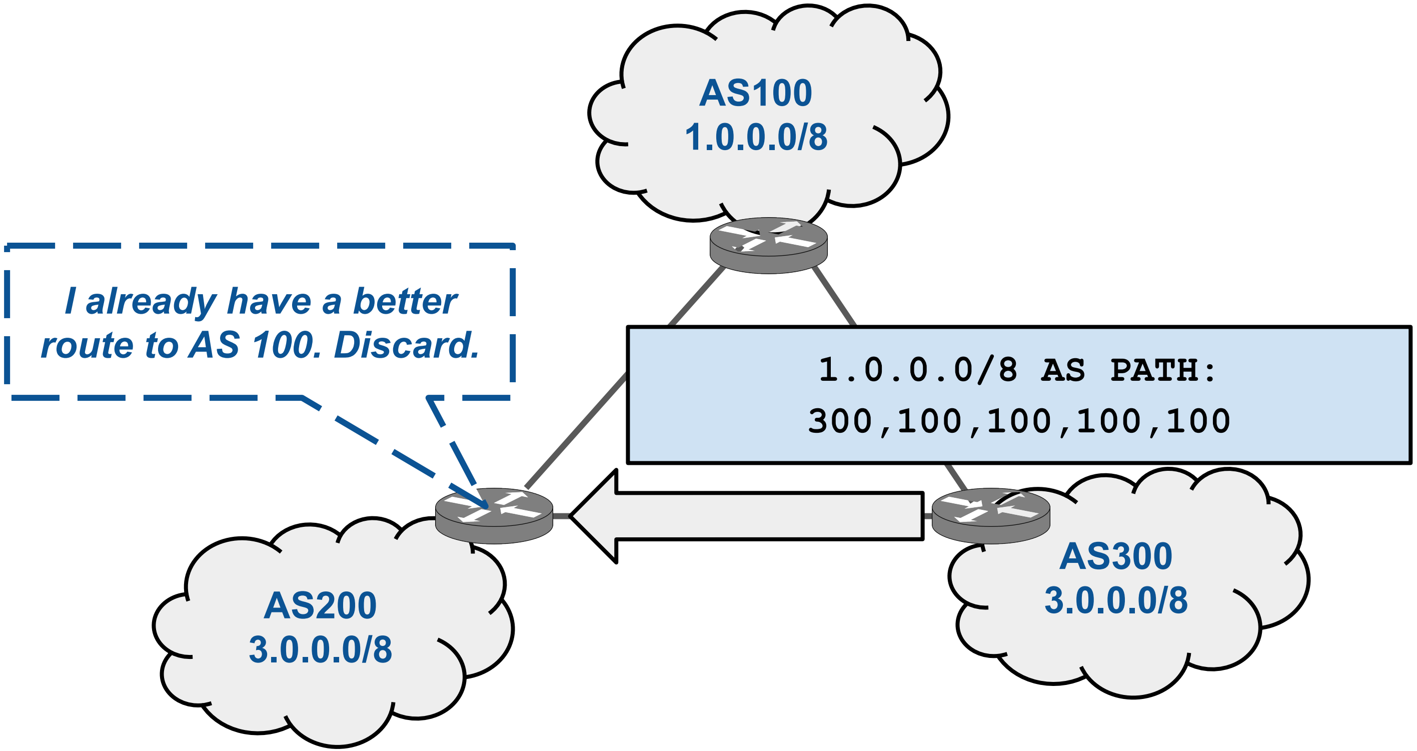
BGP Attributes Manipulation

- BGP attributes can be manipulated (modified, added, removed...)
  - in incoming BGP updates, before the actual route selection
  - in outgoing BGP updates
- For example, a well known mechanism to influence the path selection is the so-called ***AS-PATH prepend mechanism***
  - path information are changed by including dummy AS path numbers that would increase the path length and influence the traffic trajectory one way or the other
- BGP attributes are a fundamental BGP tool
- With BGP attributes routes can be chosen in order not only to minimize the number of AS traversed to reach the destination

AS PATH prepend example



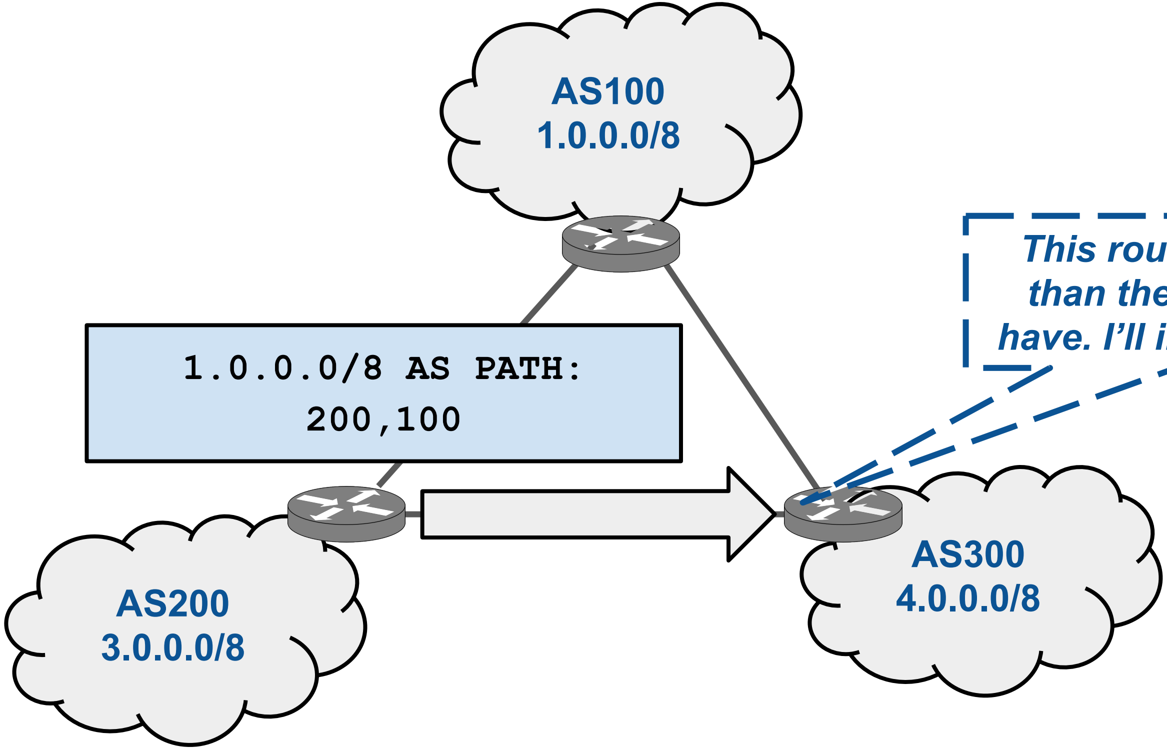
AS PATH prepend example



I already have a better route to AS 100. Discard.

```
1.0.0.0/8 AS PATH:
300,100,100,100,100
```

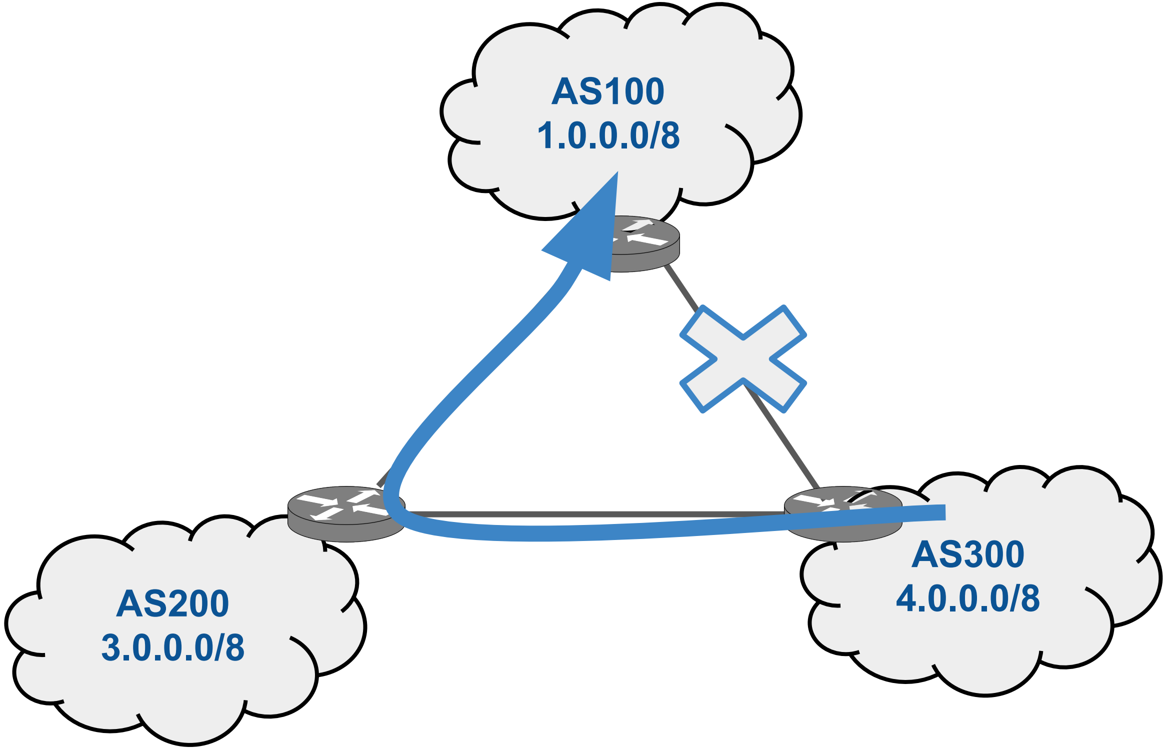
AS PATH prepend example



```
1.0.0.0/8 AS PATH:
200,100
```

This route is “shorter” than the one I already have. I’ll install this one!!

AS PATH prepend example



COMMUNITY attribute

- The BGP community attribute is an ***optional transitive attribute***
  - we’ll see that it can be used to control the BGP/MPLS VPNs topologies..
- It represents a ***numerical value*** that can be assigned to a specific prefix and advertised to other neighbors
- When the neighbour receives the prefix it will examine the community value and take proper action whether it is filtering or modifying other attributes
- A BGP community is a 32-bit number that can be included with a route. A BGP community can be displayed as a full 32-bit number (0-4,294,967,295) or as two 16-bit numbers (0-65535):(0-65535) commonly referred to as new-format.
- A number of well known “standard” community values exists (see next slide)

Well known community values

- ❑ ***NO_EXPORT (0xFFFFFF01)***
  - ❑ A route carrying this community value should not be advertised to peers outside a confederation (or the AS if it is the only AS in the confederation).
- ❑ ***NO_ADVERTISE (0xFFFFFF02)***
  - ❑ A route carrying this community value, when received, should not be advertised to any BGP peer
- ❑ ***Internet (0xFFFFFF03)***
  - ❑ A route carrying this community value, when received, should be advertised to all other routers.
- ❑ ***Local-as (0xFFFFFF04)***
  - ❑ A route carrying this community value, when received, should be advertised to peers within the AS, but not advertised to peers in an external system
- ❑ ***Custom community values can be also used***
  - ❑ The private BGP community pattern could vary from organization to organization, do not need to be registered, and could signify geographic locations for one AS while signifying a method of route advertisement in another AS

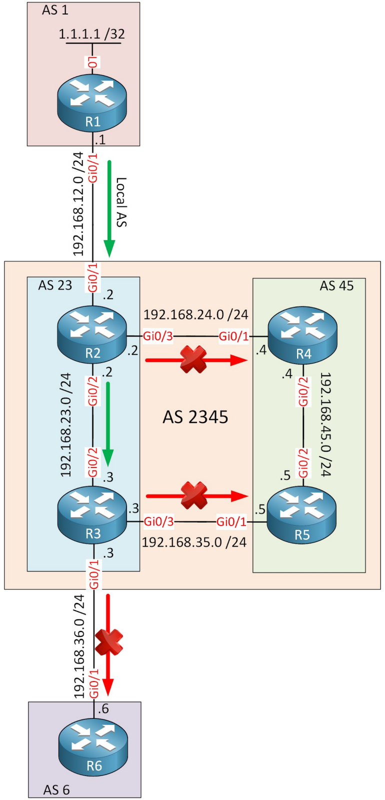


Fig. example of the Local-AS community

BGP configuration (Free Range Routing)

AS number

- To begin configuring a BGP process, issue the following familiar command:

```
Router(config)# router bgp AS-number
```

- BGP configuration commands appear on the surface to mirror the syntax of familiar IGP (for example, RIP, OSPF) commands.
- Although the syntax is similar, the function of these commands is significantly different.

