Setting up iBGP

BGP for networks who peer: Part 3

Wolfgang Tremmel
wolfgang.tremmel@de-cix.net

BGP (new) Webinars Overview

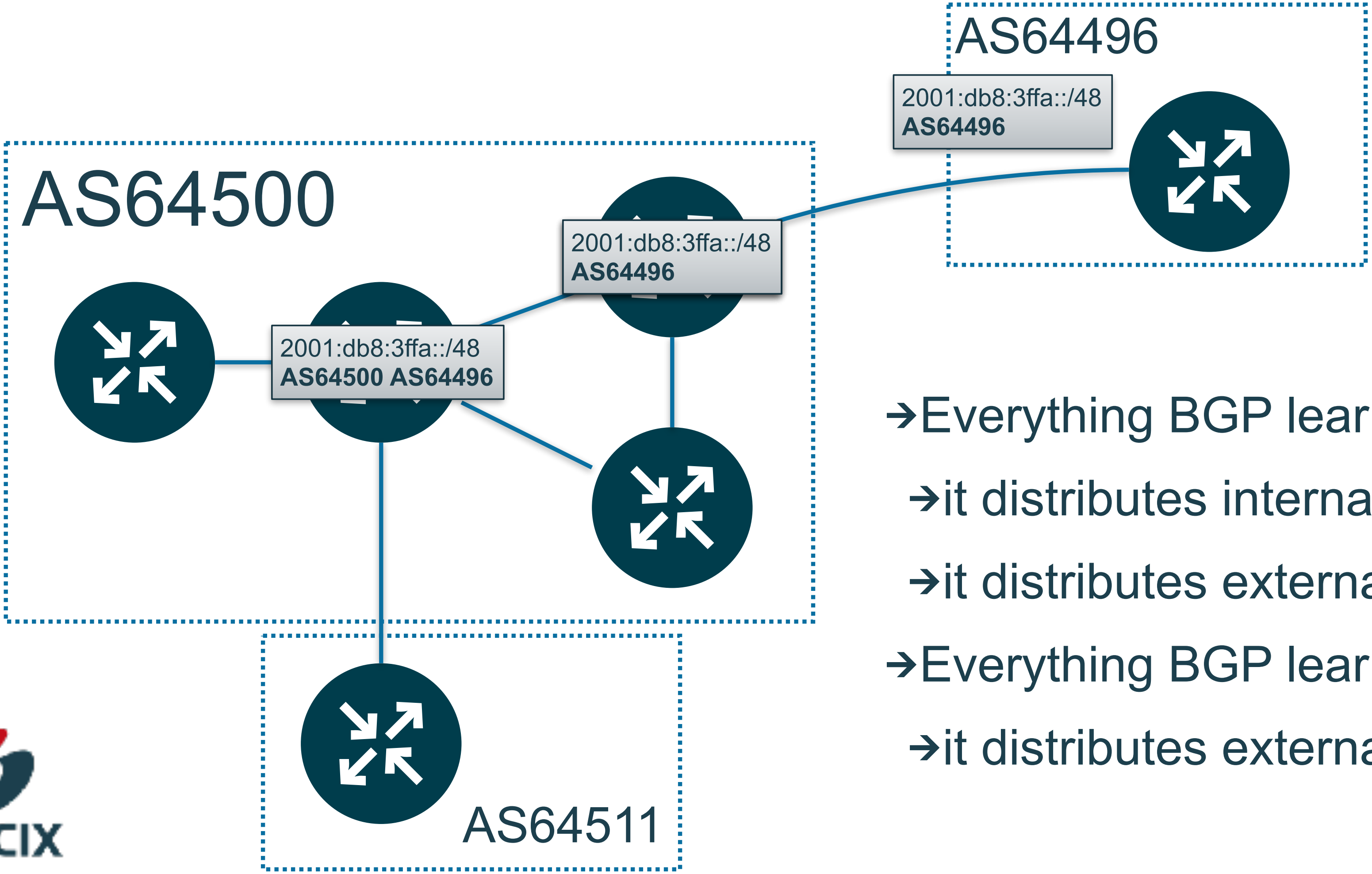
- → 01 - Prefixes and AS numbers
- → 02 - BGP Introduction
- → 03a - Setting up iBGP
- → 03b - Setting up eBGP
- → ...

BGP - not re-inventing the wheel

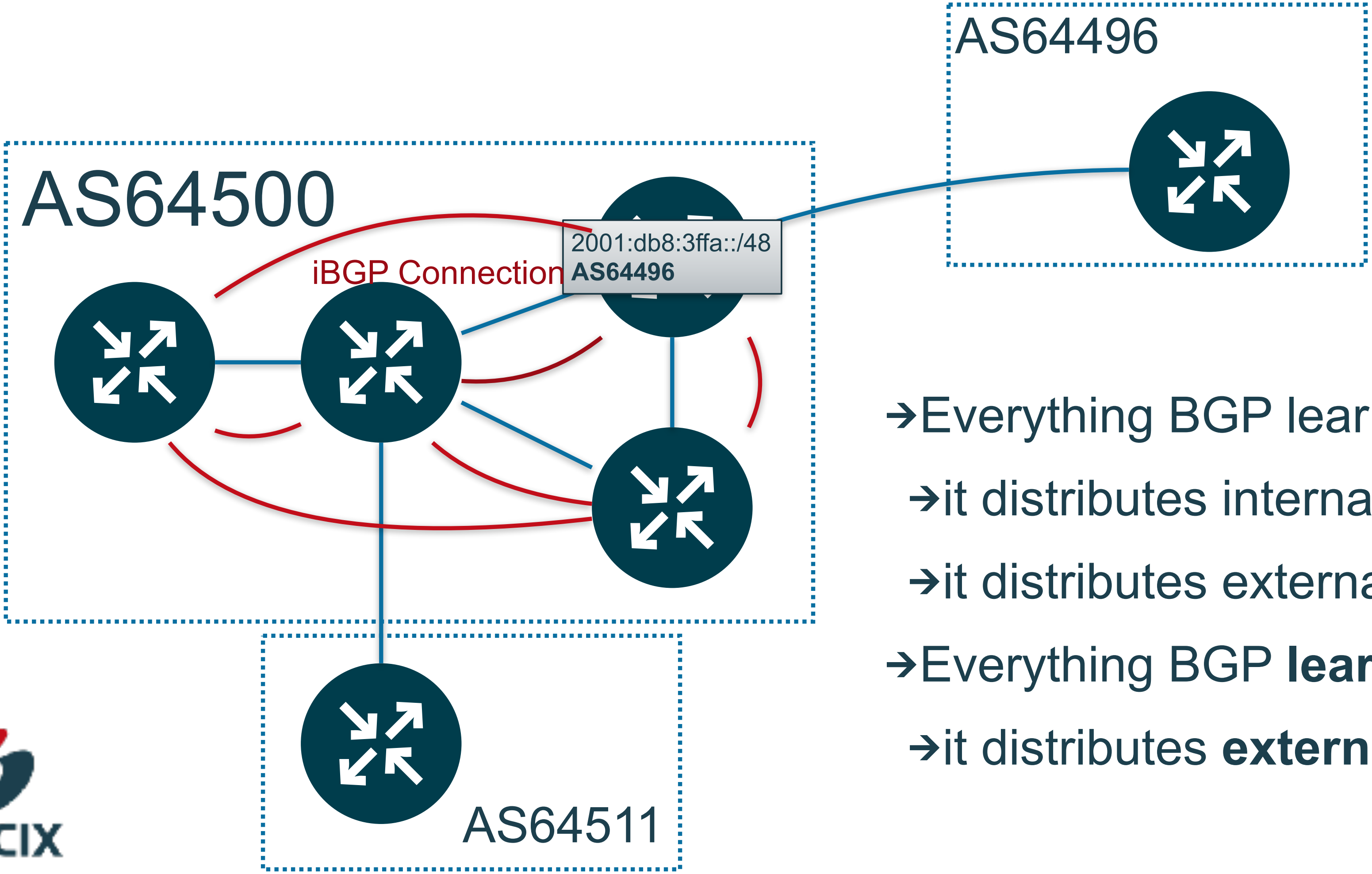
- →BGP uses TCP for transport
- →so no need to re-implement features TCP already provides, like
  - →reliable transport
  - →flow control
  - →framing
- →as long as the TCP session is up, BGP assumes its neighbors are up
  - →and have all the information sent to them

