Introduction to BGP

Workshop

We will start at 11:00

Wolfgang Tremmel
academy@de-cix.net

13
DENOG
GERMAN NETWORK OPERATORS GROUP

DE CIX

Today's Training

- →IP Prefixes and AS Numbers
- →BGP: Introduction
- →iBGP and eBGP
- →Becoming Multi-Homed
- →BGP Best Path Selection

Yes, there is a lunch break

Yes, there are coffee- and bio-breaks

About me

→Wolfgang Tremmel

→studied Informatik (Uni Karlsruhe)

  →Degree: Diploma (1994)

→Network Engineer at

  →Since 1996 Director NOC

  →Since 2000 Senior Network Planner DSL at

→2001 - 2005 Director Network Planning at VIA NET.WORKS

→2006 - 2016 Manager Customer Support at

→since 2016: Head of DE-CIX Academy

DE-CIX Academy

→"Learn from the experts"

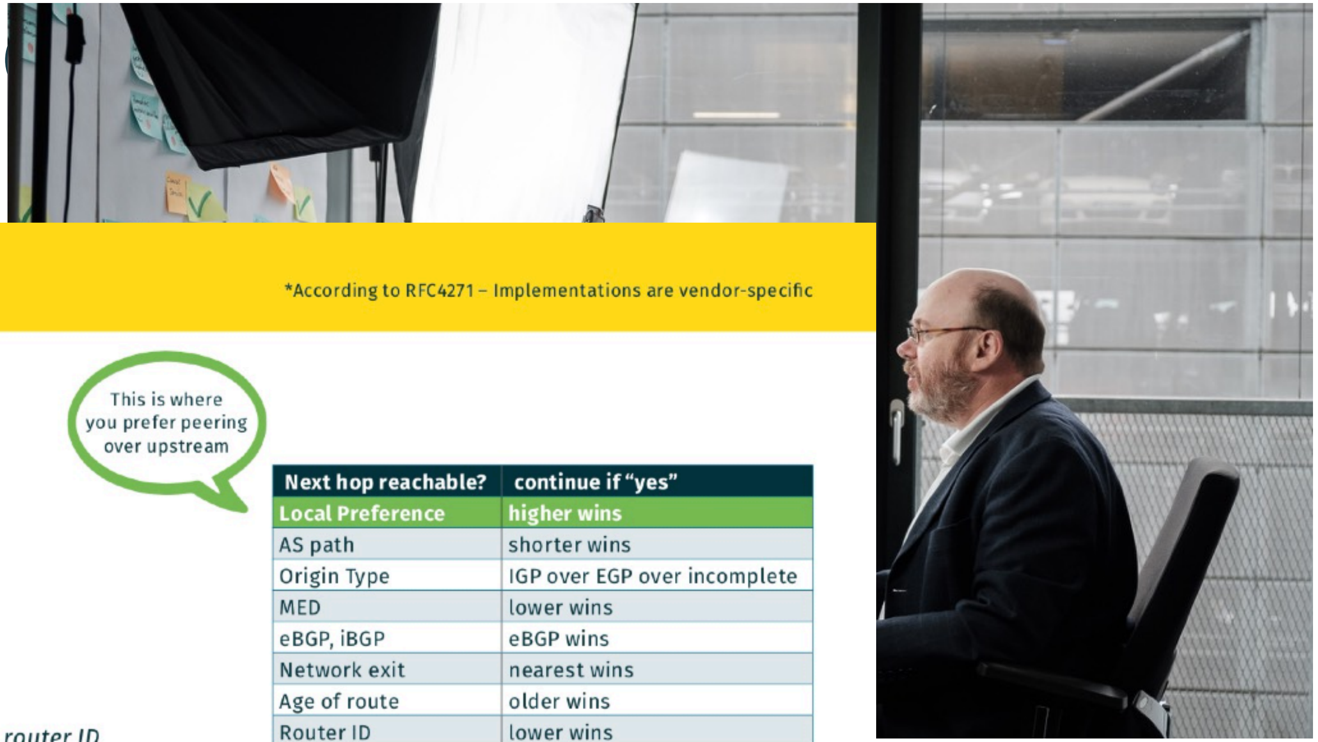
→Webinars about topics related to ISPs, routing, peering

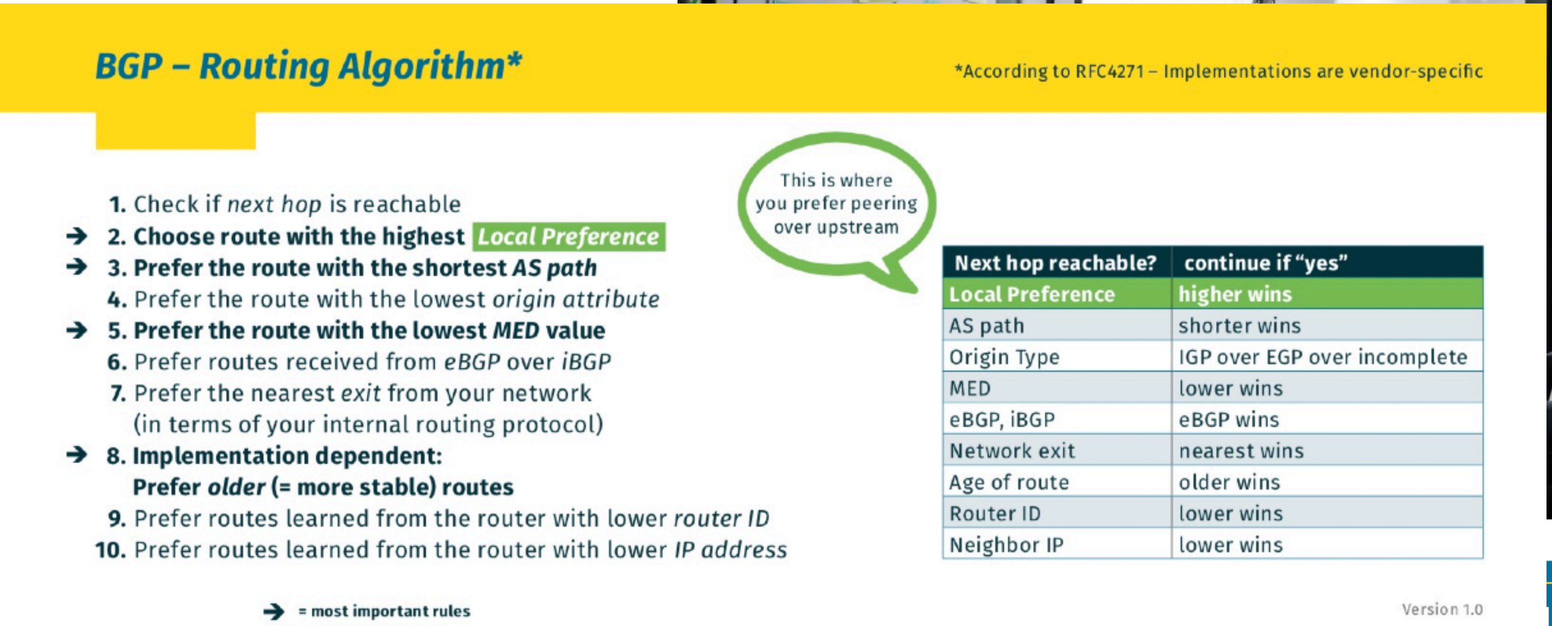
→Seminar(s) about BGP and

→Knowledge Cards

→Whi

→de-c




BGP – Routing Algorithm*

*According to RFC4271 – Implementations are vendor-specific

1. Check if *next hop* is reachable
→ **2. Choose route with the highest *Local Preference***
→ **3. Prefer the route with the shortest *AS path***
4. Prefer the route with the lowest *origin attribute*
→ **5. Prefer the route with the lowest *MED* value**
6. Prefer routes received from *eBGP* over *iBGP*
7. Prefer the nearest *exit* from your network (in terms of your internal routing protocol)
→ **8. Implementation dependent: Prefer *older* (= more stable) routes**
9. Prefer routes learned from the router with lower *router ID*
10. Prefer routes learned from the router with lower *IP address*

This is where you prefer peering over upstream

Next hop reachable?	continue if "yes"
Local Preference	higher wins
AS path	shorter wins
Origin Type	IGP over EGP over incomplete
MED	lower wins
eBGP, iBGP	eBGP wins
Network exit	nearest wins
Age of route	older wins
Router ID	lower wins
Neighbor IP	lower wins

→ = most important rules

Version 1.0

Introduce yourself

→Who are you?

→Who are you working for?

→Why are you here at DENOG?

→Why are you here at this workshop?

?