Manually advertise a network

```
Router(config-router)#network network-number [mask network-mask]
```

- ❏ The ***network*** command is used with IGPs, such as OSPF, to determine the interfaces on which to send and receive updates, as well as which directly connected networks to advertise.
- ❏ When configuring BGP, the network command does not affect what interfaces BGP runs on.
- ❏ In BGP, the network command tells the BGP process ***what locally learned networks to advertise.***
- ❏ The networks can be connected routes, static routes, or routes learned via a dynamic routing protocol, such as OSPF

Manually advertise a network

- ❑ ***These networks must also exist in the local router’s routing table (show ip route), or they will not be sent out in updates.***
- ❑ You can use the mask keyword with the network command to specify individual subnets.
- ❑ Routes learned by the BGP process are propagated by default, but are often filtered by a routing policy.

BGP peering

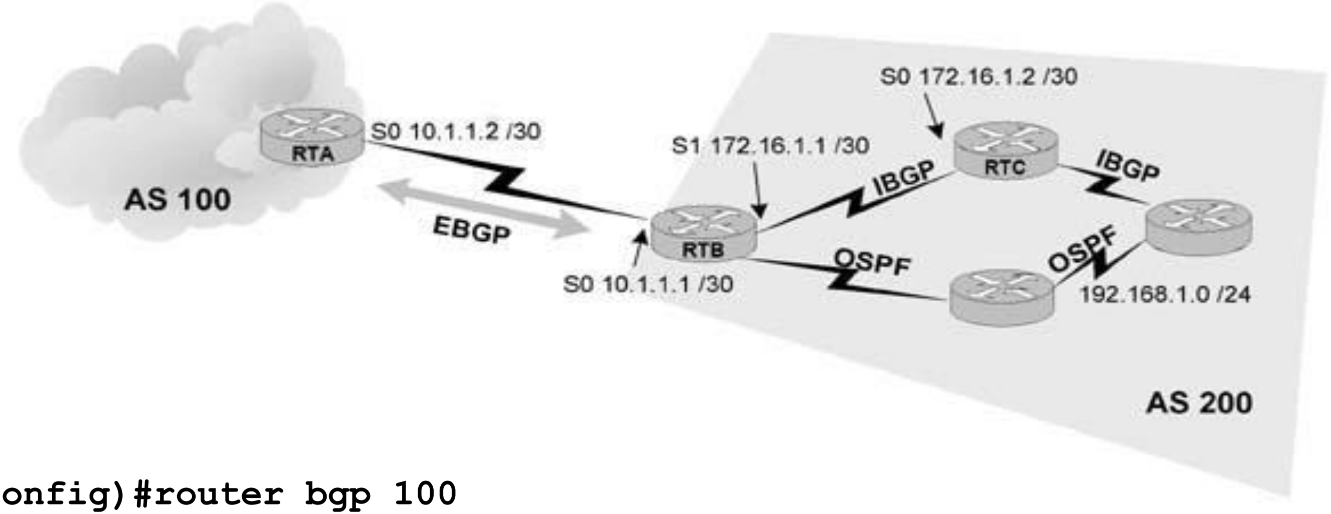
```
Router(config-router)#neighbor ip-address remote-as AS-number
```

- In order for a BGP router to ***establish a neighbor relationship with another BGP router,*** you must issue the above configuration command.
- This command serves to identify a peer router with which the local router will establish a session.
- The ***AS-number*** argument determines whether the neighbor router is an EBGP or an IBGP neighbor.

BGP and Loopback interfaces

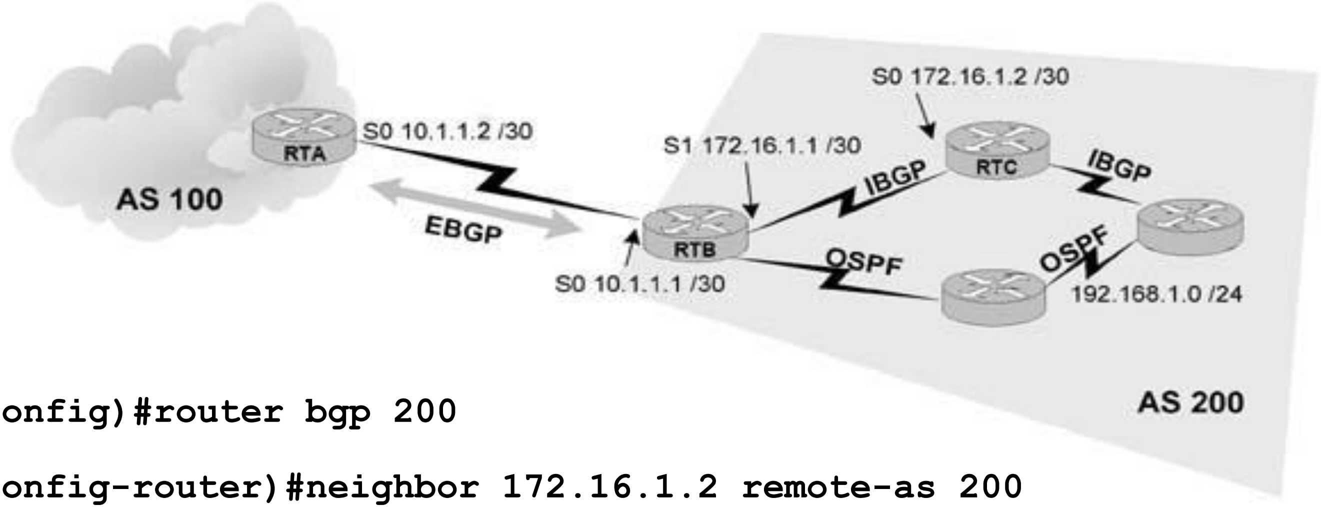
- BGP peering session may use ***loopback interfaces***
- Loopback interface ensures that the neighbor stays up and is not affected by malfunctioning hardware of a specific interface
- Useful when there are multiple paths between the BGP peers
  - iBGP peering are among internal routers generally connected by different IGP (OSPF) routes
- Useful also in case of multilink, for load balancing
- Best Common Practice
  - IBGP on loopback
  - EBGP on physical interface (unless multilink)

eBGP peering



```
RTA(config)#router bgp 100

RTA(config-router)#neighbor 10.1.1.1 remote-as 200

RTB(config)#router bgp 200

RTB(config-router)#neighbor 10.1.1.2 remote-as 100
```

iBGP peering



```
RTB(config)#router bgp 200

RTB(config-router)#neighbor 172.16.1.2 remote-as 200

RTB(config-router)#neighbor 172.16.1.2 update-source loopback 0

RTC(config)#router bgp 200

RTC(config-router)#neighbor 172.16.1.1 remote-as 200

RTC(config-router)#neighbor 172.16.1.1 update-source loopback 0
```

Loopback source

- The ***update-source loopback 0*** command is used to instruct the router to use any operational interface for TCP connections (as long as Lo0 is up and configured with an IP address).
- Without the ***update-source loopback 0*** command, BGP routers can use only the closest IP interface to the peer.
- The ability to use any operational interface provides BGP with robustness in the event the link to the closet interface fails.
  - Since eBGP sessions are typically point-to-point, there is no need to use this command with eBGP.

Reset the BGP process

- Finally, whenever you are configuring BGP, you will notice that changes you make to an existing configuration may not appear immediately.
- To force BGP to clear its table and reset BGP sessions, use the clear ip bgp command. The easiest way to enter this command is as follows:

```
Router# clear ip bgp *
```

- ***Use this command with CAUTION, better yet not at all, in a production network.***
    - remember October 4, 2021 Meta Inc. blackout?!
        - https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/
        - https://blog.cloudflare.com/october-2021-facebook-outage/

Verifying BGP Configurations

```
RTA#show ip bgp
BGP table version is 3, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i1.0.0.0          192.168.1.6              0    100      0 200 400 e
*>i10.1.1.1/32      10.1.1.1                 0    100      0 i
*>i172.16.1.0/24    10.1.1.1                 0    100      0 i
* i192.168.1.32/27  192.168.1.6              0    100      0 200 i
```

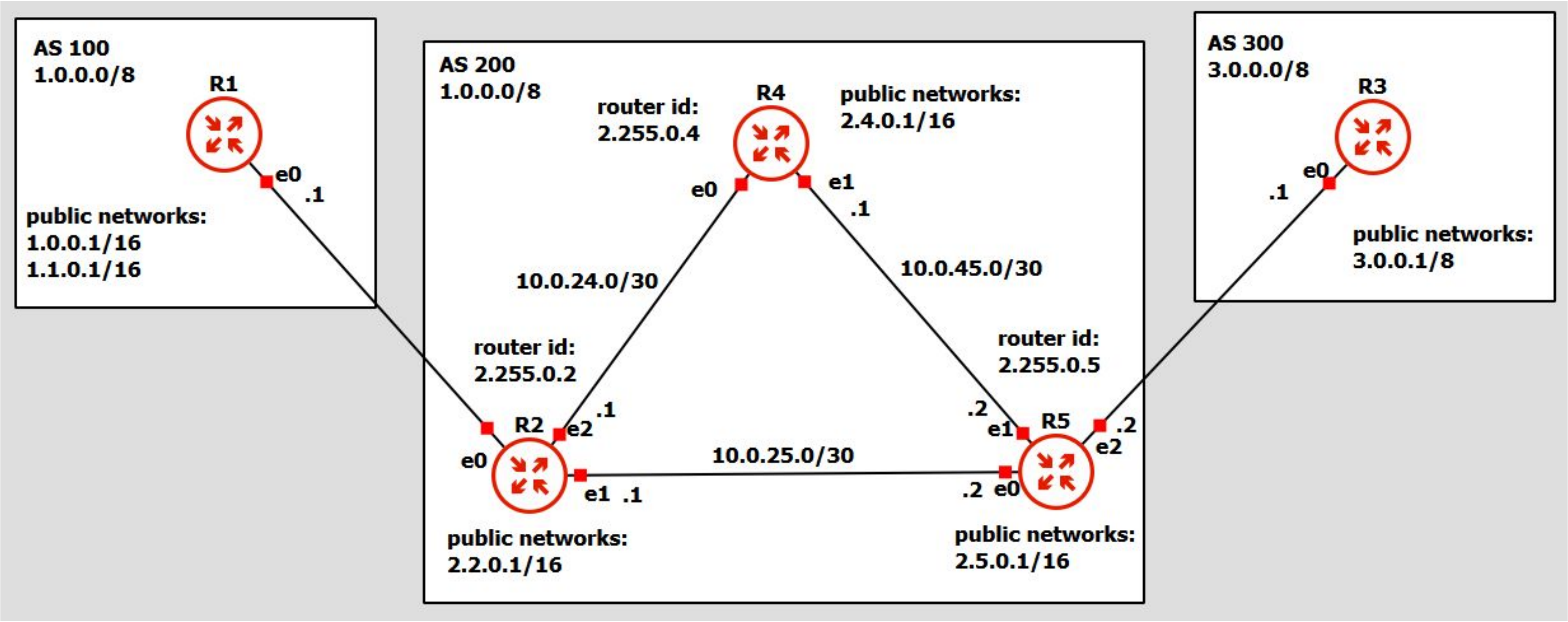
Verifying BGP Configurations

```
RTA#show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 200, external link
  BGP version 4, remote router ID 172.16.1.1
  BGP state = Established, up for 00:03:25
  Last read 00:00:25, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received
    Address family IPv4 Unicast: advertised and received
  Received 7 messages, 0 notifications, 0 in queue
  Sent 8 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds
```

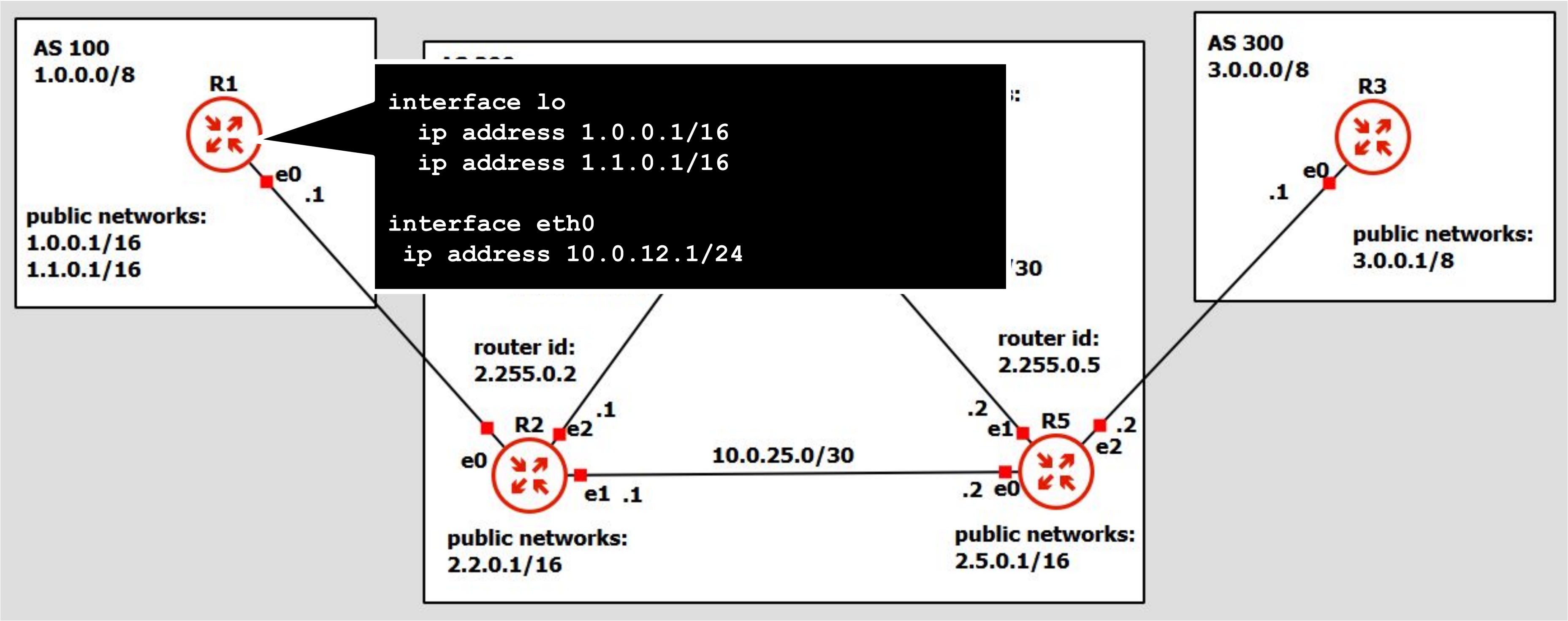
it would be interesting to show you how to manipulate BGP attributes to enforce some kind of routing policies. Unfortunately we don’t have time for this...