BGP - (re)distributing prefixes

- →a BGP speaking router
  - →learns prefixes
  - →distributes prefixes to its BGP neighbors
- →Everything BGP learns from external
  - →it distributes internal
  - →it distributes external
- →Everything BGP learns from internal
  - →it distributes external



- →Everything BGP learns from external
  - →it distributes internal
  - →it distributes external
- →Everything BGP learns from internal
  - →it distributes external



→Everything BGP learns from external

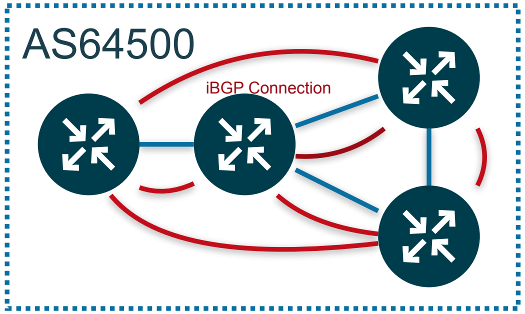
→it distributes internal

→it distributes external

→Everything BGP **learns from internal**

→it distributes **external**

Which routers do need to speak BGP?



- →Thats a hard question
- →Depends on your network design
- →In general:
  - →all router with an **outside connection** to another AS
  - →big core routers inside your network
    - →but they might not need the full routing table
    - →a default-route to the nearest exit might do
    - →you can also distribute a default route via (i)BGP

BGP - not re-inventing the wheel

→BGP uses **TCP for transport**

→so no need to re-implement features TCP already provides, like

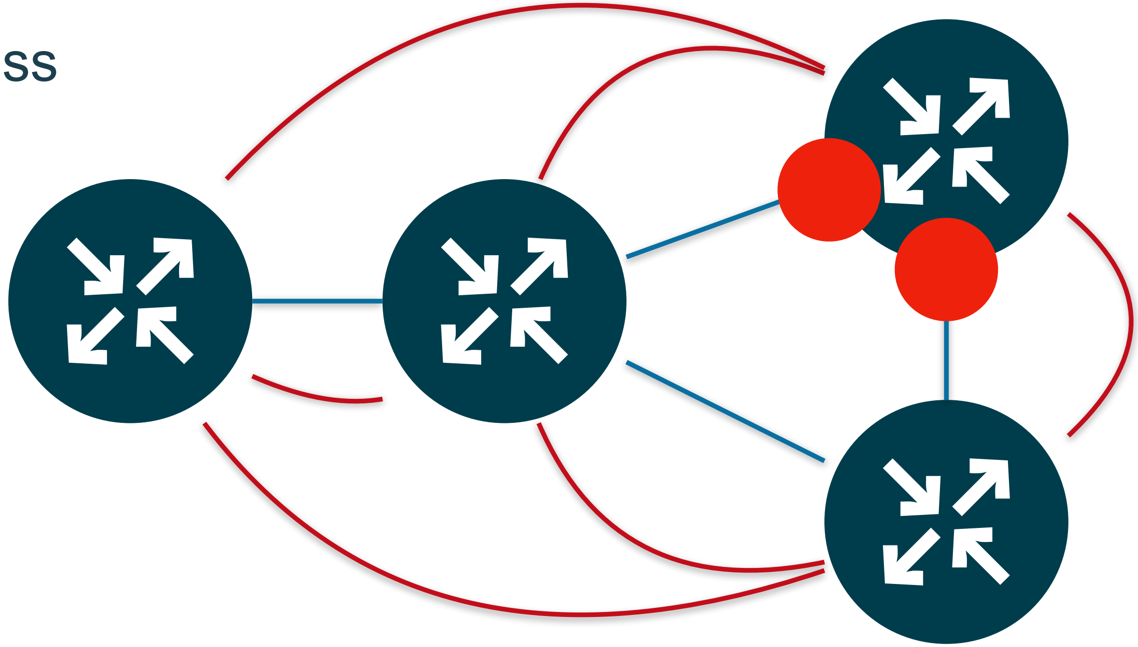
- →reliable transport
- →flow control
- →framing

→iBGP needs for session setup (via TCP)