Prefixes and Netmasks

IPv4 and IPv6

Wolfgang Tremmel
wolfgang.tremmel@de-cix.net

Today's Training

→**IP Prefixes and AS Numbers**

→BGP: Introduction

→iBGP and eBGP

→Becoming Multi-Homed

→BGP Best Path Selection

IPv4 Addresses

10.3.8.17

IPv4 Addresses
IPv4 Prefixes

10.3.8.17

10.3.8.0/22

IPv4 Prefixes

10.3.8.0/22

1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16	17 18 19 20	21 22 23 24	25 25 27 28	29 30 31 32
0000	1010	0000	0011	0000	1000	0000	0000

- → IPv4 (and IPv6) addresses have a network and a host part
- → A **prefix** is just the network part
- → Important:
  - **The boundary between network and host can be anywhere!**

Characteristics of Prefixes: IPv4

10.3.8.0/22

1 2 3 4	5 6 7 8	9 10 11 12	3 14 15 16	17 18 19 20	21 22 23 24	25 25 27 28	29 30 31 32
0000	1010	000	0011	0000	1000	0000	0000

<u>Notation:</u>

- 4 Numbers 0-255
- Separated by "."
- a "/", followed by 0-32

Characteristics of Prefixes: IPv4

10.3.8.0/22

Prefix-Length: 0-32

1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16	17 18 19 20	21 22 23 24	25 25 27 28	29 30 31 32
0000	1010	0000	0011	0000	1000	0000	0000

Characteristics of Prefixes: IPv4

10.3.8.0/22

1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16	17 18 19 20	21 22 23 24	25 25 27 28	29 30 31 32
0000	1010	0000	0011	0000	1000	0000	0000

32 Bits long

DE CIX

Characteristics of Prefixes: IPv4

10.3.8.0/22

1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16	17 18 19 20	21 22 23 24	25 25 27 28	29 30 31 32
0000	1010	0000	0011	0000	1000	0000	0000

Host-part all zero

Characteristics of Prefixes: IPv4

10.3.8.0/22

Prefix-Length: 0-32

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 25 27 28 29 30 31 32

0000 1010 0000 0011 0000 1000 0000 0000

Notation:

- 4 Numbers 0-255
- Separated by "."
- a "/", followed by

Host-part all zero

32 Bits long

IPv6 Addresses

IPv6 Prefixes

2003:de:274f:400:226:b0ff:fed8:3d8a

2003:de:274f:400:/64

Characteristics of Prefixes: IPv6

2003:de:274f:400::/64

0 01 02 03 04 05 06 07 08 09 0a 0b 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f

<u>Notation:</u>

- 4 digit hex numbers (0-9,a-f)
- Separated by ":"
- 8 Numbers max.
- "::" = fill up with zeros

Characteristics of Prefixes: IPv6

Prefix-Length: 0-128

2003:de:274f:400::/64

0 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f

Characteristics of Prefixes: IPv6

2003:de:274f:400::/64

0 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f

128 Bits long

Characteristics of Prefixes: IPv6

2003:de:274f:400::/64

0 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f

Host-part all zero

Characteristics of Prefixes: IPv6

Prefix-Length: 0-128

2003:de:274f:400::/64

0 01 02 03 04 05 06 07 08 09 0a 0b 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f

Notation:
- 4 digit hex numbers (0-9,a-f)
- Separated by ":"
- "::" = fill up with zeros

Host-part all zero

128 Bits long

IP Adresses and Prefixes

	IPv4	IPv6
Length	32 Bit	128 Bit
	0-32 Prefix Length	0-128 Prefix Length
Notation	4 Numbers, 0-255	8 Numbers, 0-fffff
Separator	.	:
Prefix: Host part (Bits)	all zero	
Address: Host part (Bits)	**not** all zero / **not** all one	
Example (Prefix)	198.51.100.0/24	2001:db8:4f30::/48

What is an Autonomous System?

And why do I need one?

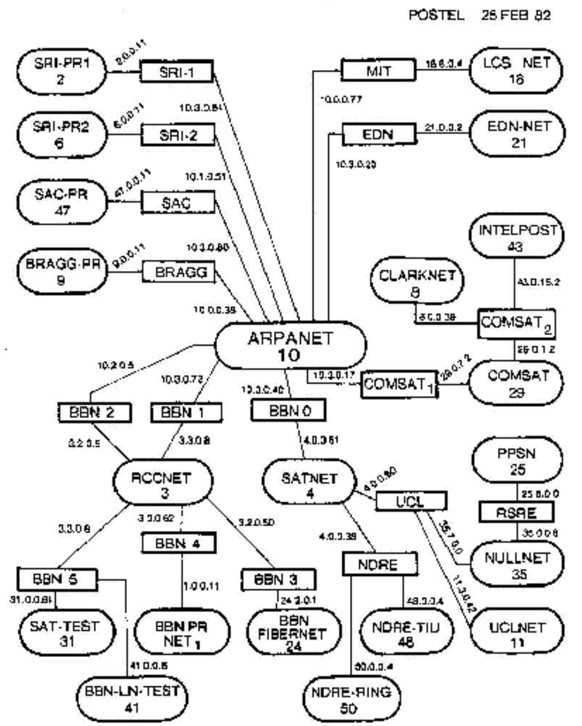
Wolfgang Tremmel
academy@de-cix.net

A brief history of the Internet

According to the Internet Hall of Fame

- → 1982 – Arpanet (successor of Internet)
- → 1982: RFC827 defines Exterior Gateway Protocol:

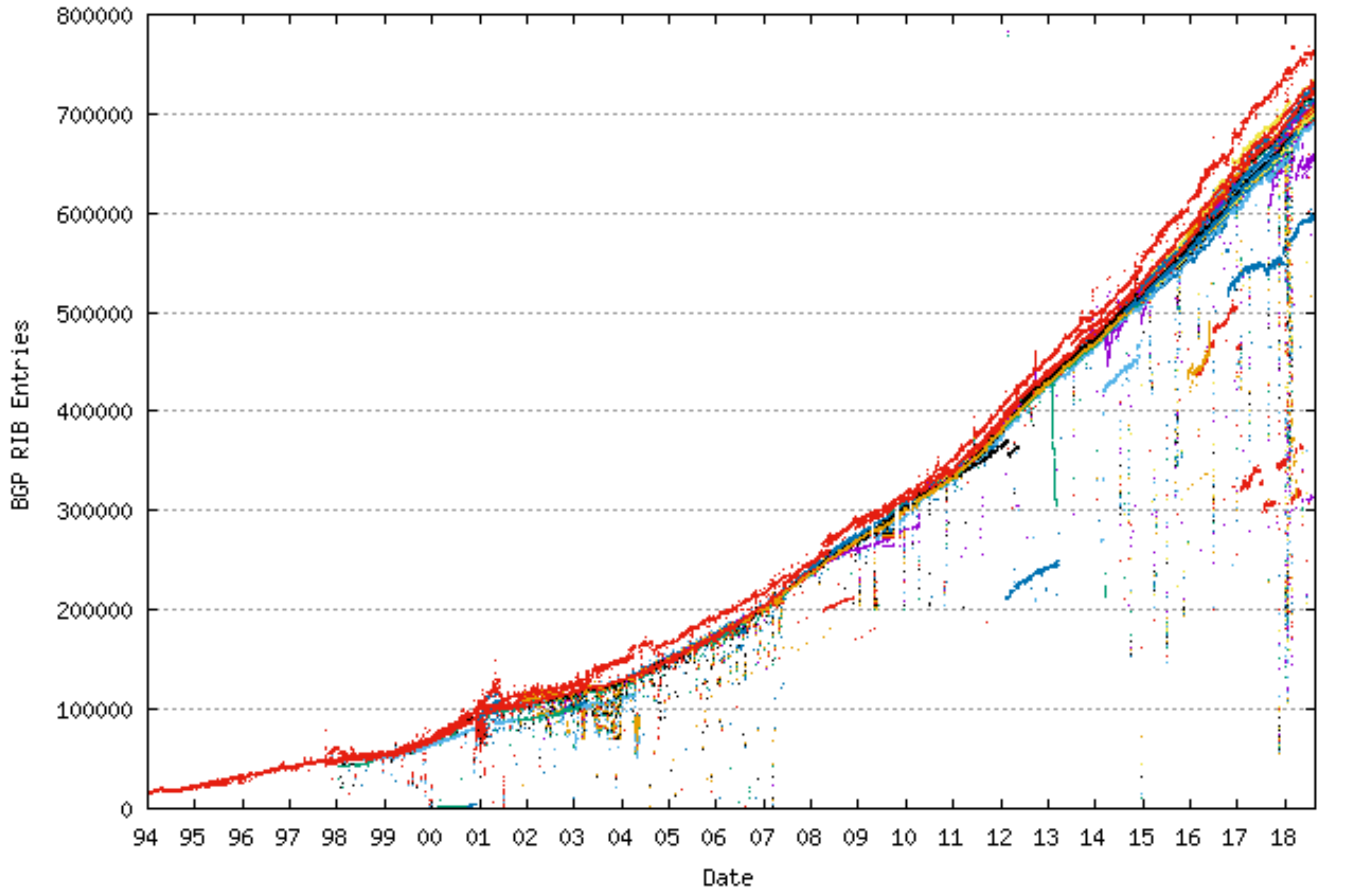
"Autonomous systems will be assigned 16-bit identification numbers (in much the same ways as network and protocol numbers are now assigned)"



DE CIX

Some years later...