Laboratory: a simple BGP scenario

Topology

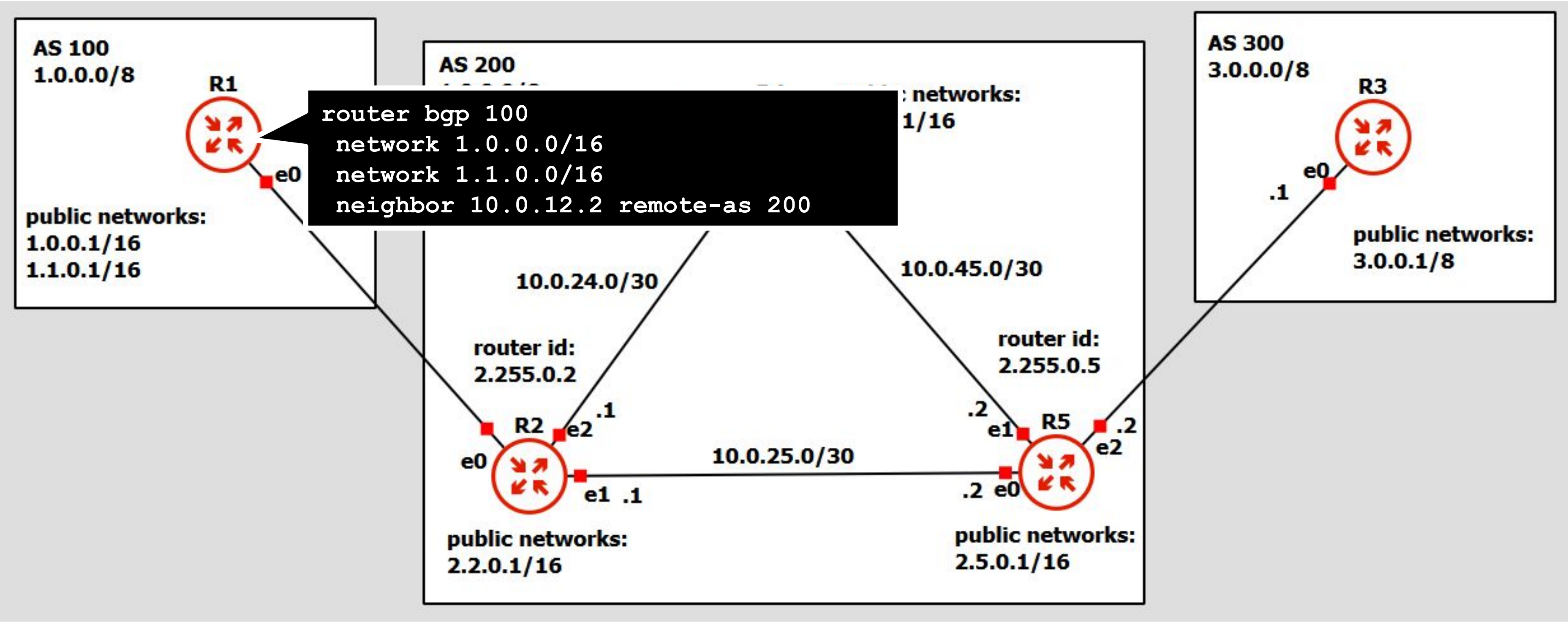

AS 100
1.0.0.0/8
R1
e0
.1
public networks:
1.0.0.1/16
1.1.0.1/16
AS 200
1.0.0.0/8
router id:
2.255.0.4
R4
public networks:
2.4.0.1/16
e0
e1
.1
10.0.24.0/30
10.0.45.0/30
router id:
2.255.0.2
.1
R2
e2
e0
e1
.1
10.0.25.0/30
router id:
2.255.0.5
.2
e1
R5
.2
e2
.2
e0
public networks:
2.2.0.1/16
public networks:
2.5.0.1/16
AS 300
3.0.0.0/8
R3
e0
.1
public networks:
3.0.0.1/8

IP configuration on R1



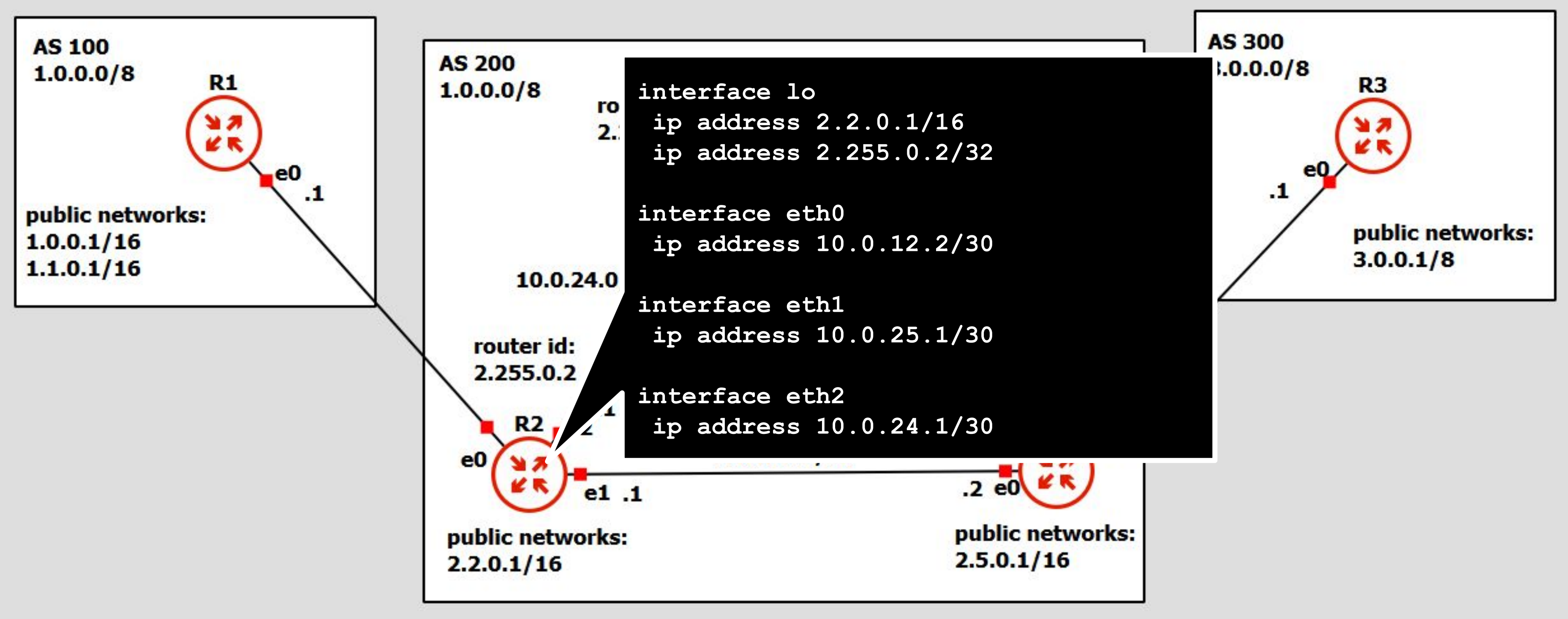
```
interface lo
  ip address 1.0.0.1/16
  ip address 1.1.0.1/16

interface eth0
 ip address 10.0.12.1/24
```

BGP configuration on R1



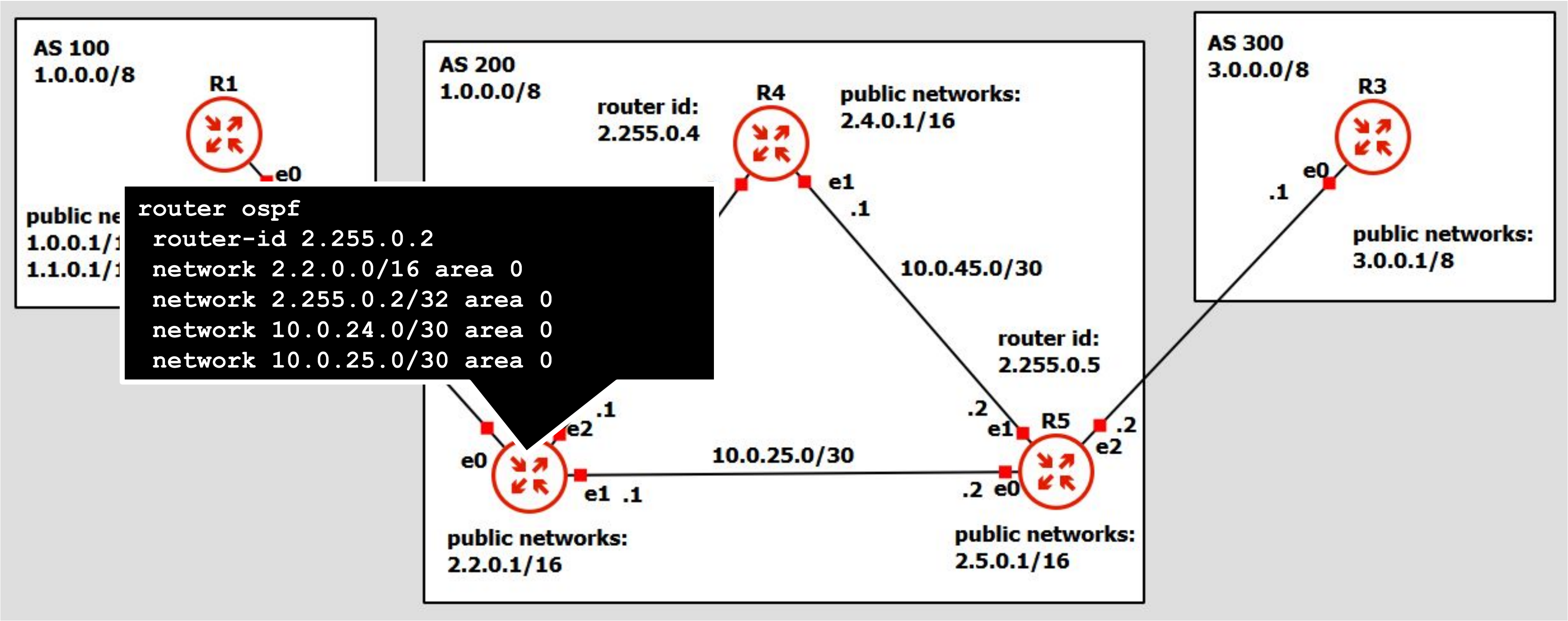
```
router bgp 100
 network 1.0.0.0/16
 network 1.1.0.0/16
 neighbor 10.0.12.2 remote-as 200
```