- →a **source IP** address
- →a **destination IP** address

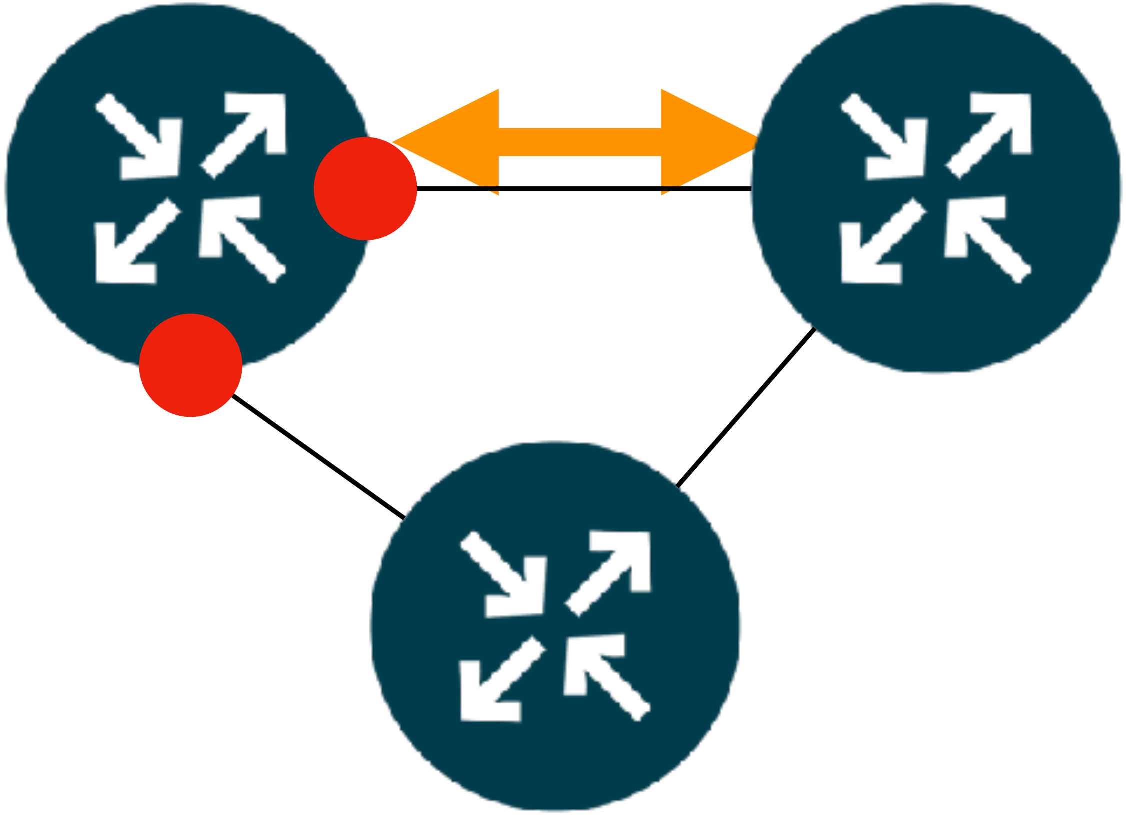
iBGP Session Setup - addressing

- →iBGP needs for session setup (via TCP)
  - →a **source IP** address
  - →a **destination IP** address

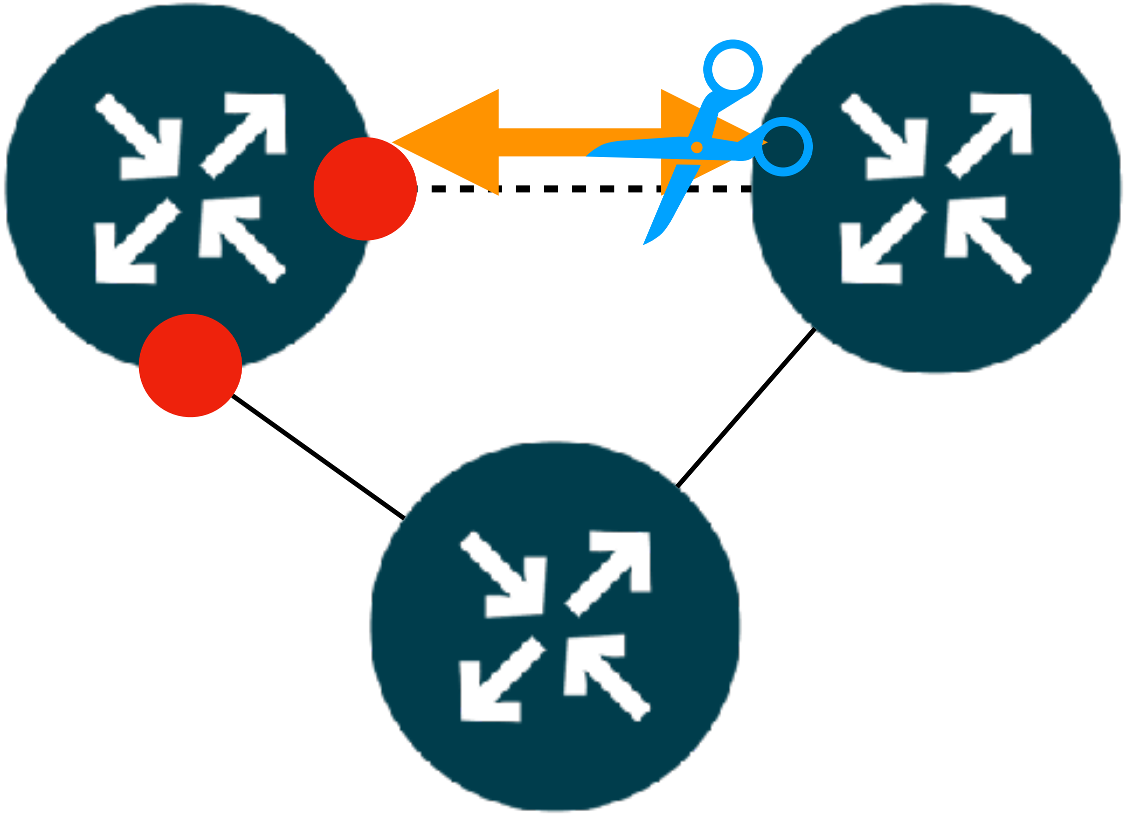


- →**Source IP**: Which one?
- → Default:
  - →address of the interface on which the **packets leave the router**

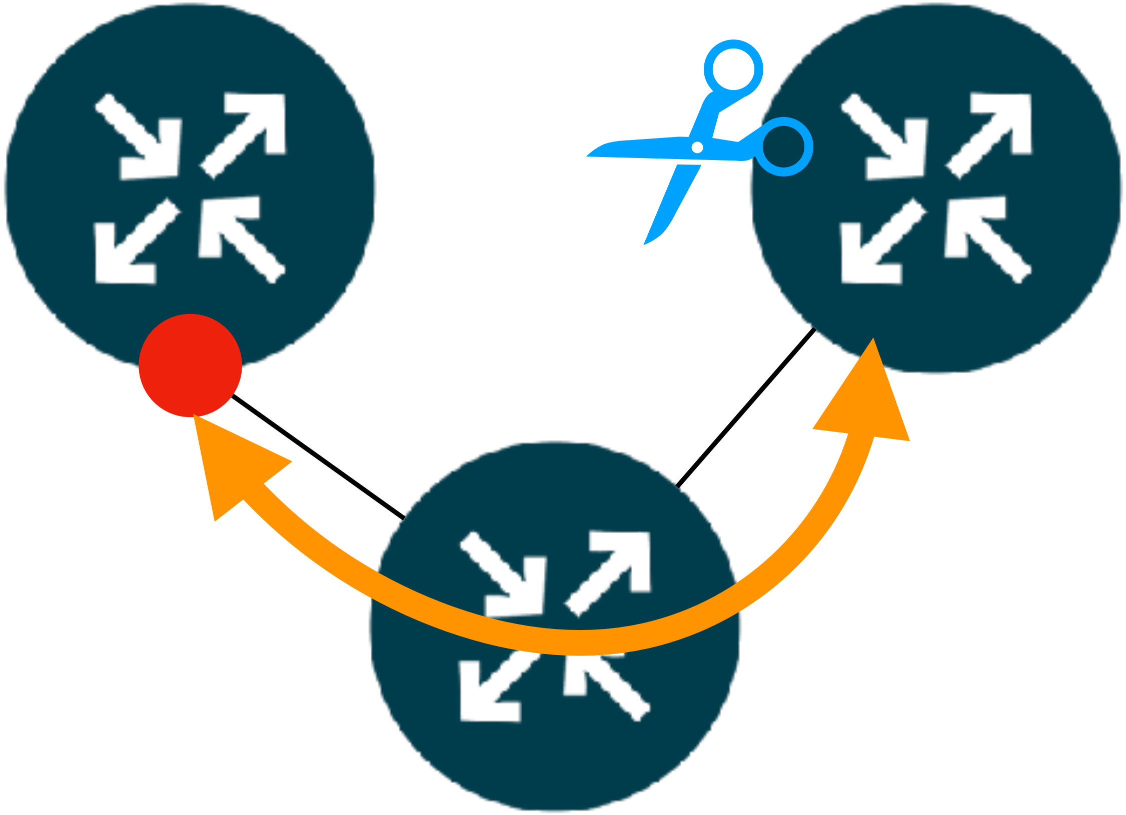
iBGP - Which IP addresses to use?