- October 2019: There are 67150 active ASs (source: http://bgp.he.net/report/prefixes#_networks)
- In 2001, planning
- This was finalize 6793)
- Today, 4-byte AS
  - They are supp
  - You can no lo
  - There is also

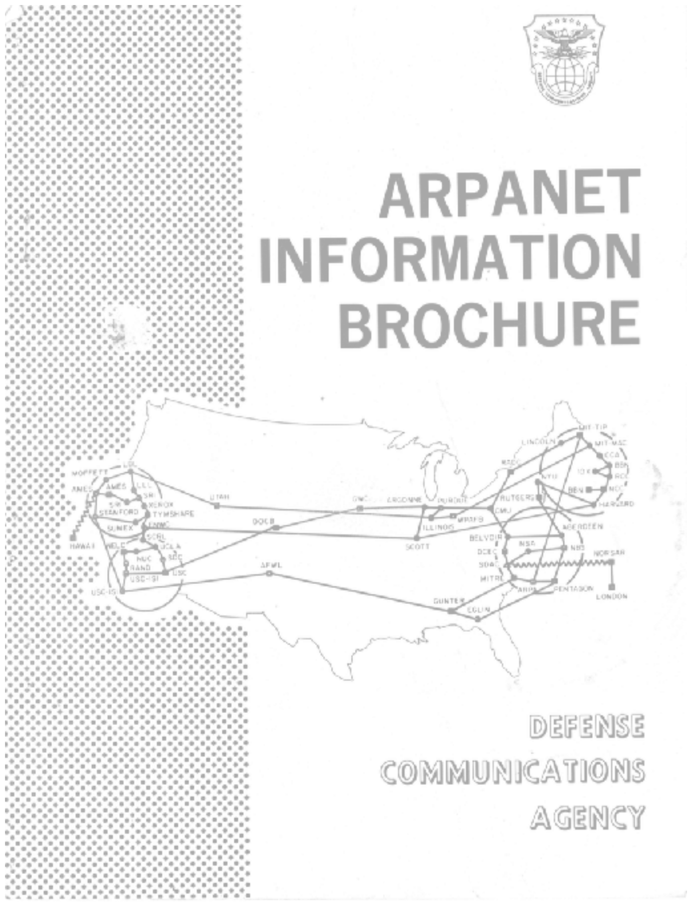


But what is an Autonomous System?

```
The classic definition of an Autonomous System
is a set of routers under a single technical
administration, using an interior gateway
protocol and common metrics to route packets
within the AS, and using an exterior gateway
protocol to route packets to other ASes.
```

→1996 – Defined in RFC1930 (earlier definitions exist)

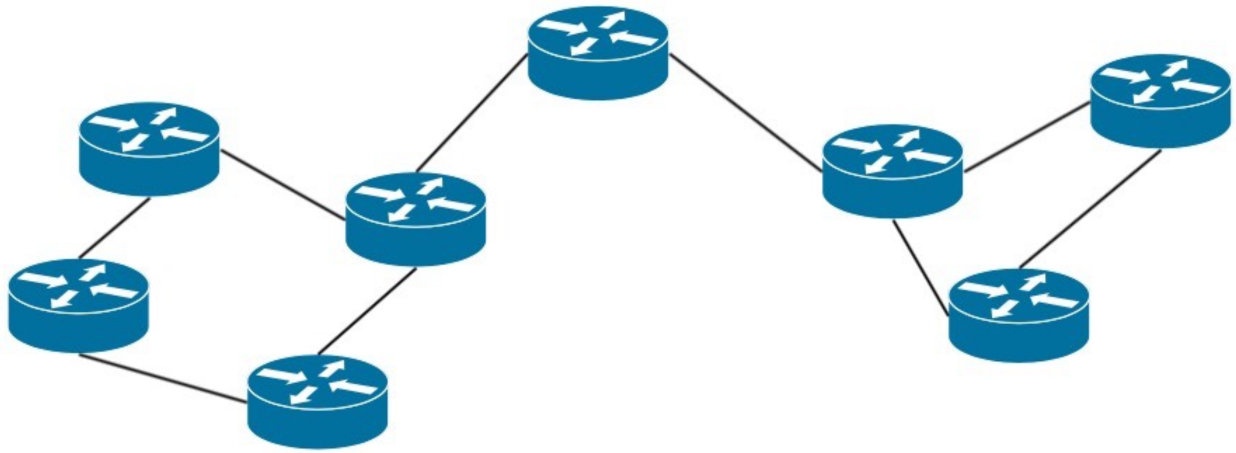
→What does this mean?



But what is an Autonomous System?

"An AS is a **connected** group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy."

→"connected": An autonomous system is continuous.
All entities within it are connected somehow with each other.



But what is an Autonomous System?

"An AS is a connected **group of one or more IP prefixes** run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy."

→"group of IP prefixes": This is about IP prefixes, not about devices. Routers are not even mentioned.

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 212.114.64.0/19  80.81.192.40            50    100          8859 i
*> 194.77.145.0/24  80.81.192.40            50    100          8859 i
*> 194.45.27.0/24   80.81.192.40            50    100          8859 i
*> 193.17.21.0/24   80.81.192.40            50    100          8859 i
*> 213.241.128.0/18 80.81.192.40            50    100          8859 i
```

But what is an Autonomous System?

"An AS is a connected group of one or more IP prefixes **run by one or more network operators** which has a SINGLE and CLEARLY DEFINED routing policy."

→"run by one or more network operators": An AS does not have to be run by only one operator, if all other conditions are matched.