IP configuration on R2



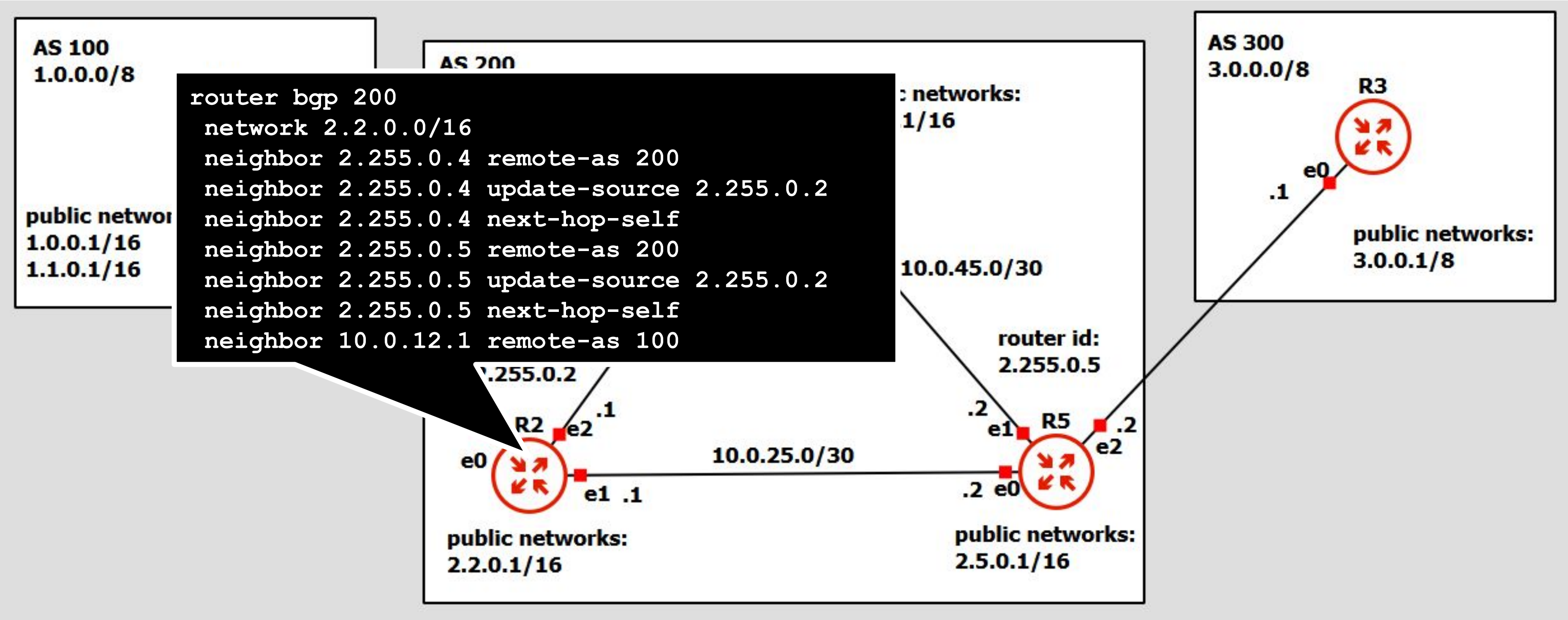
```
interface lo
 ip address 2.2.0.1/16
 ip address 2.255.0.2/32

interface eth0
 ip address 10.0.12.2/30

interface eth1
 ip address 10.0.25.1/30

interface eth2
 ip address 10.0.24.1/30
```

OSPF configuration on R2



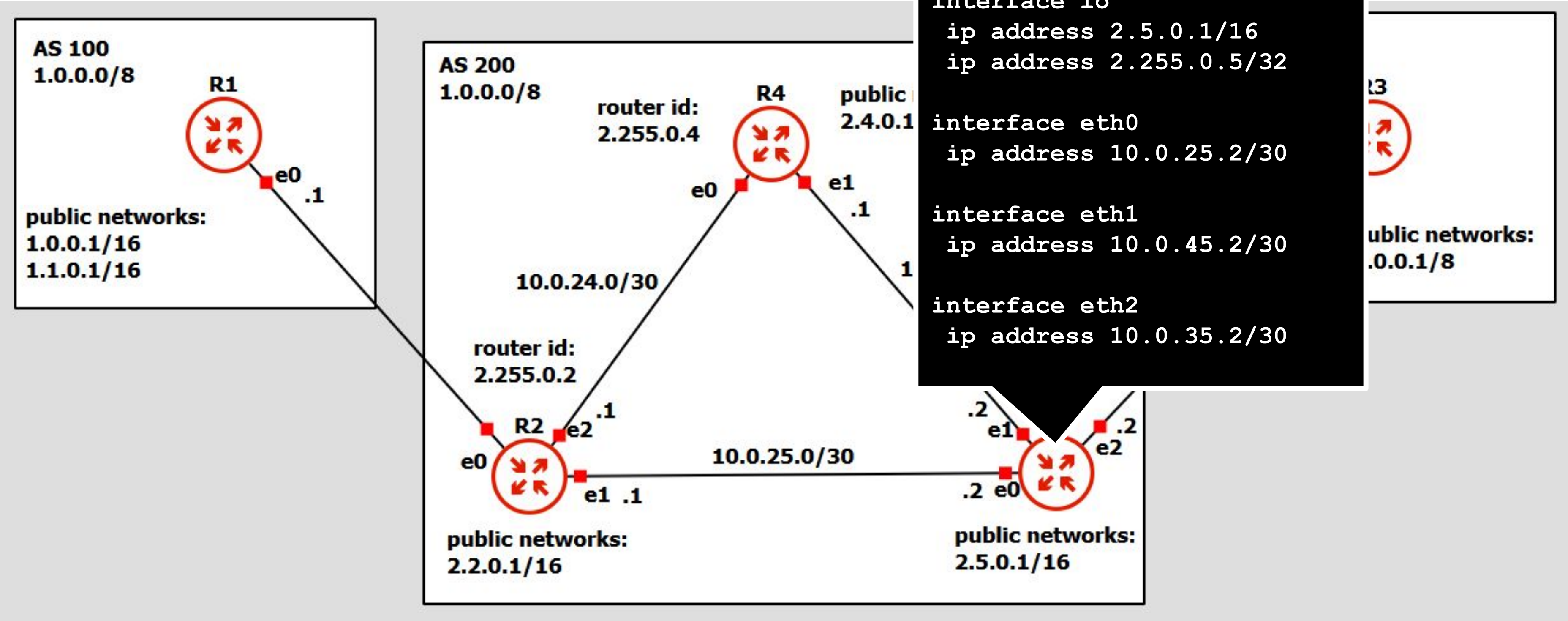
```
router ospf
 router-id 2.255.0.2
 network 2.2.0.0/16 area 0
 network 2.255.0.2/32 area 0
 network 10.0.24.0/30 area 0
 network 10.0.25.0/30 area 0
```

BGP configuration on R2



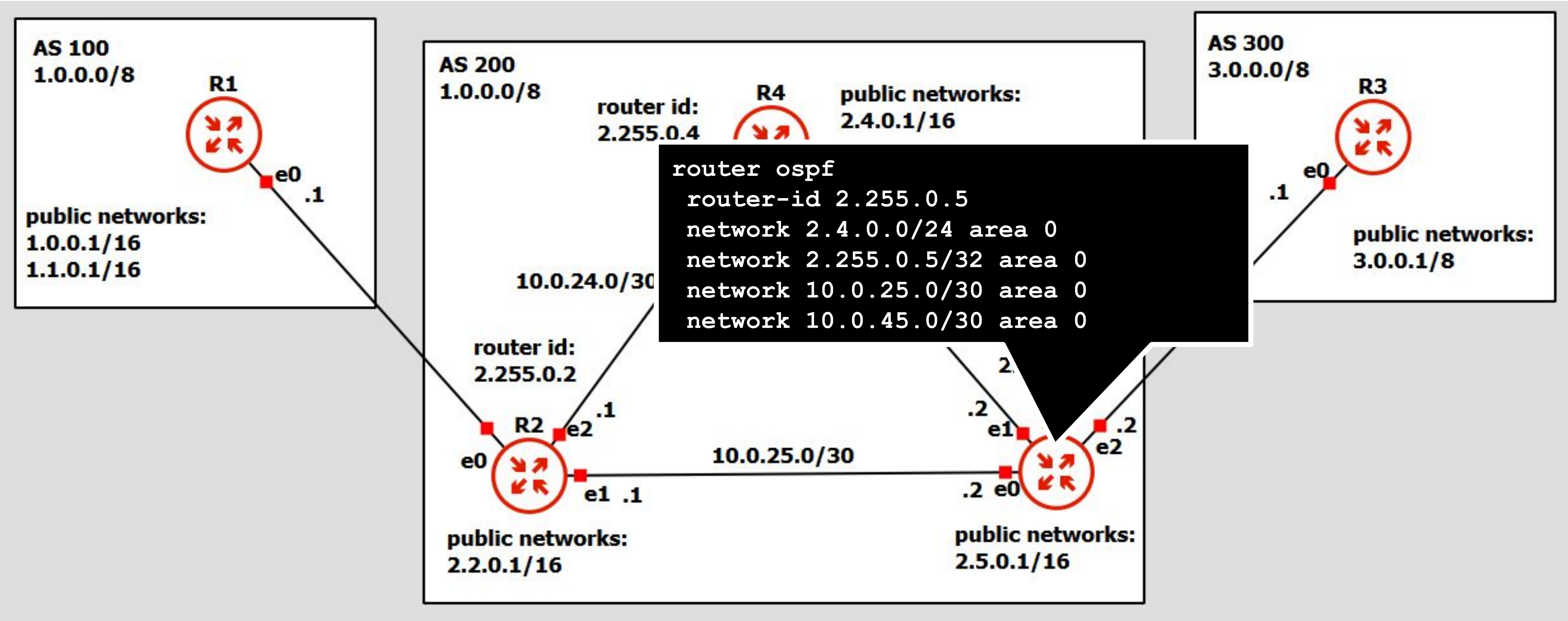
```
router bgp 200
 network 2.2.0.0/16
 neighbor 2.255.0.4 remote-as 200
 neighbor 2.255.0.4 update-source 2.255.0.2
 neighbor 2.255.0.4 next-hop-self
 neighbor 2.255.0.5 remote-as 200
 neighbor 2.255.0.5 update-source 2.255.0.2
 neighbor 2.255.0.5 next-hop-self
 neighbor 10.0.12.1 remote-as 100
```

IP configuration on R5



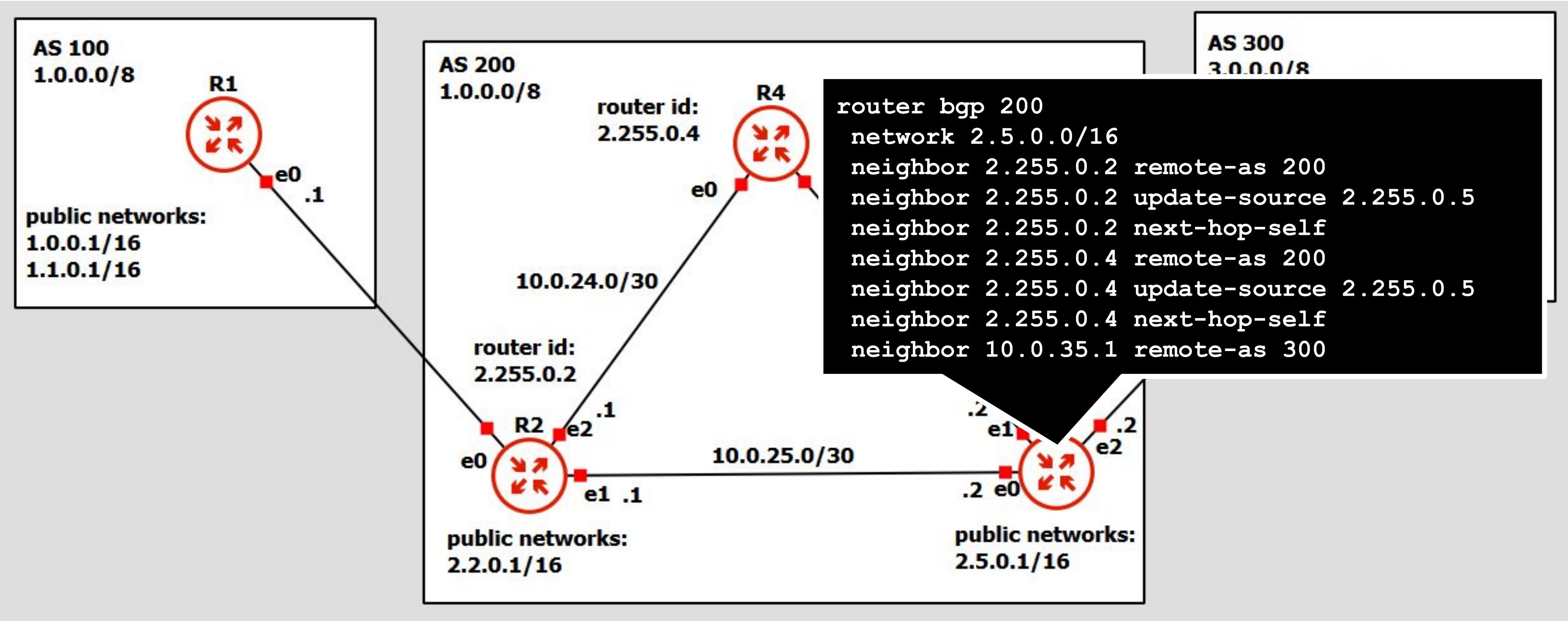
```
interface lo
 ip address 2.5.0.1/16
 ip address 2.255.0.5/32

interface eth0
 ip address 10.0.25.2/30

interface eth1
 ip address 10.0.45.2/30

interface eth2
 ip address 10.0.35.2/30
```