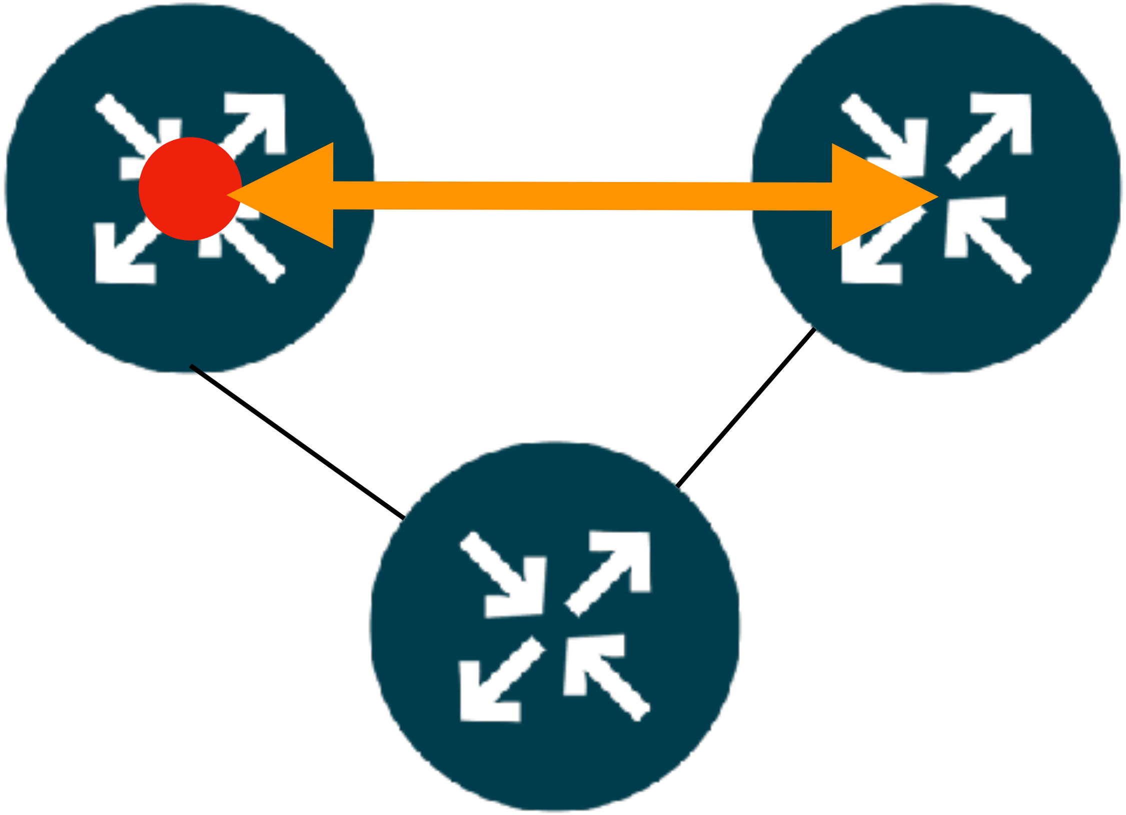
iBGP - Which IP addresses to use?



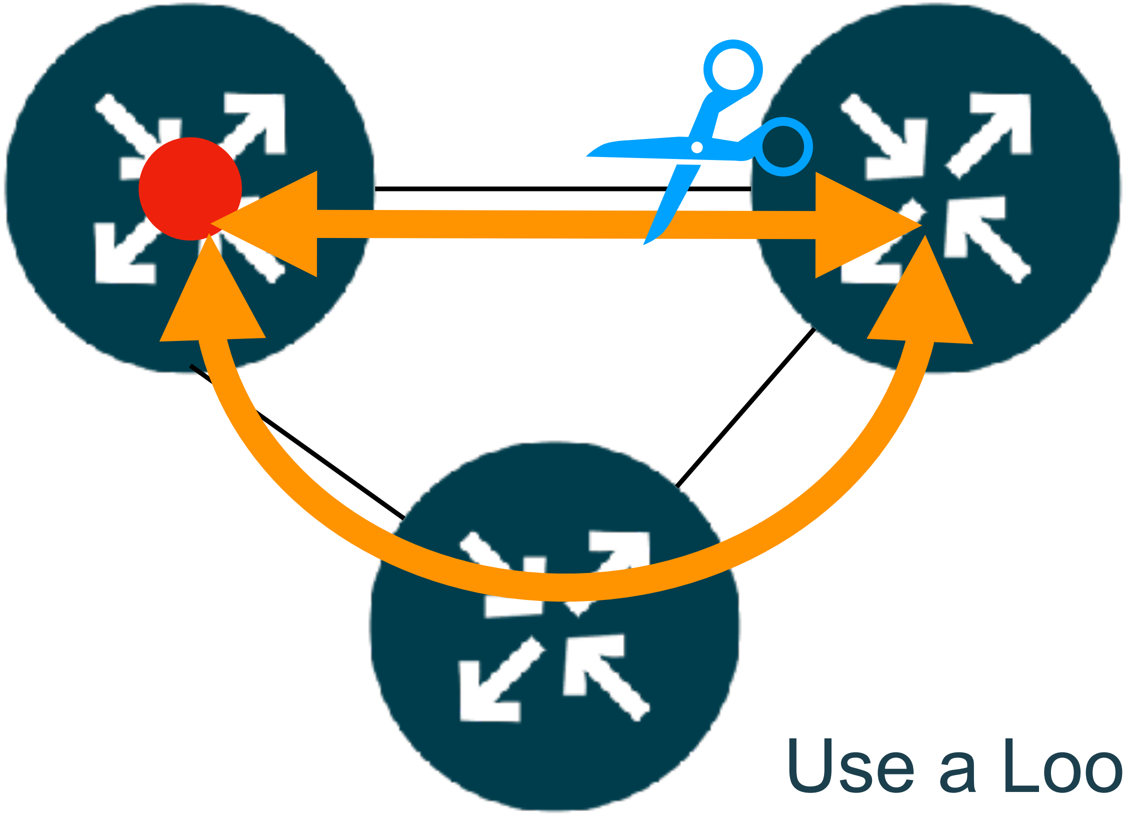
iBGP - Which IP addresses to use?



iBGP - Which IP addresses to use?