```
aut-num:        AS6695
as-name:        DECIX-AS
descr:          DE-CIX Management GmbH
descr:          DE-CIX, the German Internet Exchange
descr:          DE
org:            ORG-DtGI1-RIPE
status:         ASSIGNED
mnt-by:         RIPE-NCC-END-MNT
admin-c:        DXSU6695-RIPE
tech-c:         DXSU6695-RIPE
tech-c:         BH6695-RIPE
mnt-by:         DECIX-MNT
mnt-lower:      DECIX-MNT
```

But what is an Autonomous System?

"An AS is a connected group of one or more IP prefixes run by one or more network operators which **has a SINGLE and CLEARLY DEFINED routing policy**."

→"has a SINGLE and CLEARLY DEFINED routing policy": The most important part.

→"routing policy": This is how routing decisions are made.

→An AS has only **one** routing policy.

→This policy is not defined for each single prefix, but for groups of prefixes.

→This group is called Autonomous System, ASs (RFC1930)

So this is an Autonomous System!

"An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy."

- →**So now you know:**
  - →You do not need a router
  - →However, you need prefixes to be routed
- →**Most commonly:**
  - →you do have a router
  - →… or more than one
  - →and it "belongs" to an AS by running BGP

What is an Autonomous System good for?

	If you have an AS	Without an AS
Redundancy	You can have multiple upstream ISPs and Peering	You only can have one upstream ISP
Control	You have full control over your outgoing traffic	Your upstream ISP controls your traffic
Cost	You can optimize your traffic for cost	You just pay your upstream ISP
Peering	You can setup your own peering policy and have full control	Your upstream ISP makes all decisions

Sounds good, I want an AS, where can I get one?

→AS numbers are globally unique

→So some sort of authority must exist for handing them out

→This authority is IANA - the Internet Assigned Numbers Authority

→But no, you cannot go to IANA and just ask for an AS – they delegated the task

→to five Regional Internet Registries (RIRs)

→have a look at the map to see who is responsible for your region

Regional Internet Registries (RIRs)

- →Talking about everything what RIRs do would be beyond the scope of this training
- →So, let's focus on AS numbers
- →And for now, let's focus on Europe
- →The RIR responsible for Europe, Russia and the Middle East is the RIPE NCC
- →RIPE means Réseaux IP Européens – the founders wanted a French name
- →NCC means Network Coordination Center
- →RIPE is not the same as RIPE NCC, see the website for the difference.
- →Back to how to get an AS number …

RIPE NCC
RIPE NETWORK COORDINATION CENTRE

Getting an AS number from RIPE NCC: The easy way

→Just become a customer

→You have to be a legal entity

→Fill out the forms

→Pay the sign-up fee (and annual fee)

→Request your AS number

→You have to be/want to be multi-homed (peering counts!)

→RIPE Academy offers lots of online / offline trainings to help you get started.

Getting an AS number without becoming a RIPE NCC member

→You can also get an AS from someone who already is a RIPE NCC customer

→This is called a "sponsoring LIR"

→Basically they request the AS from RIPE NCC for you

→… and may charge you for this service

Now I have an AS – how can I route my IP prefix?

- →Hmm, this depends where you have your IP space from
- →In general, IPv4 prefixes of /24 or larger are routable via BGP
- →In IPv6 you can route /48 or larger
- →If you have just become a new RIPE NCC member, you can also request IP space
  - →~~As there is not much IPv4 left, you get a /22 once (and not more)~~
  - →**IPv4 is out! No more IPv4 addresses (except by transfers)**
    - →Yes - new RIPE NCC members can still request a /24 via the waiting list
  - →But plenty of IPv6 available...
- →To check whether your current space is routable from your new AS, the best way is to check with whom you got that IP space from

AS Numbers

AS Numbers

→AS Numbers are...

  →originally 16 bit: 0 - 65535

→This was not enough

  →so 32-bit AS numbers were introduced: 65536 - 4294967295

→AS numbers (used on the Internet) are **globally unique**

→So if you need one for the Internet, get a **unique** one

→If you want to use AS numbers in a lab or disconnected network, use a **private AS number**

Private AS numbers

- →You all know private IPv4 networks like 192.168.0.0/16 or 10.0.0.0/8
- →Like this, there are private AS numbers (see [RFC6996](RFC6996))
  - →for 16-bit ASes this is 64512 - 65534
  - →for 32-bit ASes this is 4200000000 - 4294967294
- →Also, there are AS numbers set aside for documentation purposes:
  - →16-bit documentation AS numbers: 64496-64511
  - →32-bit documentation AS numbers: 65536-65551
- →Feel free to use them, just **never** announce them on the Internet

Open Questions?

BGP - an introduction

BGP for networks who peer: Part 1

Wolfgang Tremmel
wolfgang.tremmel@de-cix.net

Today's Training

- →IP Prefixes and AS Numbers
- →**BGP: Introduction**
- →iBGP and eBGP
- →Becoming Multi-Homed
- →BGP Best Path Selection

Today you will learn...

- → ... how to build and run a global network
- → ... how to operate routers with upstreams and peerings
- → ... how to reduce cost, increase performance and resilience

I am joking!

*But you will learn about **BGP**, the foundation of Internet routing*

BGP

BGP

Bridg

No!

Protocol

BGP

Border

Gateway

Protocol

BGP

→ P - a **PROTOCOL**

- spoken between Internet routers

→ B - spoken on the **BORDER** between two providers

→ G - on the **GATEWAYs** - the routers connecting two providers

DE CIX

Where networks meet

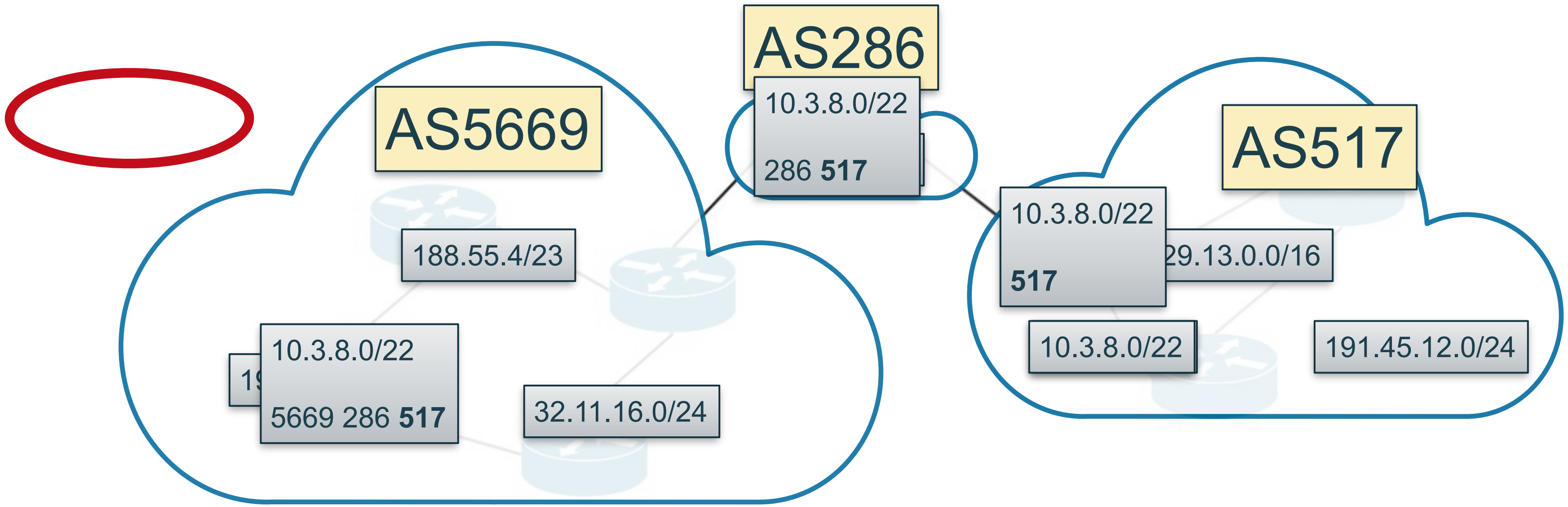
BGP Key Concepts

→IPv4 and IPv6 prefixes

→Autonomous Systems (AS)

→The Autonomous System Path

BGP - Key Concepts: The AS Path



A real live example