OSPF configuration on R5



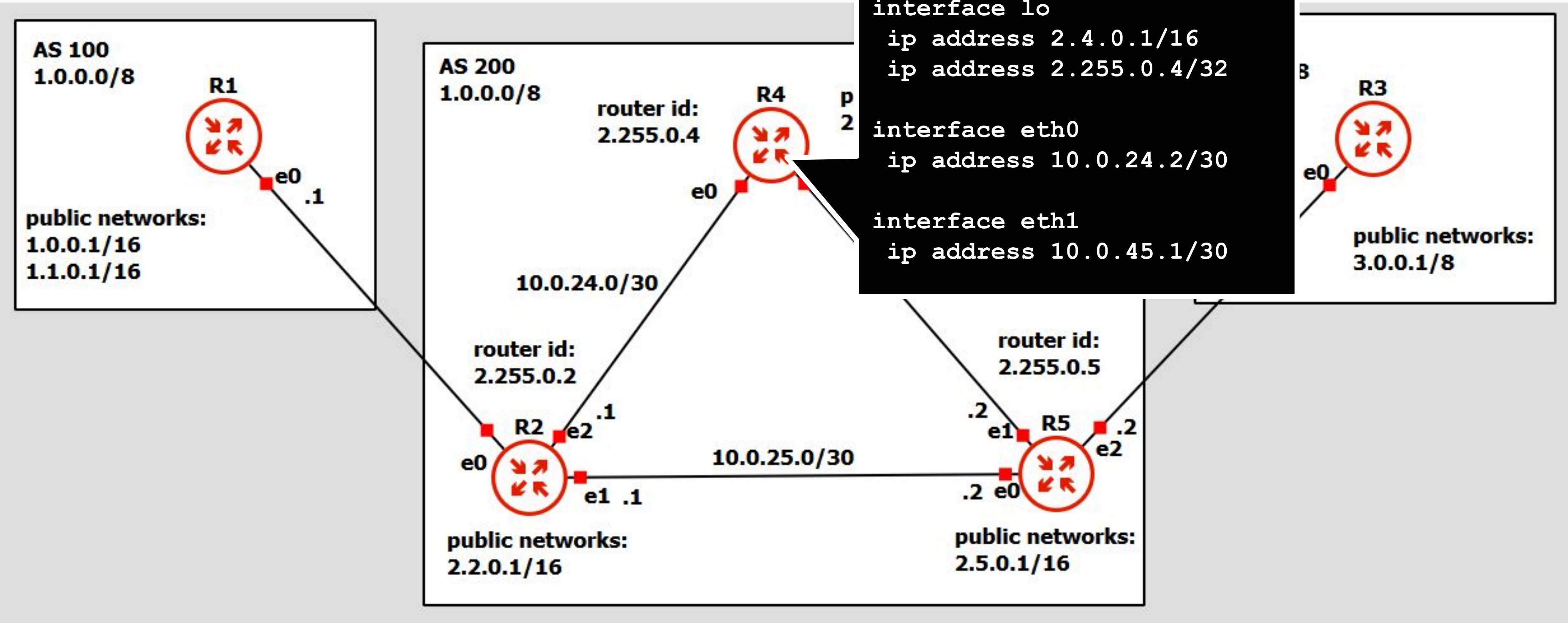
```
router ospf
 router-id 2.255.0.5
 network 2.4.0.0/24 area 0
 network 2.255.0.5/32 area 0
 network 10.0.25.0/30 area 0
 network 10.0.45.0/30 area 0
```

BGP configuration on R5



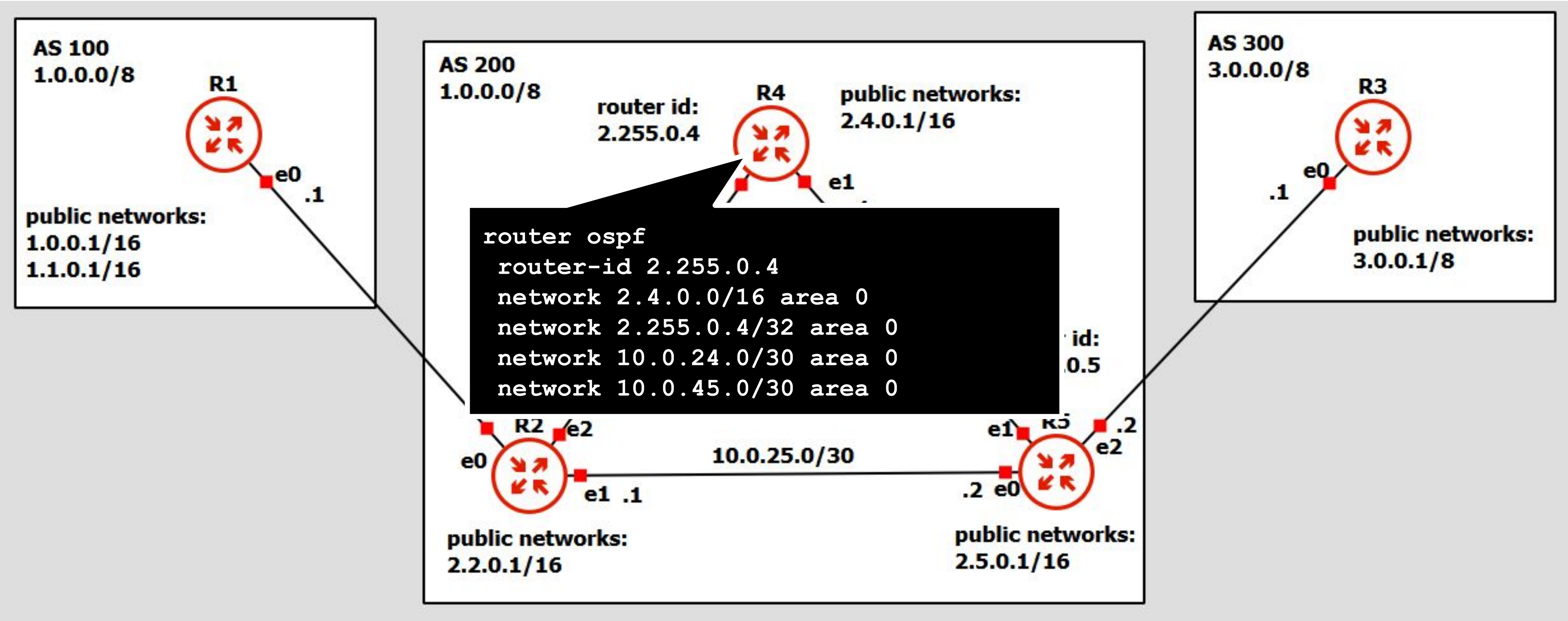
```
router bgp 200
 network 2.5.0.0/16
 neighbor 2.255.0.2 remote-as 200
 neighbor 2.255.0.2 update-source 2.255.0.5
 neighbor 2.255.0.2 next-hop-self
 neighbor 2.255.0.4 remote-as 200
 neighbor 2.255.0.4 update-source 2.255.0.5
 neighbor 2.255.0.4 next-hop-self
 neighbor 10.0.35.1 remote-as 300
```

IP configuration on R4



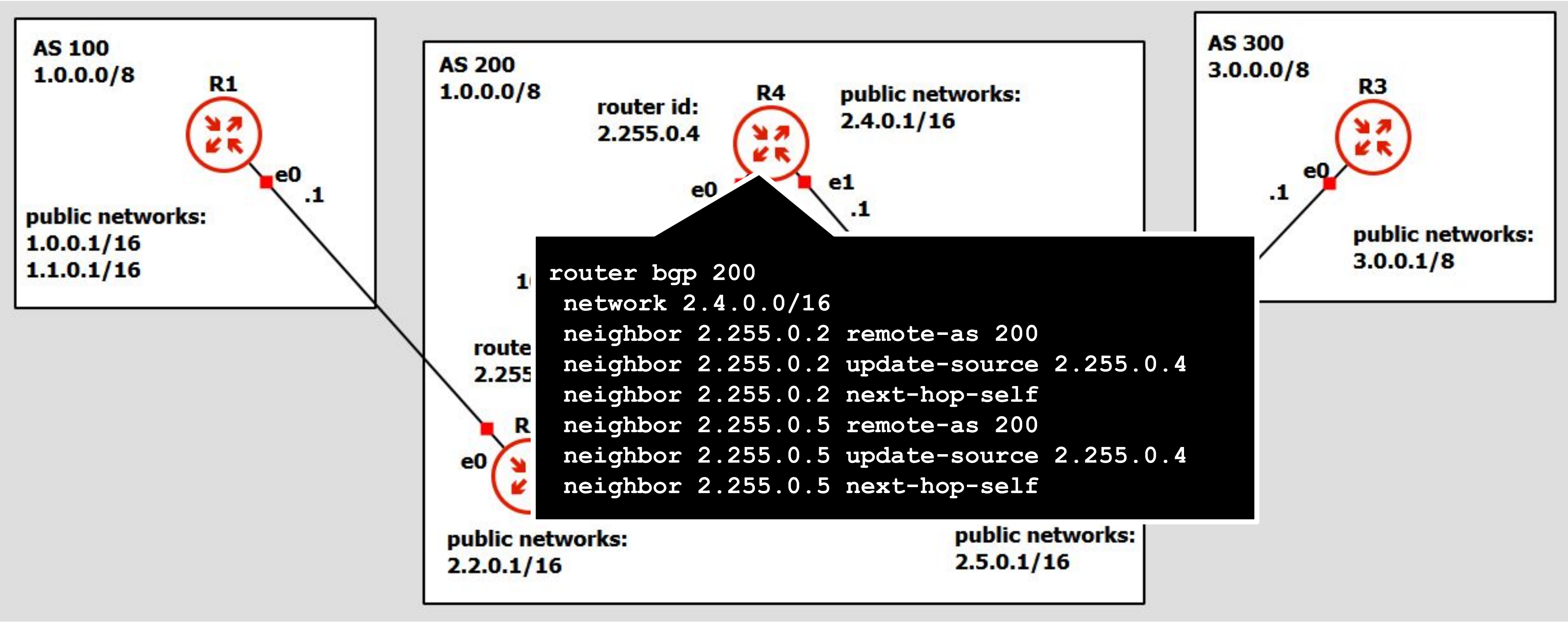
```
interface lo
 ip address 2.4.0.1/16
 ip address 2.255.0.4/32

interface eth0
 ip address 10.0.24.2/30

interface eth1
 ip address 10.0.45.1/30
```