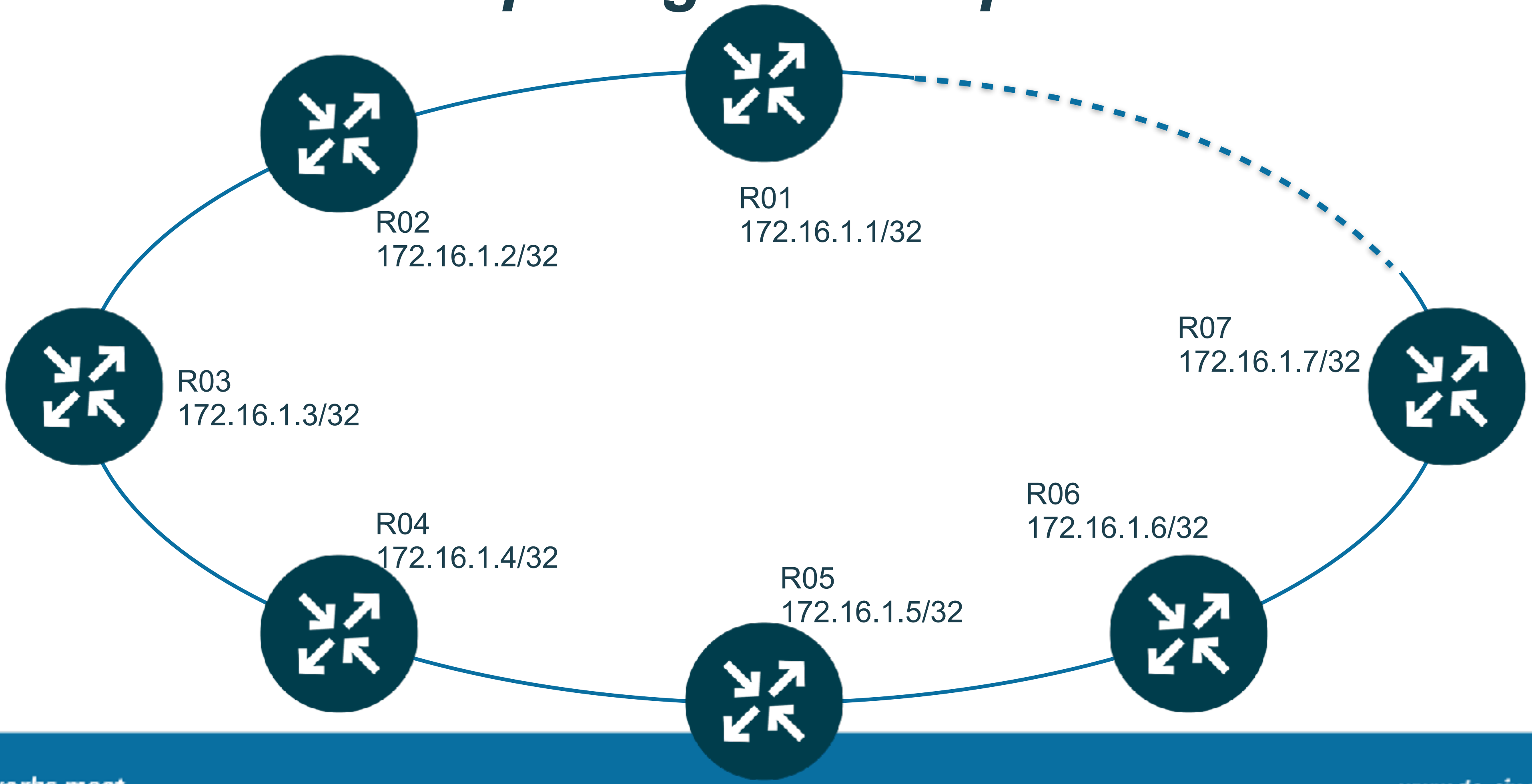
iBGP - Which IP addresses to use?



Use a Loopback address!

DE CIX

Network setup: Logical Setup



R02
172.16.1.2/32

R01
172.16.1.1/32

R07
172.16.1.7/32

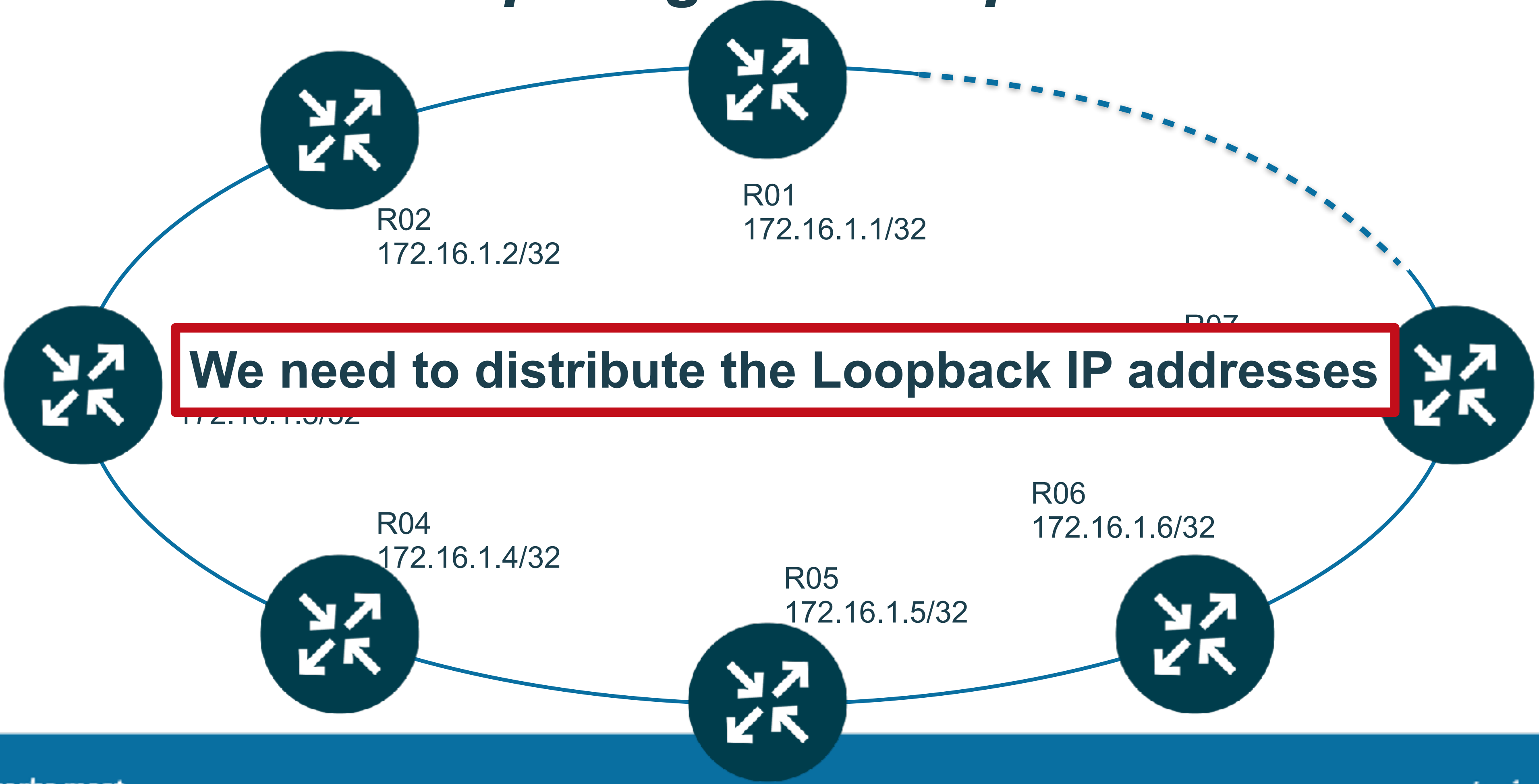
R03
172.16.1.3/32

R06
172.16.1.6/32

R04
172.16.1.4/32

R05
172.16.1.5/32

Network setup: Logical Setup



We need to distribute the Loopback IP addresses

We need to distribute the Loopback IP addresses

- →For this we need (another) routing protocol
- →OSPF - Open Shortest Path First
  - →works on IPv4 only
  - →still widely used
- →OSPFv3
  - →works on IPv6 link-local addressing
- →IS-IS
  - →is truly protocol independent, works on Layer 2 directly
- →Static routing
  - →yes, this also works but does not scale

Use OSPFv2 + OSPFv3 or IS-IS

→Most of the time not your choice

→In an existing network you have to use what's there

→...and what is supported best by your routers...