```
asd2-rs-02>show bgp ipv4 unicast 129.13.0.0
Load for five secs: 1%/0%; one minute: 4%; five minutes: 5%
Time source is NTP, 09:14:07.268 UTC Thu Aug 17 2017
BGP routing table entry for 129.13.0.0/16, version 2944571
Paths: (13 available, best #10, table default)
....
  125 286 517
    134.222.85.126 from 134.222.85.126 (134.222.85.126)
      Origin IGP, metric 0, localpref 80, valid, internal
      Community: 286:18 286:19 286:28 286:29 286:49 286:800 286:888
```

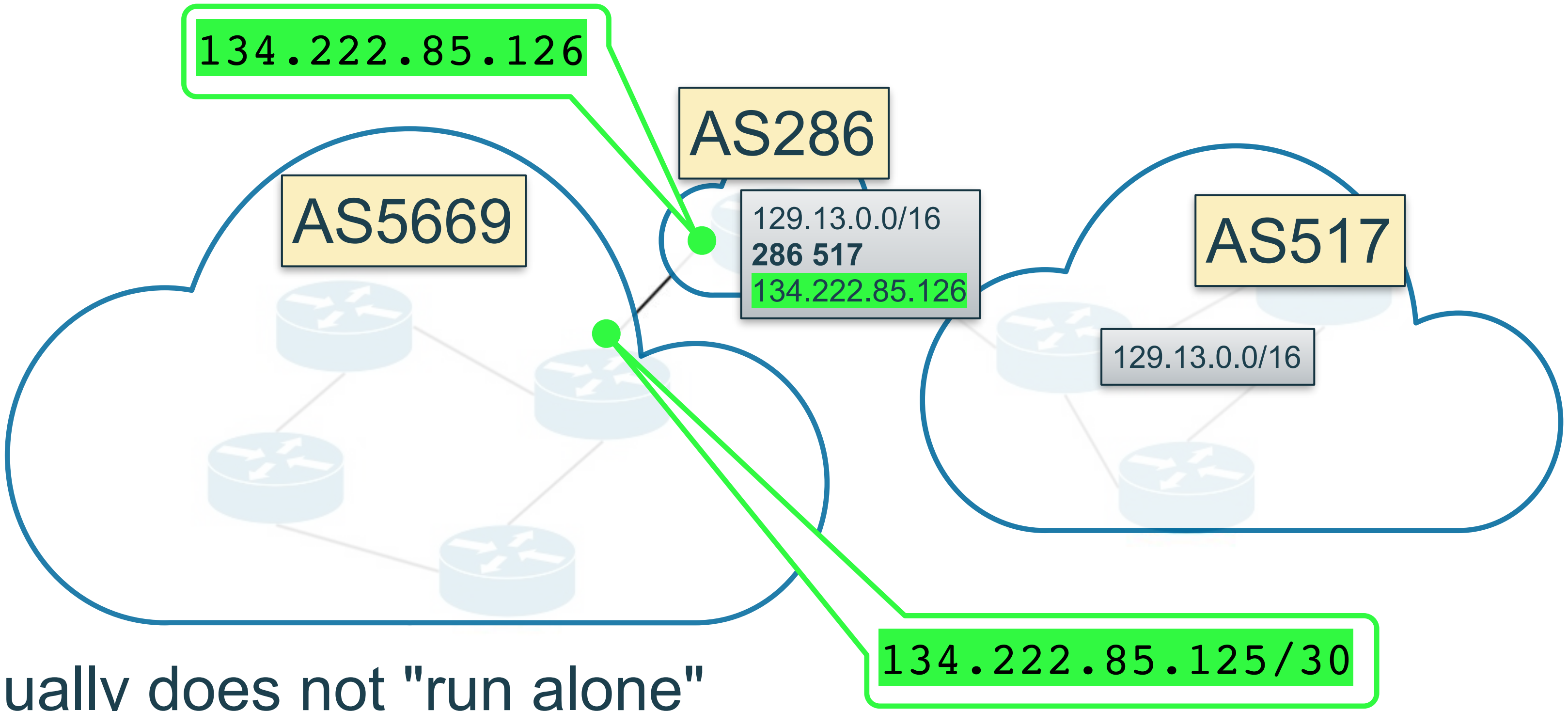
Prefix

AS-Path

Next Hop IP

Originator AS

BGP - Also important : Next Hop Address

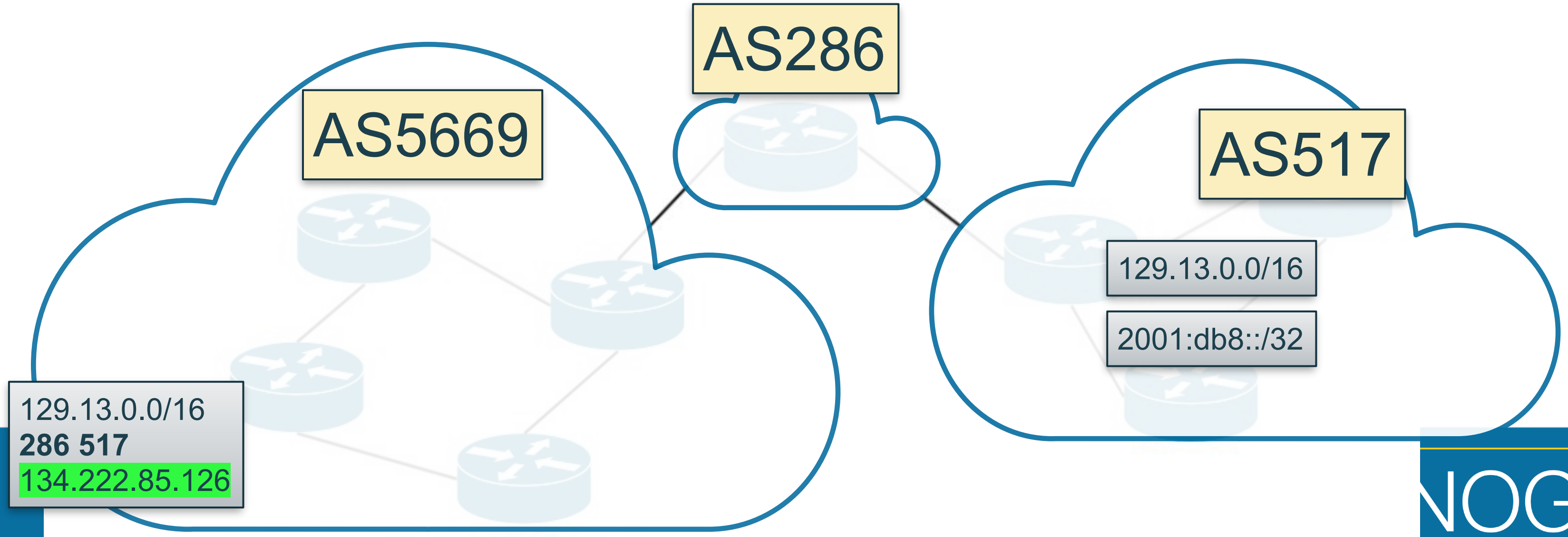


BGP usually does not "run alone"
Another routing protocol is needed to distribute interface addresses

BGP - Key Concepts: Summary

- Prefixes
- AS Numbers
- AS Path
- Next Hop

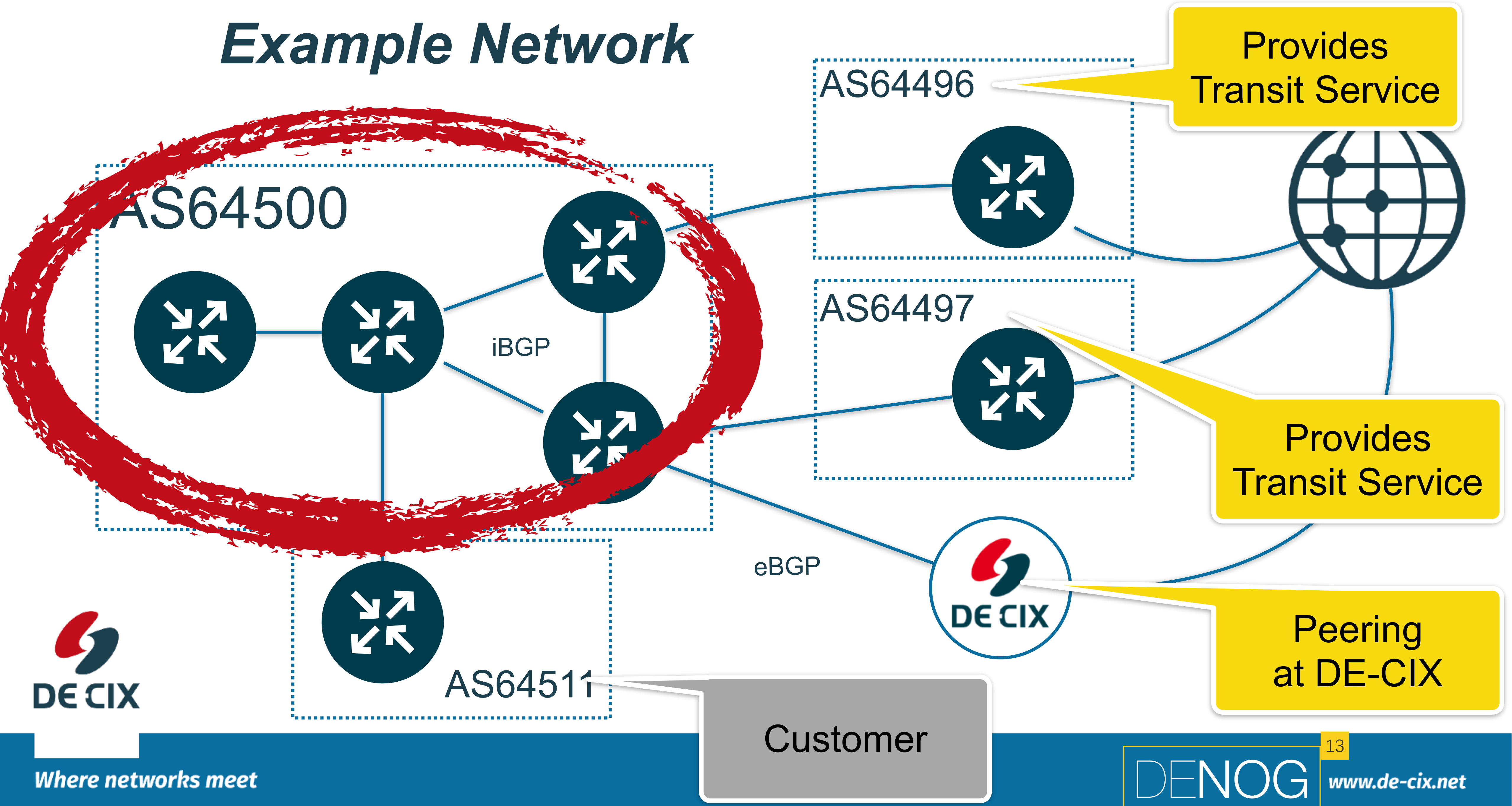
Originator AS



BGP: Example

- → BGP speaking routers do not "find themselves"
  - Everything needs to be configured
- → If you want to try yourself:
  - Install GNS3: https://gns3.com
  - Add a few routers (you need router software for this)
  - Start configuring

Example Network



BGP - Characteristics

→Routers setup **BGP sessions** between each other

→BGP uses **TCP** as transport protocol

→BGP works **incremental**

  →**after setup**, each router tells the other **all prefixes** it wants to announce

  →then **only updates** are sent

    →**withdraws**, if a prefix goes away

    →**adds**, if a prefix is added

BGP - not re-inventing the wheel

→BGP uses **TCP** for transport

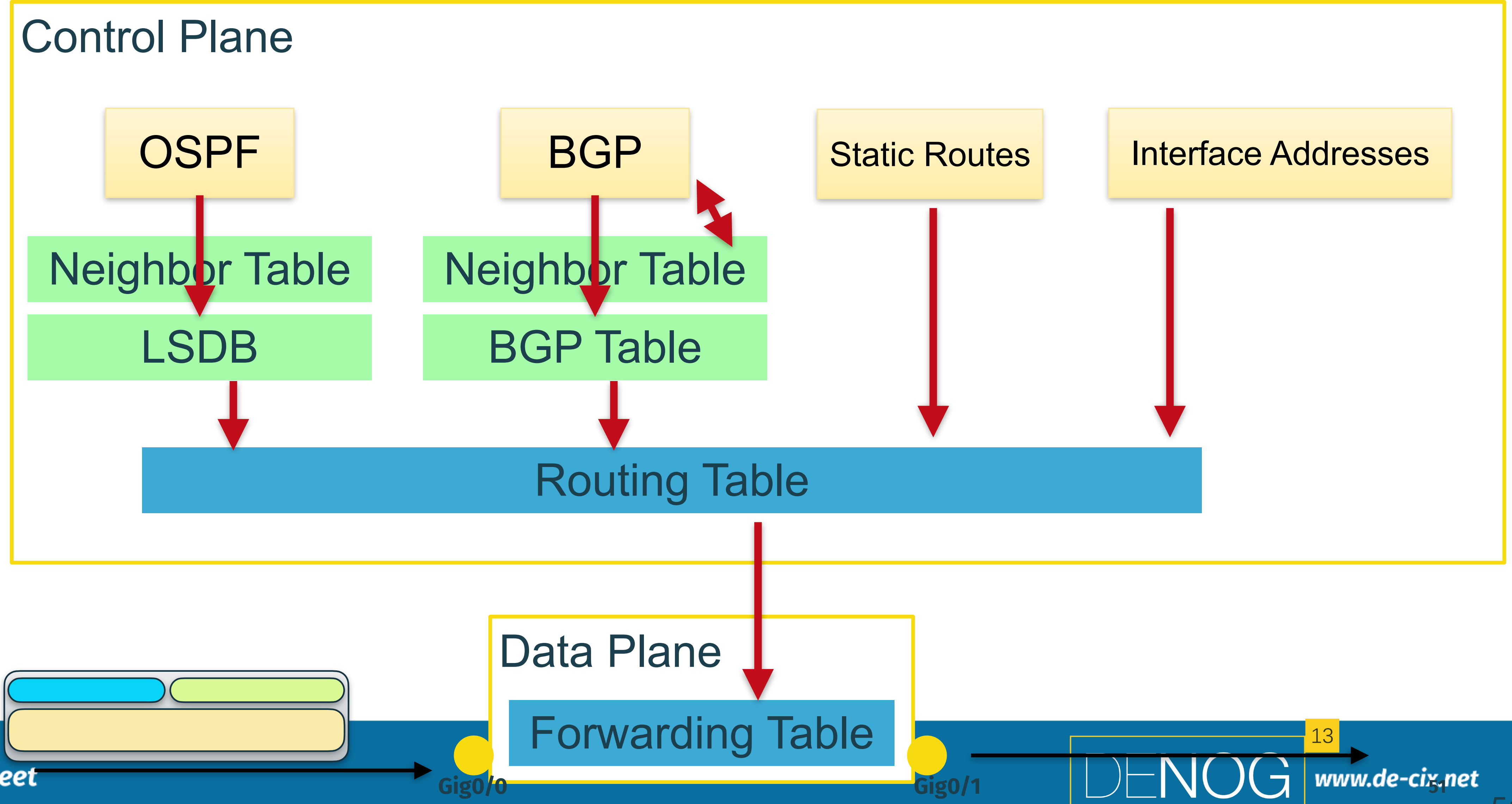
→so no need to re-implement features TCP already provides, like

- →reliable transport
- →flow control
- →framing

→as long as the TCP session is up, BGP assumes its neighbors are still there

- →and have all the information sent to them

How a router works



Open Questions?

BGP for networks who peer

00 - Connecting to the experiments

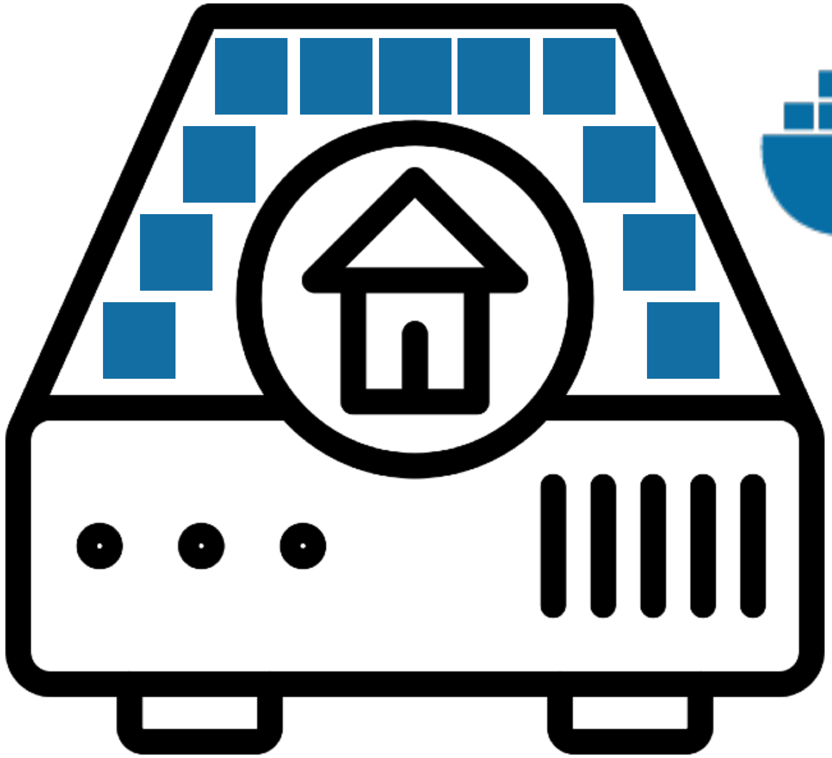
Wolfgang Tremmel

academy@de-cix.net

Network setup: Using Docker