OSPF configuration on R4



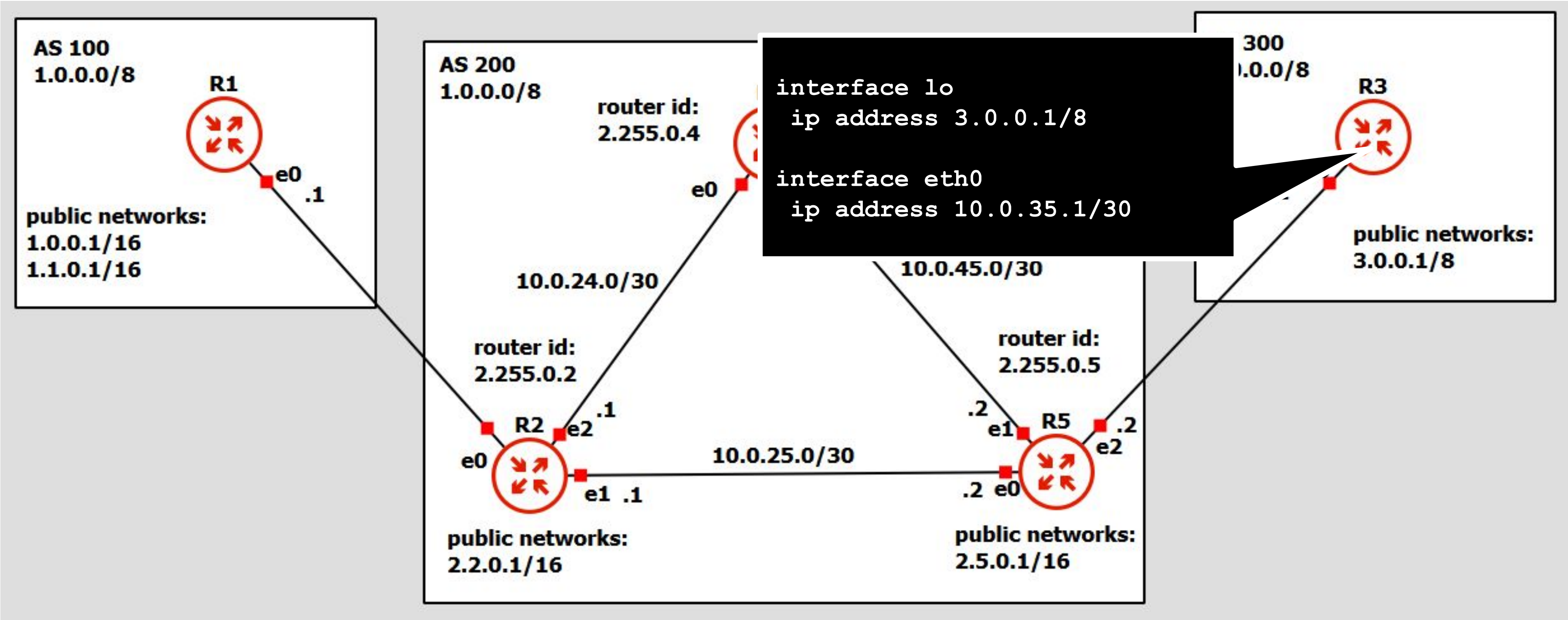
```
router ospf
 router-id 2.255.0.4
 network 2.4.0.0/16 area 0
 network 2.255.0.4/32 area 0
 network 10.0.24.0/30 area 0
 network 10.0.45.0/30 area 0
```

BGP configuration on R4



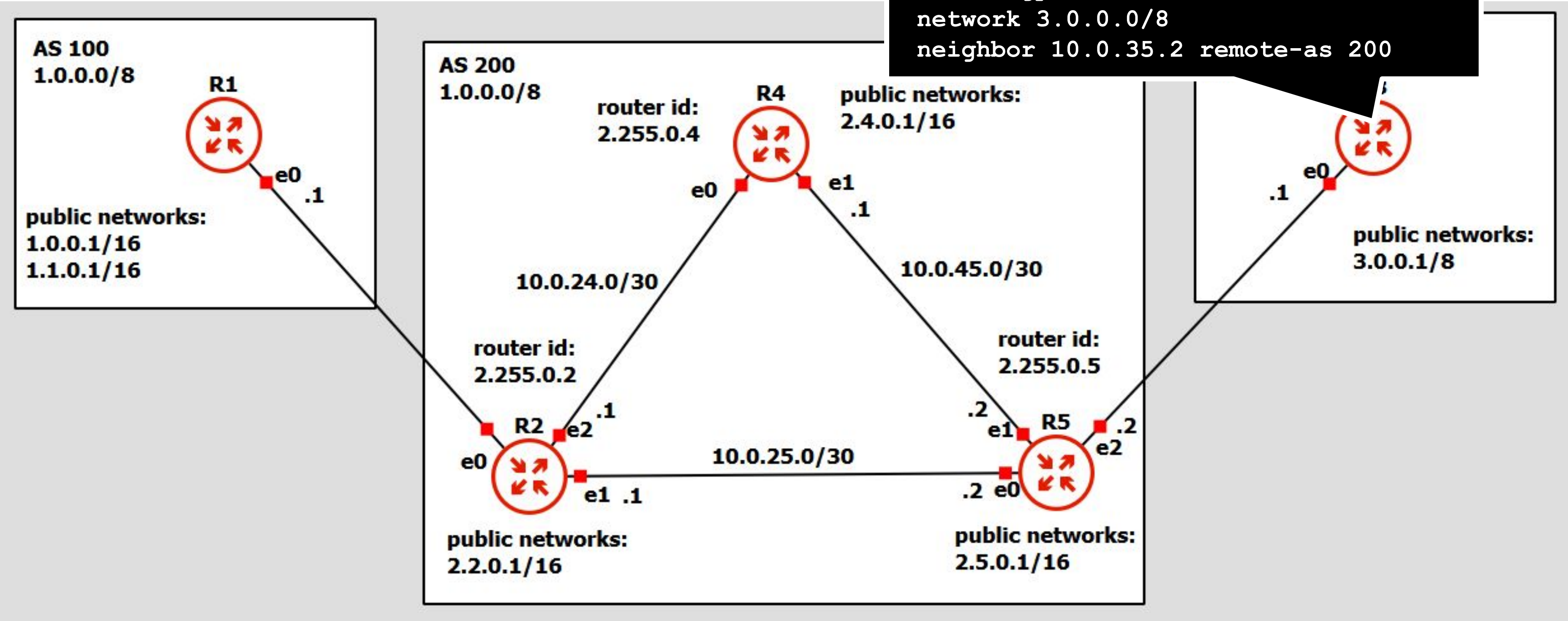
```
router bgp 200
 network 2.4.0.0/16
 neighbor 2.255.0.2 remote-as 200
 neighbor 2.255.0.2 update-source 2.255.0.4
 neighbor 2.255.0.2 next-hop-self
 neighbor 2.255.0.5 remote-as 200
 neighbor 2.255.0.5 update-source 2.255.0.4
 neighbor 2.255.0.5 next-hop-self
```

IP configuration on R3



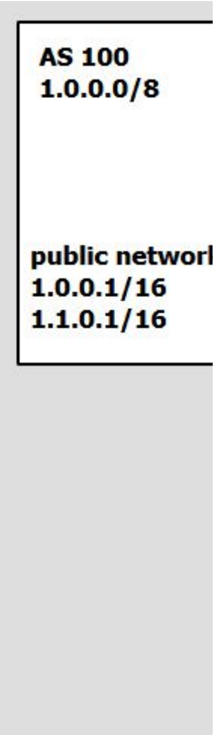
```
interface lo
 ip address 3.0.0.1/8

interface eth0
 ip address 10.0.35.1/30
```

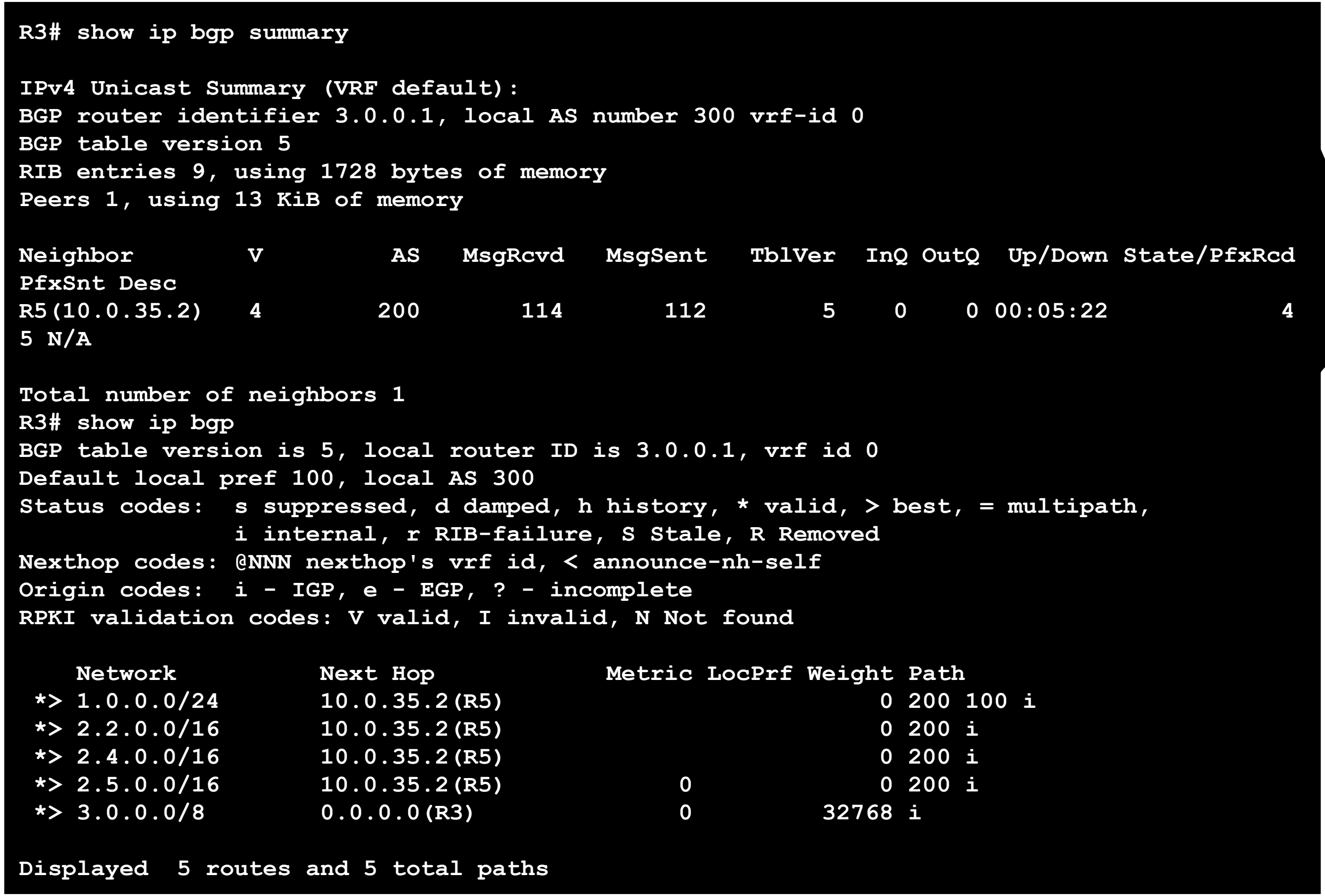
BGP configuration on R3



```
router bgp 300
 network 3.0.0.0/8
 neighbor 10.0.35.2 remote-as 200
```