→Clean slate installation: Use IS-IS

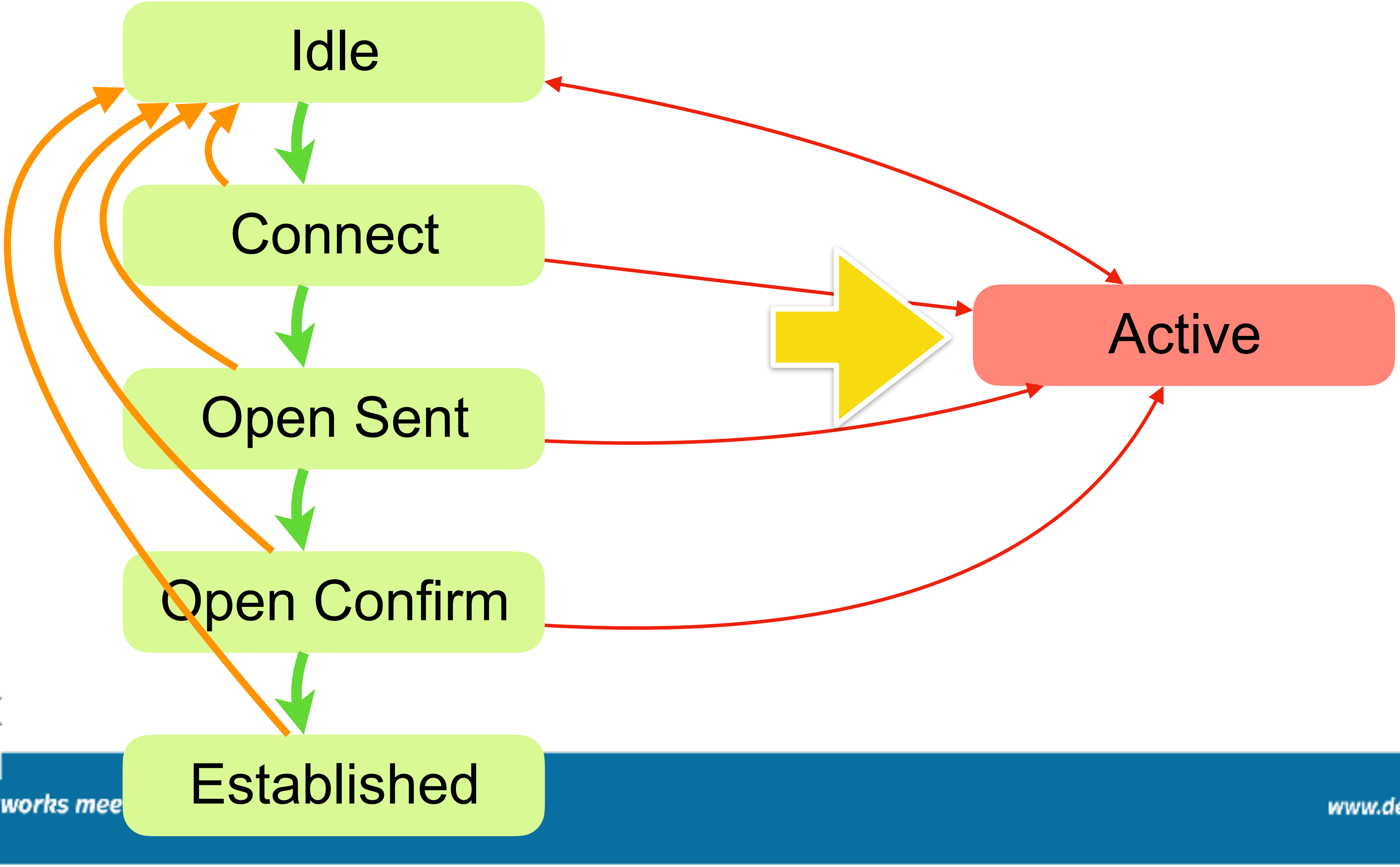
→Today: IS-IS is already set up in the lab

  →we only set up iBGP

BGP Session Setup

- →BGP uses **TCP for transport**
- →TCP already provides **reliable** transport
- →but a bit more is needed
  - →some information exchange at **setup**
  - →some mechanism for **keepalive**
- →a **state model** and timers

State model for a BGP session (incomplete)



Experiment: Setup iBGP

experiment 01d, 01e

iBGP - why fully meshed?

→BGP receives prefixes from external - eBGP

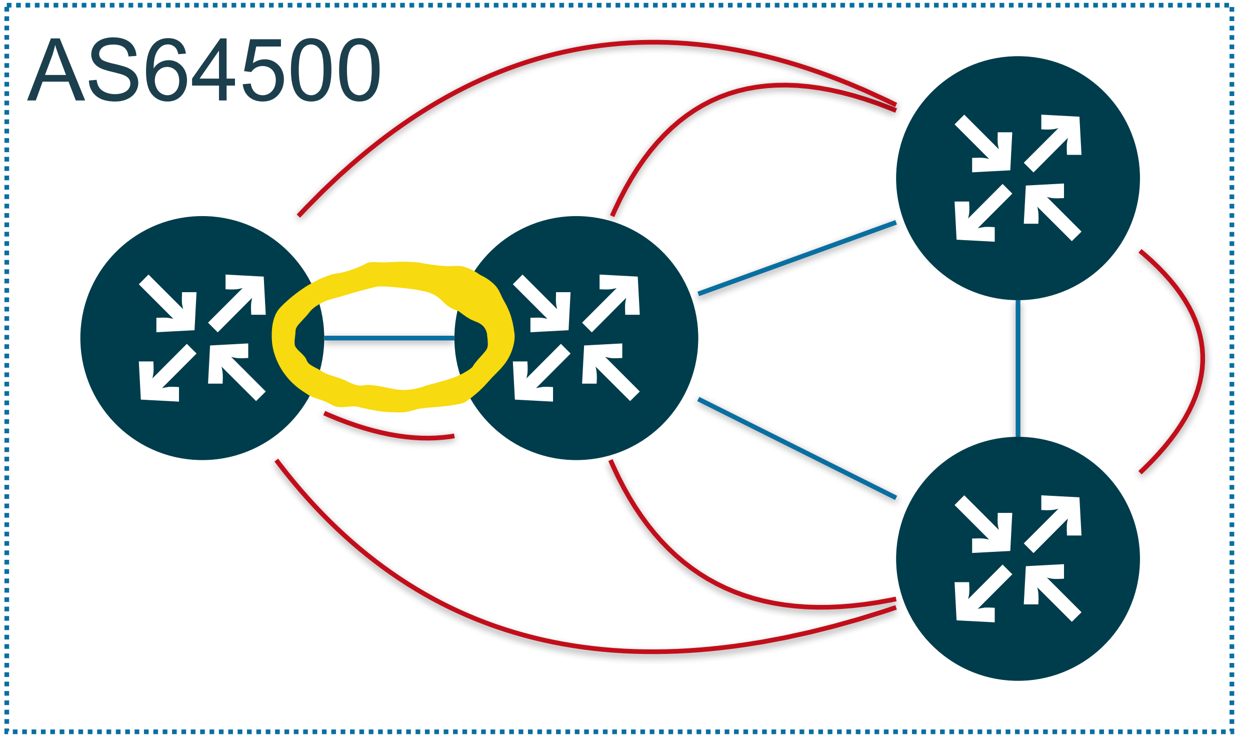
→BGP sends **all** prefixes to external (unless filtered)