https://bgplab.as196610.net:70xx

Network setup: Using Docker



docker

Network setup: Using Docker

Docker Container

- Alpine Linux
- FRRouting Software
- Supervisord
- TTYd

Network setup: FRRouting

→Open Source routing daemon

→based on Quagga

→Actively developed

→"Cisco-like" configuration syntax

→Not only BGP, but a lot of other protocols as well

→See frrouting.org

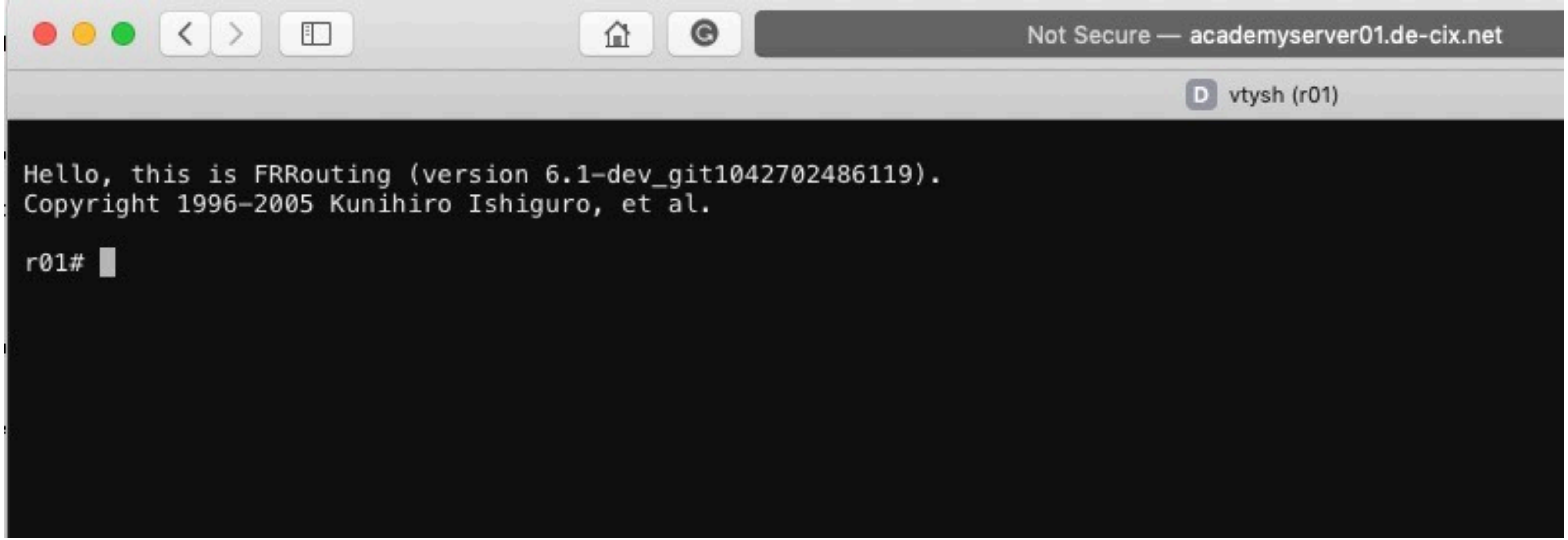
Connect now

→Your router:

- →using a Browser: https://bgplab.as196610.net:70xx

- →**XX** - see your confirmation email

- →xx is different for each of you


Not Secure — academyserver01.de-cix.net
vtysh (r01)
Hello, this is FRRouting (version 6.1-dev_git1042702486119).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
r01#

Experiment: Connecting to your router

experiment 00

BGP for networks who peer

01 - IGP and iBGP

Wolfgang Tremmel
academy@de-cix.net

Today's Training

→IP Prefixes and AS Numbers

→**BGP: Introduction**

→iBGP and eBGP

→Becoming Multi-Homed