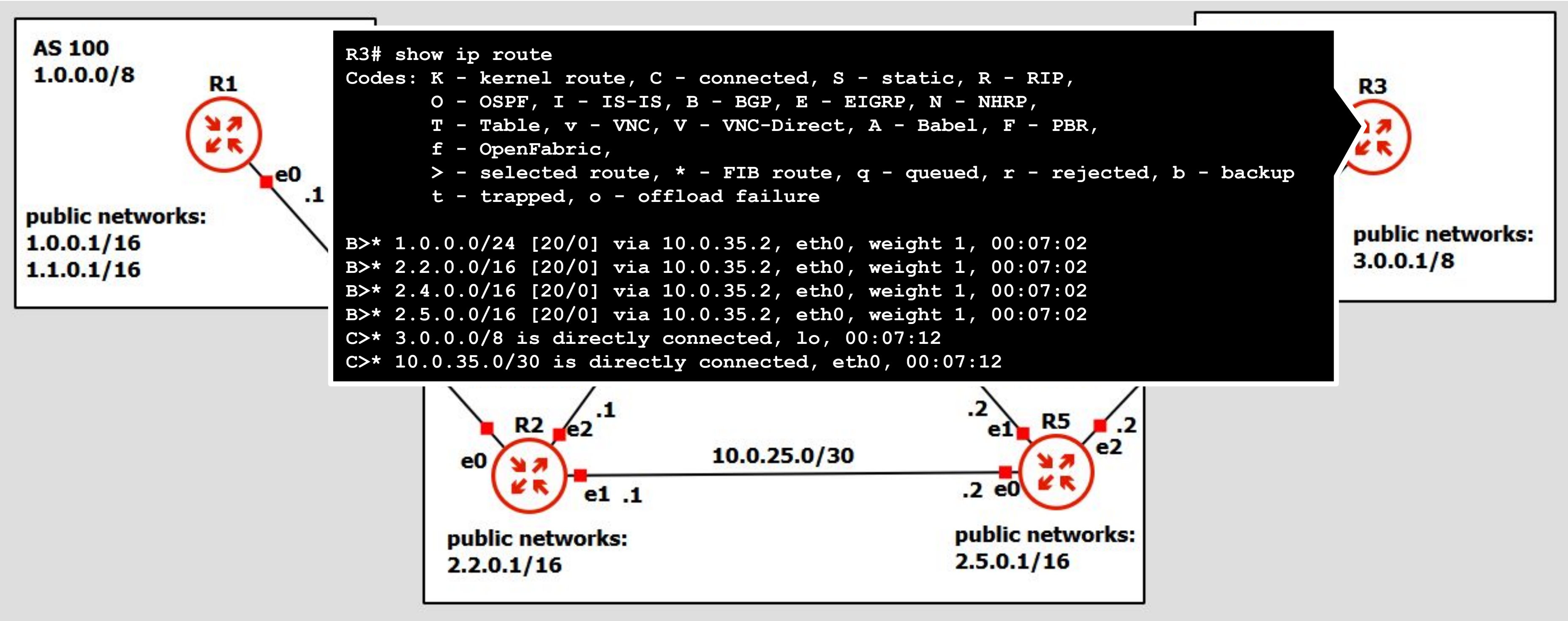
Test 1: check BGP peers and table in R3





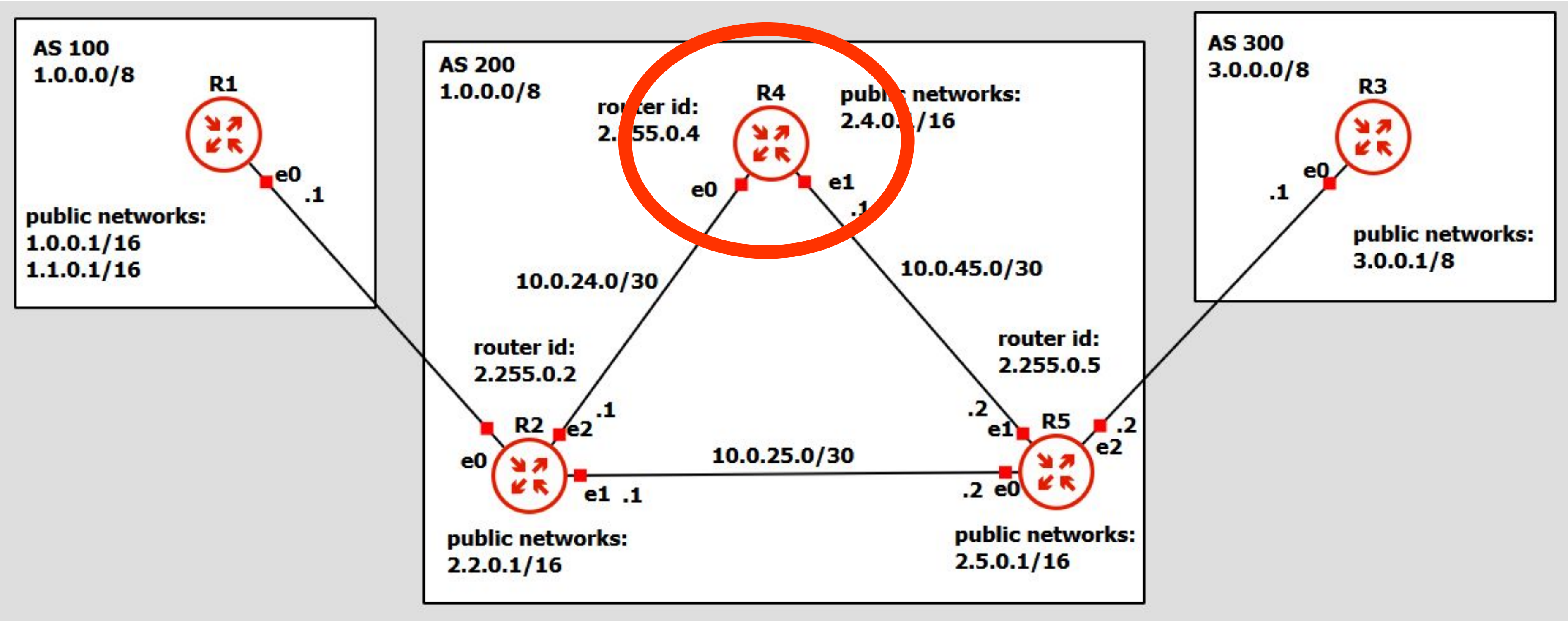
```
R3# show ip bgp summary

IPv4 Unicast Summary (VRF default):
BGP router identifier 3.0.0.1, local AS number 300 vrf-id 0
BGP table version 5
RIB entries 9, using 1728 bytes of memory
Peers 1, using 13 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd
PfxSnt Desc
R5(10.0.35.2)   4        200       114       112        5    0    0 00:05:22            4
5 N/A

Total number of neighbors 1
R3# show ip bgp
BGP table version is 5, local router ID is 3.0.0.1, vrf id 0
Default local pref 100, local AS 300
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *> 1.0.0.0/24       10.0.35.2(R5)                            0 200 100 i
 *> 2.2.0.0/16       10.0.35.2(R5)                            0 200 i
 *> 2.4.0.0/16       10.0.35.2(R5)                            0 200 i
 *> 2.5.0.0/16       10.0.35.2(R5)            0               0 200 i
 *> 3.0.0.0/8        0.0.0.0(R3)              0           32768 i

Displayed  5 routes and 5 total paths
```

Test case 1: check BGP peers and table in R3

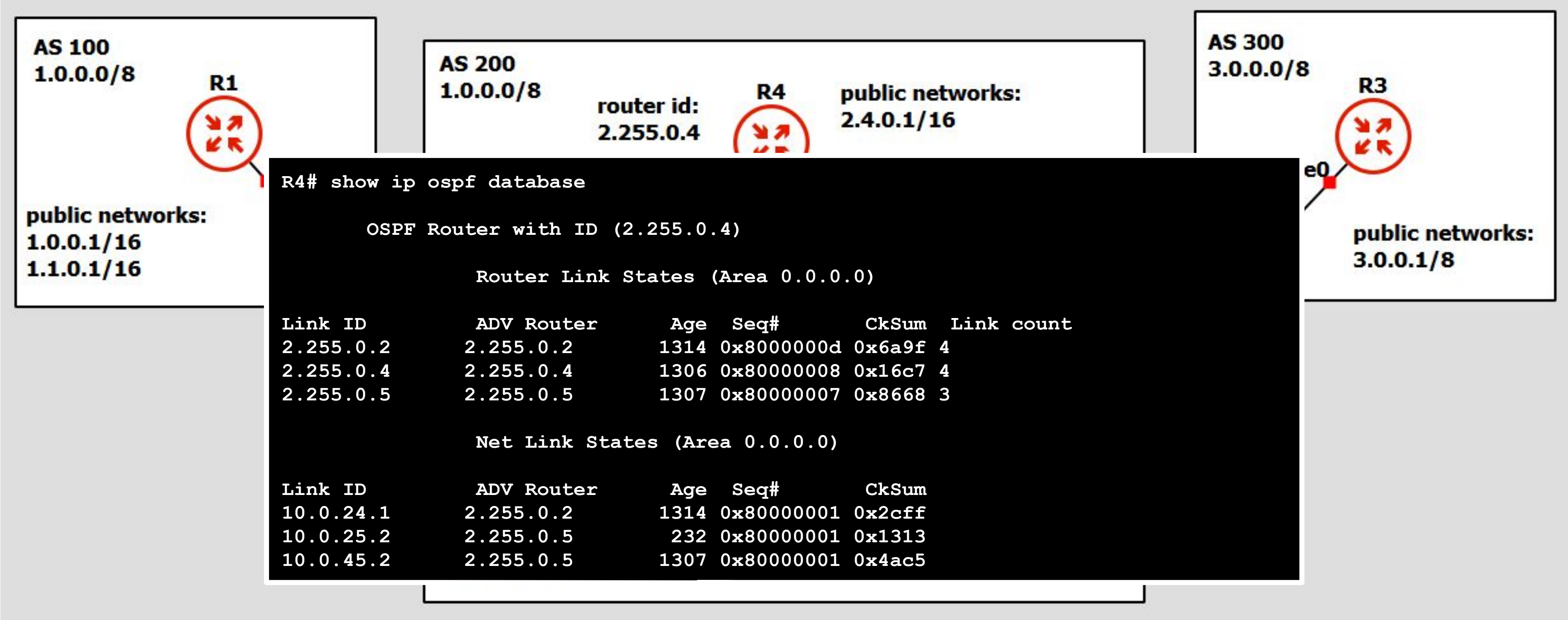


```
R3# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

B>* 1.0.0.0/24 [20/0] via 10.0.35.2, eth0, weight 1, 00:07:02
B>* 2.2.0.0/16 [20/0] via 10.0.35.2, eth0, weight 1, 00:07:02
B>* 2.4.0.0/16 [20/0] via 10.0.35.2, eth0, weight 1, 00:07:02
B>* 2.5.0.0/16 [20/0] via 10.0.35.2, eth0, weight 1, 00:07:02
C>* 3.0.0.0/8 is directly connected, lo, 00:07:12
C>* 10.0.35.0/30 is directly connected, eth0, 00:07:12
```