→BGP sends prefixes **received from external** to internal

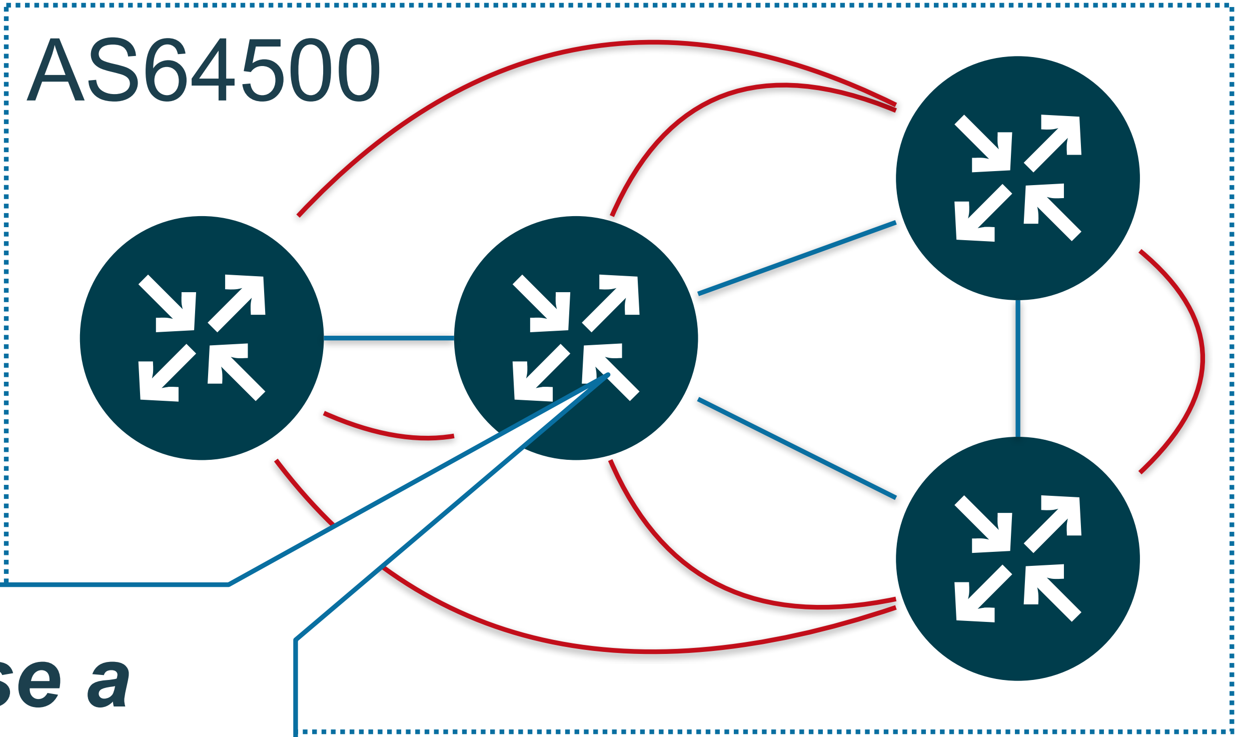
→BGP does **<u>not</u>** send prefixes received from internal to internal

→**unless...**

Example Network: Fully meshed iBGP?

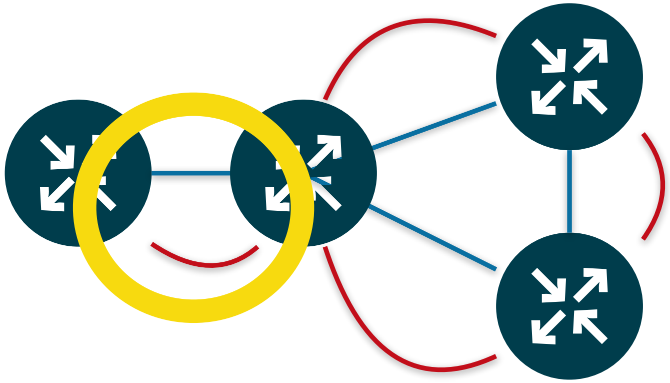


Example Network: ~~Fully meshed~~ iBGP?



Use a route reflector

iBGP: Route Reflector



→"Normal" prefix forwarding rule for iBGP

- →do **<u>not</u>** send out anything learned via iBGP

→route-reflector

- →defined in RFC4456
- →send out one best path of all prefixes to each route-reflector client

→how to configure

```
→neighbor x.x.x.x route-reflector-client
```

- →no special config on client side

Summary

- →BGP uses TCP
- →iBGP is BGP within an Autonomous System
- →BGP distributes prefixes
  - →from external to external
  - →from external to internal
  - →from internal to external
  - →**NOT** from internal to internal (exception: Route Reflector)

Thank you!

DE-CIX Management GmbH | Lindleystr. 12 | 60314 Frankfurt | Germany
Phone + 49 69 1730 902 0 | sales@de-cix.net | www.de-cix.net

Connect: http://go.de-cix.net/webinarlab

Connecting

- You only need a web browser
- The URL to connect to see above - you will be redirected to a router
- All routers run on the same server
- They run on different *ports* - port **9001** is router r01 and so on
- Connect to your router now

Entering commands

- Your router has two modes of operation
- **Terminal Mode** - this is the one you start with.
- The prompt in terminal mode is your router name followed by “#”
- Your router can *autocomplete* commands - just use the *tab* key (type twice for a list of possible completions).
- Or type “?” for a list of possible things you can enter

Lets try that now!

```
r01# sh?
```

Setup iBGP for IPv4

- We use *peer-groups* for configuration
- All *common* config statements go into the peer-group
- All individual statements to into the peer entry

```
router bgp 64500
 no bgp default ipv4-unicast

 neighbor internal peer-group
 neighbor internal remote-as 64500
 neighbor internal update-source dummy0
 address-family ipv4 unicast
   neighbor internal activate
 exit-address-family

 neighbor 172.16.1.yy peer-group internal
 ...
```

- Repeat the last line for all neighbors
- For *yy* use the router ids of the other routers (all except your own)
- We change the default so IPv4 is not automatically activated

Show commands:

- show bgp ipv4 summary
- show bgp ipv4 neighbors
- show bgp ipv4 neighbor 172.16.1.yy

Setup iBGP for IPv6

- We again use a *peer-group*
- You need separate peer-groups for IPv4 and IPv6
- All *common* config statements go into the peer-group
- All individual statements to into the peer entry

```
router bgp 64500
 no bgp default ipv4-unicast

 neighbor internal-v6 peer-group
 neighbor internal-v6 remote-as 64500
 neighbor internal-v6 update-source dummy0
 address-family ipv6 unicast
  neighbor internal-v6 activate
 exit-address-family

 neighbor 2001:db8:500::1:yy peer-group internal-v6
 ...
```

- See the *address-family* entry?
- This is about what is being *transported* via BGP
- We have to activate IPv6
- and deactivate (*no bgp...*) IPv4 (we simply change the default)

Show commands:

- show bgp ipv6 summary
- show bgp ipv6 neighbors
- show bgp ipv6 neighbor 2001:db8:500::1:yy

Setup iBGP

DE-CIX Academy

Version 1.0w

1 Purpose

iBGP is simply BGP within one Autonomous System. It is used to redistribute prefixes received from other providers and to redistribute your own or your customers prefixes.