Test case 3: check R4

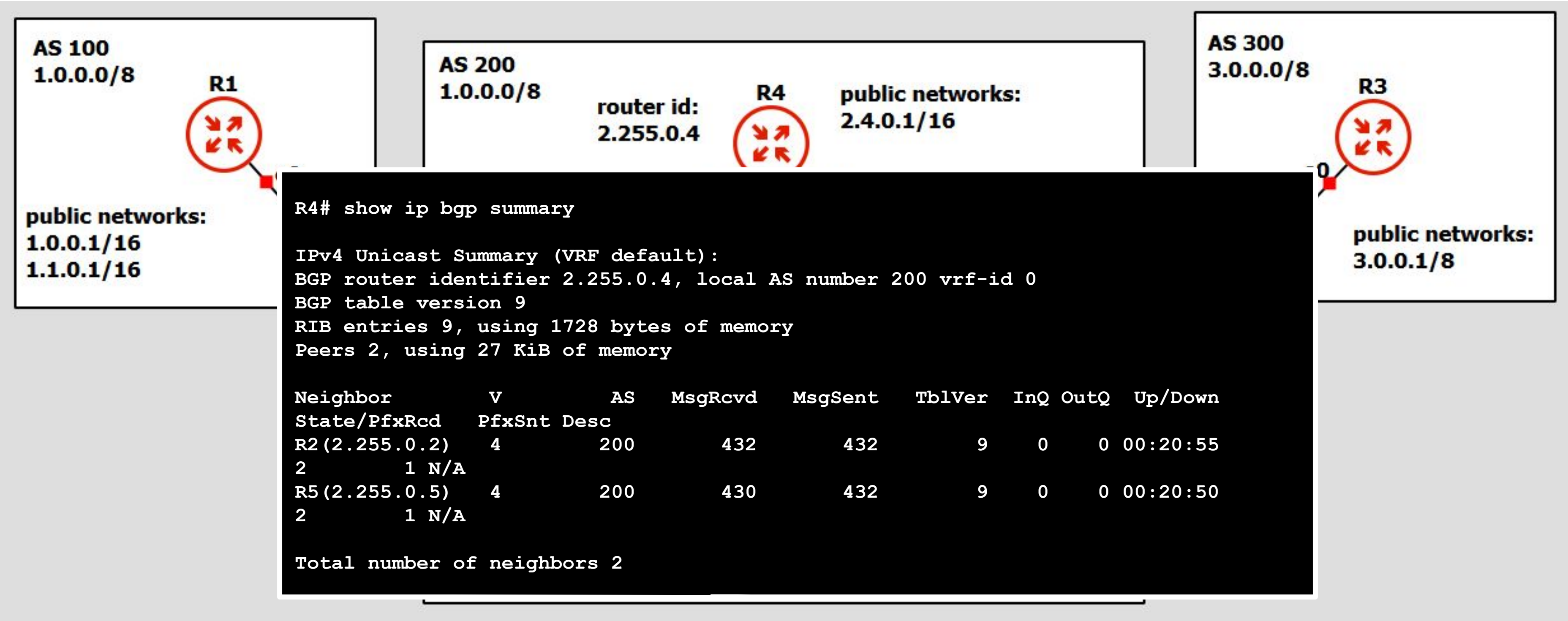


Test case 3: OSPF DB on R4



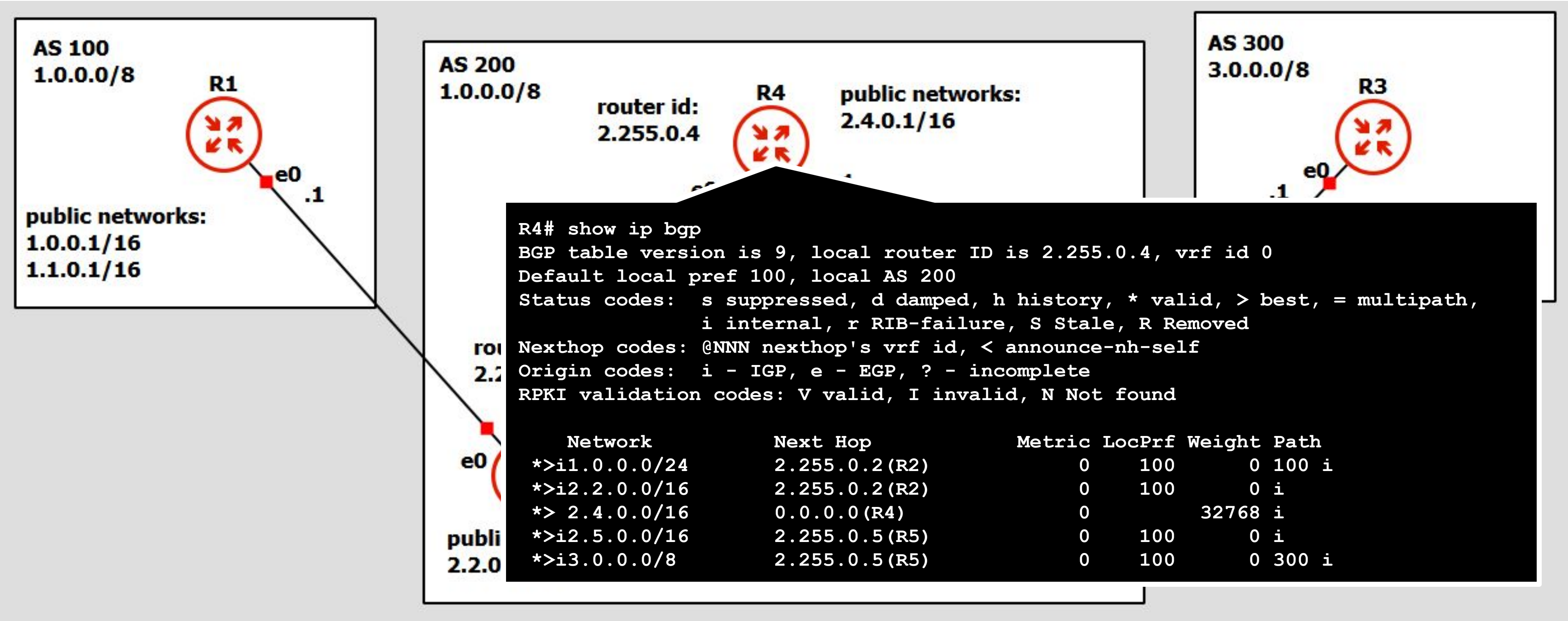
```
R4# show ip ospf database

       OSPF Router with ID (2.255.0.4)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
2.255.0.2      2.255.0.2       1314 0x8000000d 0x6a9f 4
2.255.0.4      2.255.0.4       1306 0x80000008 0x16c7 4
2.255.0.5      2.255.0.5       1307 0x80000007 0x8668 3

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.24.1      2.255.0.2       1314 0x80000001 0x2cff
10.0.25.2      2.255.0.5        232 0x80000001 0x1313
10.0.45.2      2.255.0.5       1307 0x80000001 0x4ac5
```

Test case 3: BGP summary on R4



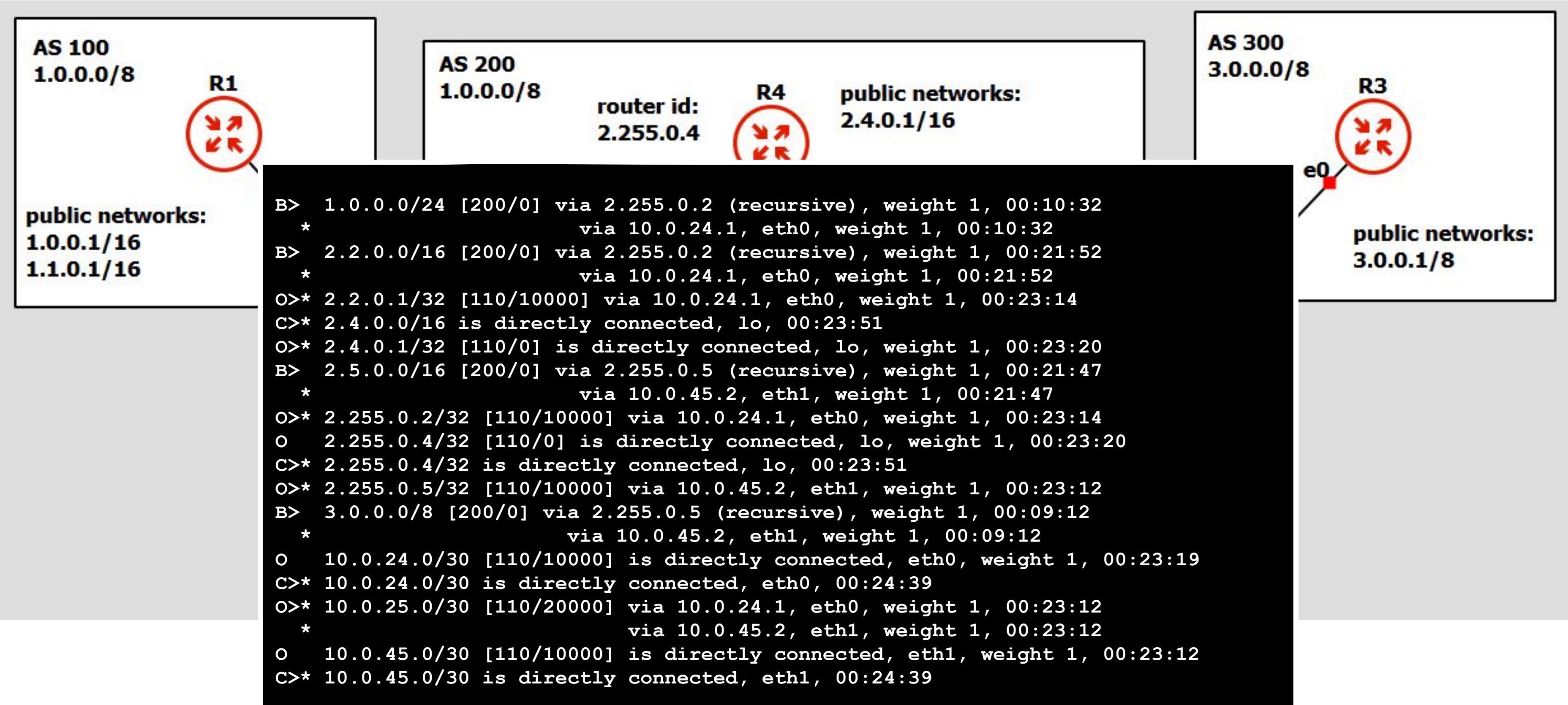
```
R4# show ip bgp summary

IPv4 Unicast Summary (VRF default):
BGP router identifier 2.255.0.4, local AS number 200 vrf-id 0
BGP table version 9
RIB entries 9, using 1728 bytes of memory
Peers 2, using 27 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down
State/PfxRcd   PfxSnt Desc
R2(2.255.0.2)   4        200       432       432        9    0    0 00:20:55
2         1 N/A
R5(2.255.0.5)   4        200       430       432        9    0    0 00:20:50
2         1 N/A

Total number of neighbors 2
```

Test case 3: BGP table on R4



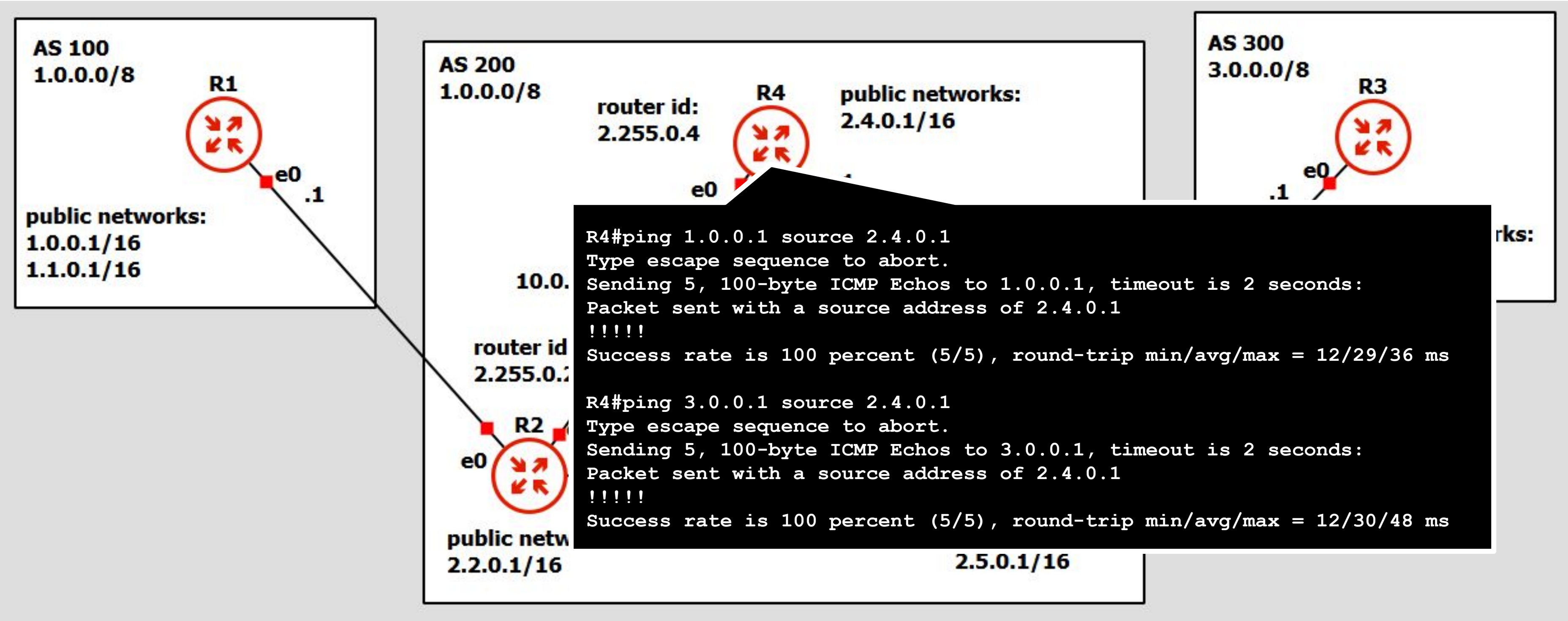
```
R4# show ip bgp
BGP table version is 9, local router ID is 2.255.0.4, vrf id 0
Default local pref 100, local AS 200
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *>i1.0.0.0/24       2.255.0.2(R2)            0    100      0 100 i
 *>i2.2.0.0/16       2.255.0.2(R2)            0    100      0 i
 *> 2.4.0.0/16       0.0.0.0(R4)              0         32768 i
 *>i2.5.0.0/16       2.255.0.5(R5)            0    100      0 i
 *>i3.0.0.0/8        2.255.0.5(R5)            0    100      0 300 i
```

Test 3: Routing table on R4



```
B>  1.0.0.0/24 [200/0] via 2.255.0.2 (recursive), weight 1, 00:10:32
  *                      via 10.0.24.1, eth0, weight 1, 00:10:32
B>  2.2.0.0/16 [200/0] via 2.255.0.2 (recursive), weight 1, 00:21:52
  *                      via 10.0.24.1, eth0, weight 1, 00:21:52
O>* 2.2.0.1/32 [110/10000] via 10.0.24.1, eth0, weight 1, 00:23:14
C>* 2.4.0.0/16 is directly connected, lo, 00:23:51
O>* 2.4.0.1/32 [110/0] is directly connected, lo, weight 1, 00:23:20
B>  2.5.0.0/16 [200/0] via 2.255.0.5 (recursive), weight 1, 00:21:47
  *                      via 10.0.45.2, eth1, weight 1, 00:21:47
O>* 2.255.0.2/32 [110/10000] via 10.0.24.1, eth0, weight 1, 00:23:14
O   2.255.0.4/32 [110/0] is directly connected, lo, weight 1, 00:23:20
C>* 2.255.0.4/32 is directly connected, lo, 00:23:51
O>* 2.255.0.5/32 [110/10000] via 10.0.45.2, eth1, weight 1, 00:23:12
B>  3.0.0.0/8 [200/0] via 2.255.0.5 (recursive), weight 1, 00:09:12
  *                     via 10.0.45.2, eth1, weight 1, 00:09:12
O   10.0.24.0/30 [110/10000] is directly connected, eth0, weight 1, 00:23:19
C>* 10.0.24.0/30 is directly connected, eth0, 00:24:39
O>* 10.0.25.0/30 [110/20000] via 10.0.24.1, eth0, weight 1, 00:23:12
  *                          via 10.0.45.2, eth1, weight 1, 00:23:12
O   10.0.45.0/30 [110/10000] is directly connected, eth1, weight 1, 00:23:12
C>* 10.0.45.0/30 is directly connected, eth1, 00:24:39
```

Test 3: Ping from R4



```
R4#ping 1.0.0.1 source 2.4.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.0.0.1, timeout is 2 seconds:
Packet sent with a source address of 2.4.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/29/36 ms

R4#ping 3.0.0.1 source 2.4.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.0.0.1, timeout is 2 seconds:
Packet sent with a source address of 2.4.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/30/48 ms
```