2 Network Setup

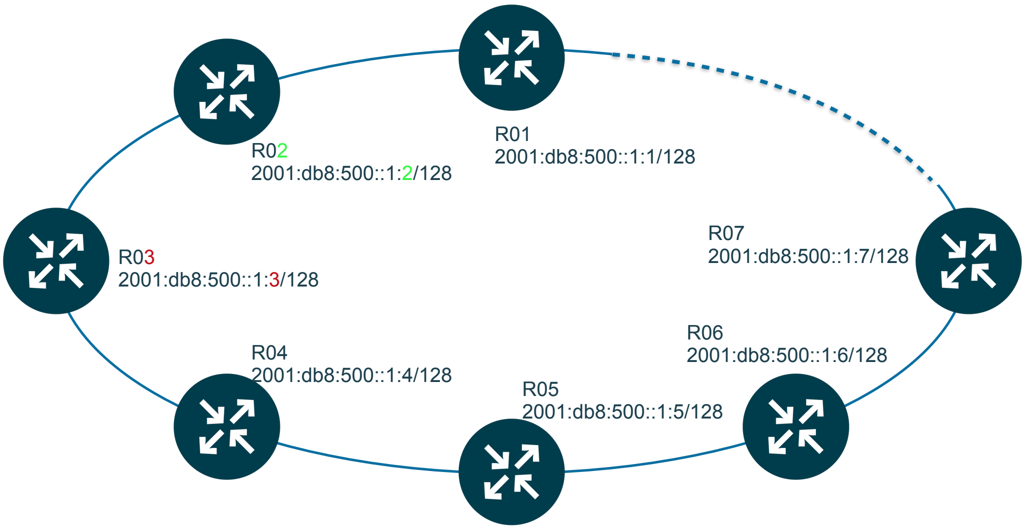


Figure 1: Network Setup

Figure 1 shows the network topology for this experiment. All devices are connected in a ring like structure, each device has two neighbors. Loopback addresses are distributed via an IGP (OSPF or IS-IS).

3 Setup iBGP

3.1 Tasks:

- Define iBGP peer groups for IPv4 and IPv6
- Configure all parameters needed in the peer group

- Set up iBGP sessions to all other routers on IPv4 and IPv6 using the peer groups
- Check if the sessions are up

3.2 To set up iBGP you need to:

- start BGP using your AS number
- define one iBGP peer group for IPv4 and one for IPv6
- configure all iBGP neighbors

Information you need:

- Your AS number: **64500**

The IP addresses of your iBGP neighbors you can find out by doing a *show ip route ospf* or *show ip route isis* (depending on the IGP in use) and look for /32 prefixes for IPv4. For IPv6 do a *show ipv6 route ospf6* or *show ipv6 route isis* and look for /128 prefixes.

3.3 To configure iBGP in config mode you need:

router bgp <asnumber> to start BGP with *<asnumber>* as your AS number

neighbor <name> peer-group to start configuring a peergroup named *<name>*

neighbor <name> remote-as <asnumber> to set a common remote AS number for all peer group members. If configuring iBGP *<asnumber>* is **your** AS number.

neighbor <name> update-source <interface> make BGP use the IP address of *<interface>* as source IP when setting up connections. Use your loopback interface name (*dummy0*) here.

neighbor <name> next-hop-self makes iBGP to set this routers IP (of *dummy0*) for next-hop address when distributing prefixes received via eBGP

neighbor <name> next-hop-self all makes iBGP to set this routers IP (of *dummy0*) for next-hop address when distributing prefixes received via eBGP *or iBGP*

neighbor <name> send-community both to forward BGP communities. We have not covered them yet, but you will need this later on.

neighbor <ip address> peer-group <name> sets up a BGP session to *<ip address>* as a member of peer-group *<name>*. That means all configs from *<name>* are copied.

address-family ipv6 switches into IPv6 context. You need to activate explicitly distributing IPv6 prefixes in this context.

address-family ipv4 switches into IPv4 context. You do not need to activate the distribution of IPv4 prefixes via IPv4 - this is activated by default. But you need to *de*activate the distribution of IPv6 addresses.

neighbor <ipv6 address> activate in address-family context activates distribution of IPv6 prefixes in this context.

neighbor <ipv4 address> activate in address-family context activates distribution of IPv4 prefixes. Not needed in IPv4 context.

no neighbor <ipv6 address> activate in address-family context prevents distribution of IPv6 prefixes.

no neighbor <ipv4 address> activate in address-family context prevents distribution of IPv4 prefixes.

It is recommended to distribute prefixes only in their own context. So you usually configure:

```
...
address-family ipv4 unicast
  neighbor <ipv4 address> activate
  ... (for all IPv4 neighbors)
  no neighbor <ipv6 address> activate
  ... (for all IPv6 neighbors)
exit-address-family
```

```
  address-family ipv6 unicast
    no neighbor <ipv4 address> activate
    ... (for all IPv4 neighbors)
    neighbor <ipv6 address> activate
    ... (for all IPv6 neighbors)
  exit-address-family
```

In FRRouting you can configure this also for a whole peer group and you can change the default so IPv4 is *not* activated automatically. See solution below.

3.4 Commands to check if iBGP is running:

show ip bgp summary shows you all configured neighbor and if sessions to them are up or not.

show ip bgp neighbors show you detailed information about all BGP neighbors (long!)

show ip bgp neigbors <ip address> same as above, but only for one neigbor. Preferred.

4 Solution

4.1 IPv4

```
  router bgp 64500
    no bgp default ipv4-unicast
    neighbor internal peer-group
    neighbor internal remote-as 64500
    neighbor internal update-source dummy0
    address-family ipv4 unicast
      neighbor internal activate
    exit-address-family
    neighbor 172.16.1.YY peer-group internal
    ...
```

Where *YY* are the IPv4 addresses of all other routers in the network (except your own).

4.2 IPv6

```
  router bgp 64500
    no bgp default ipv4-unicast
    neighbor internal-v6 peer-group
    neighbor internal-v6 remote-as 64500
    neighbor internal-v6 update-source dummy0
    address-family ipv6 unicast
      neighbor internal-v6 activate
    exit-address-family
    neighbor 2001:db8:500::1:Y peer-group internal-v6
    ...
```

Where *Y* are the IPv6 addresses of all other routers in the network (except your own).

5 Links mentioned in the webinar

5.1 RFCs

- RFC1930 - Autonomous Systems
- RFC4271 - BGP
- RFC4456 - Route Reflectors

5.2 Software

- DE-CIX Academy Lab
- FRRouting
- GNS3