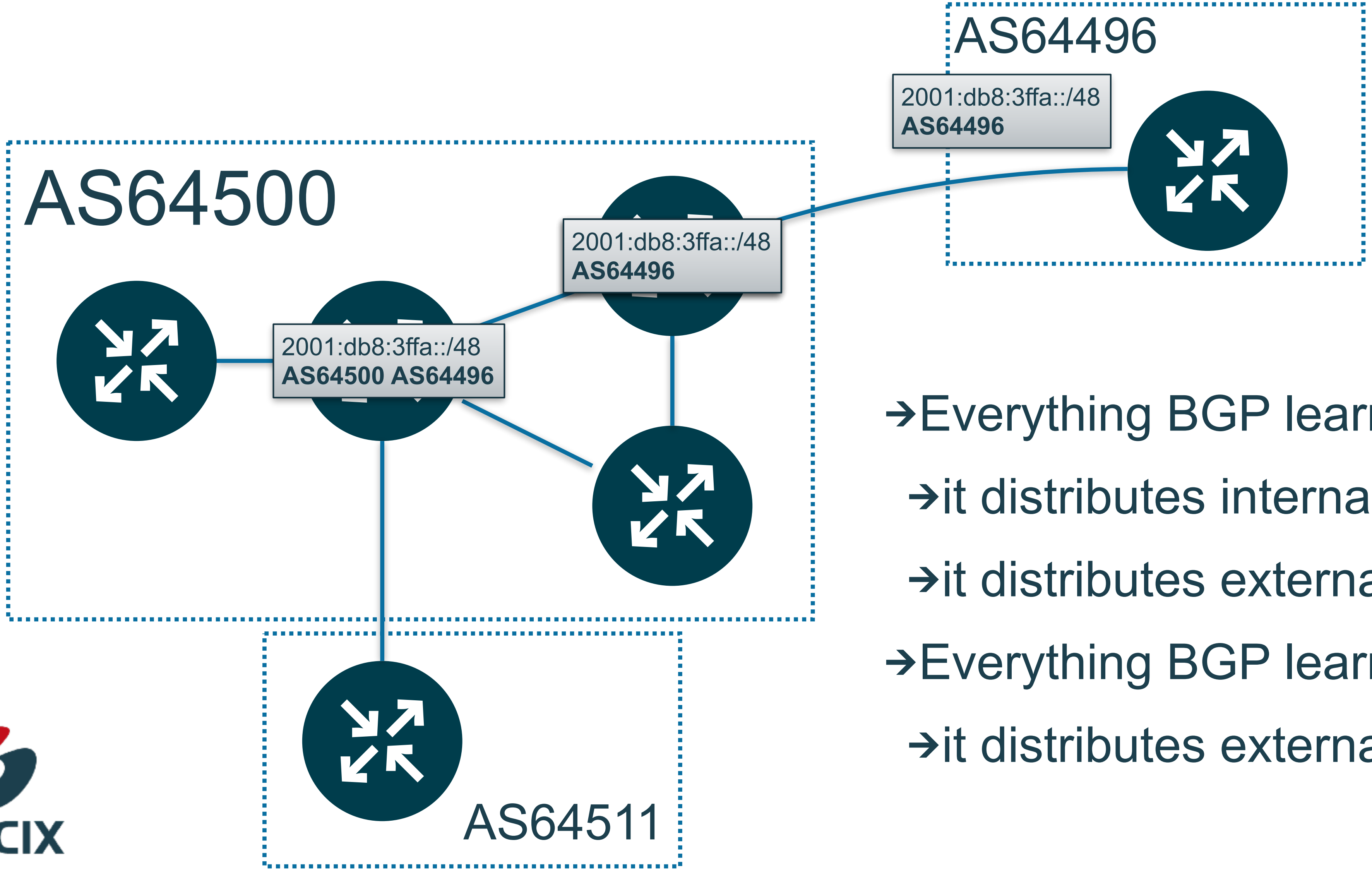
→BGP Best Path Selection

BGP - not re-inventing the wheel

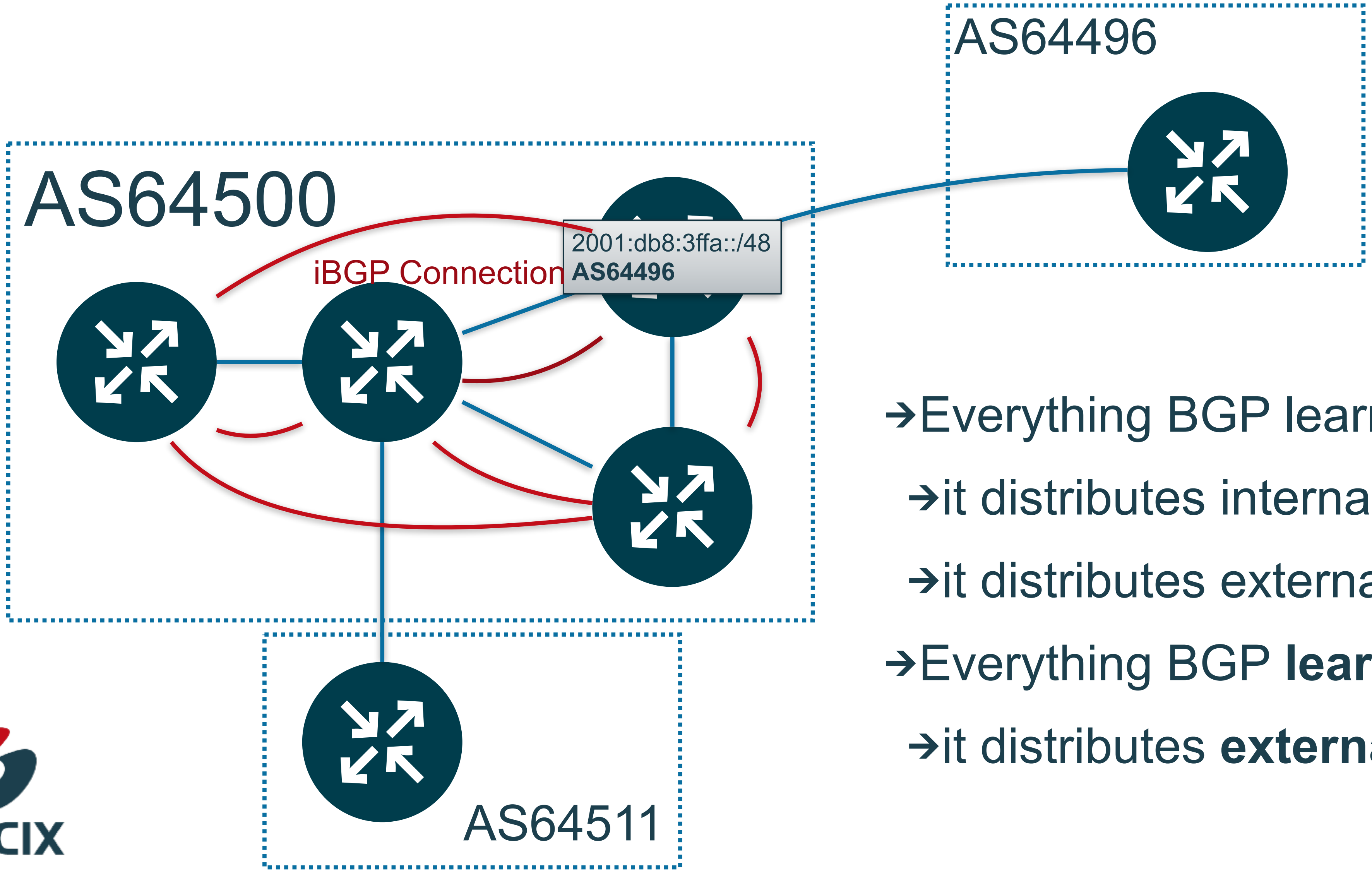
- →BGP uses TCP for transport
- →so no need to re-implement features TCP already provides, like
  - →reliable transport
  - →flow control
  - →framing
- →as long as the TCP session is up, BGP assumes its neighbors are still there
- →and have all the information sent to them

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

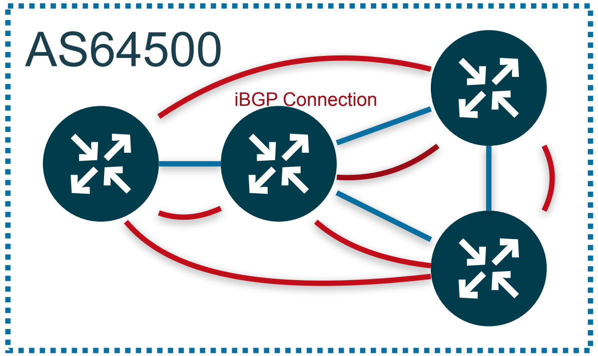
→it distributes internal

→it distributes external

→Everything BGP **learns from internal**

→it distributes **external**

Which routers do need to speak iBGP?



- →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)

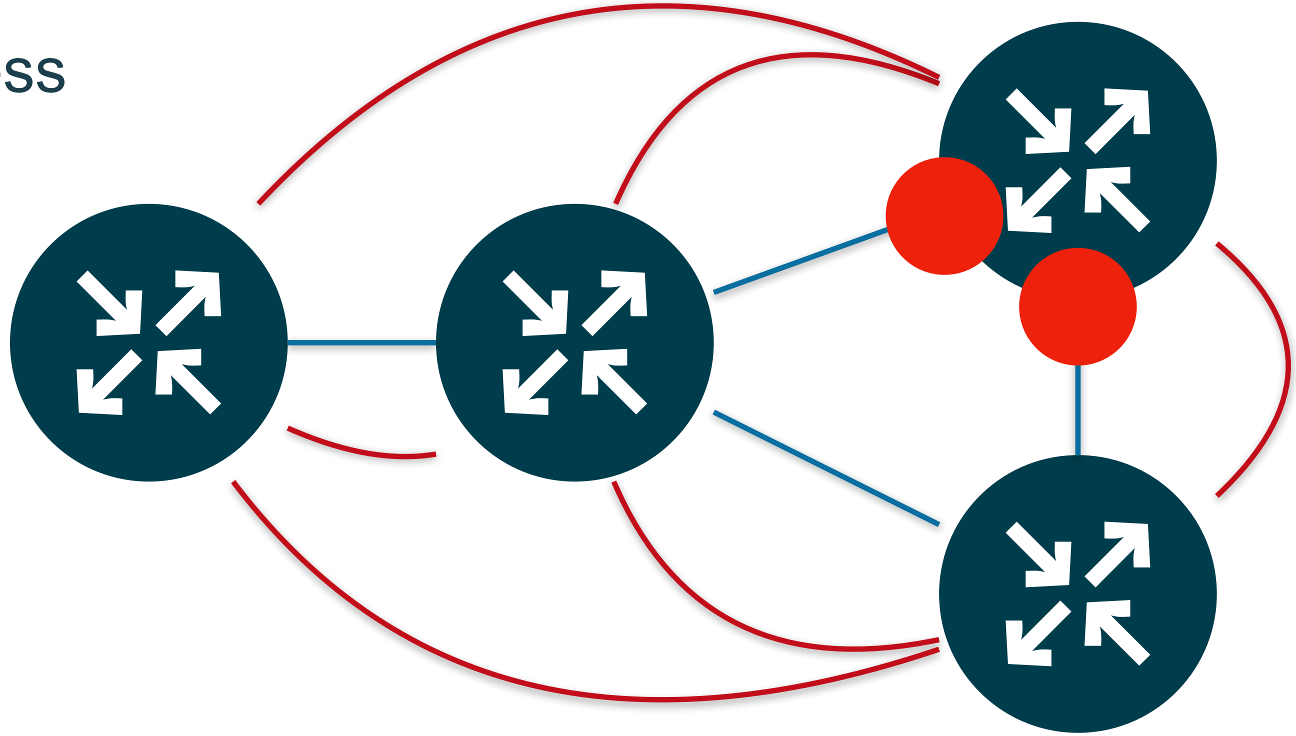
→a **source IP** address

→a **destination IP** address

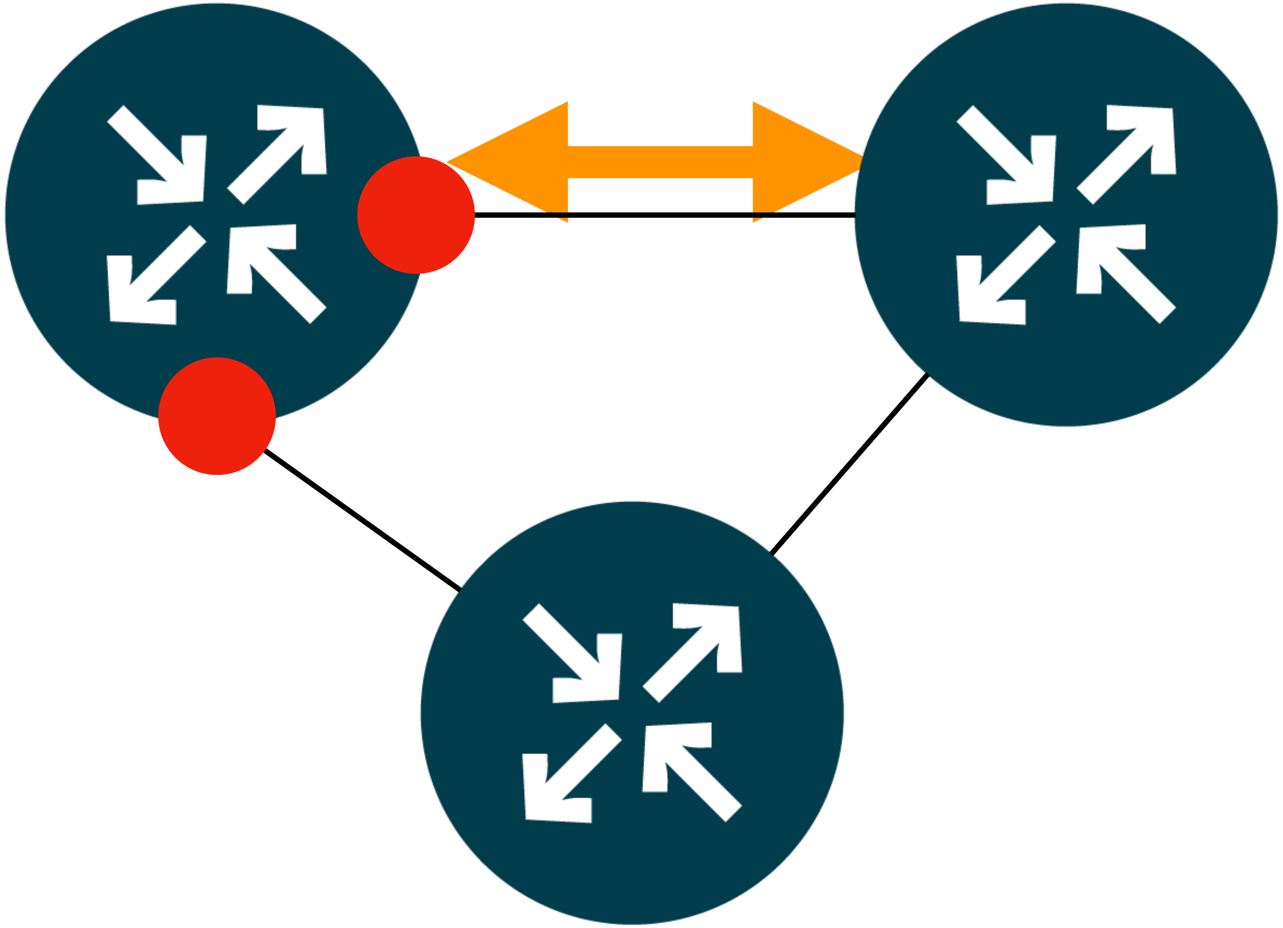
→**Source IP**: Which one?

→ Default:

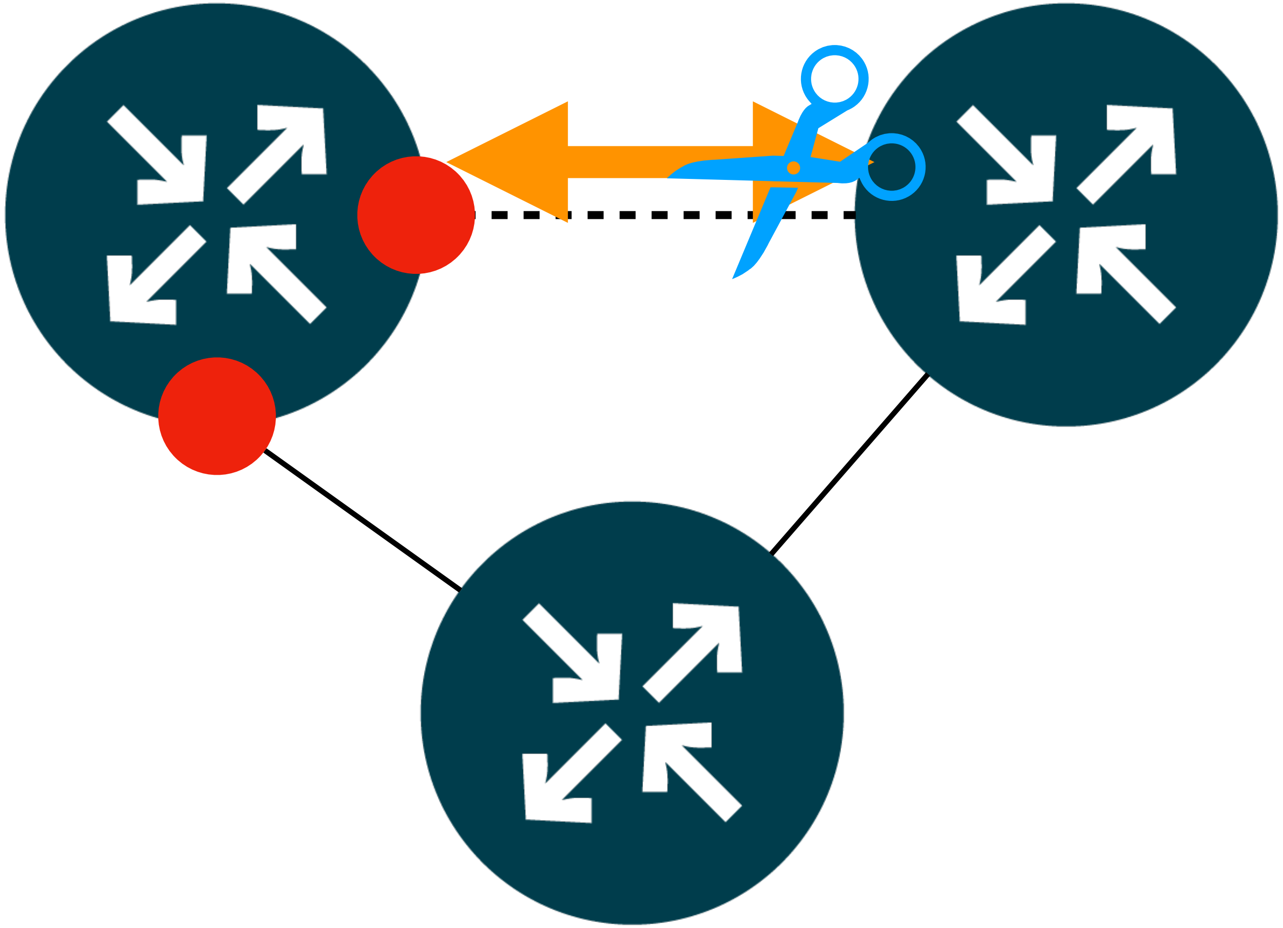
→address of the interface on which the **packets leave the router**



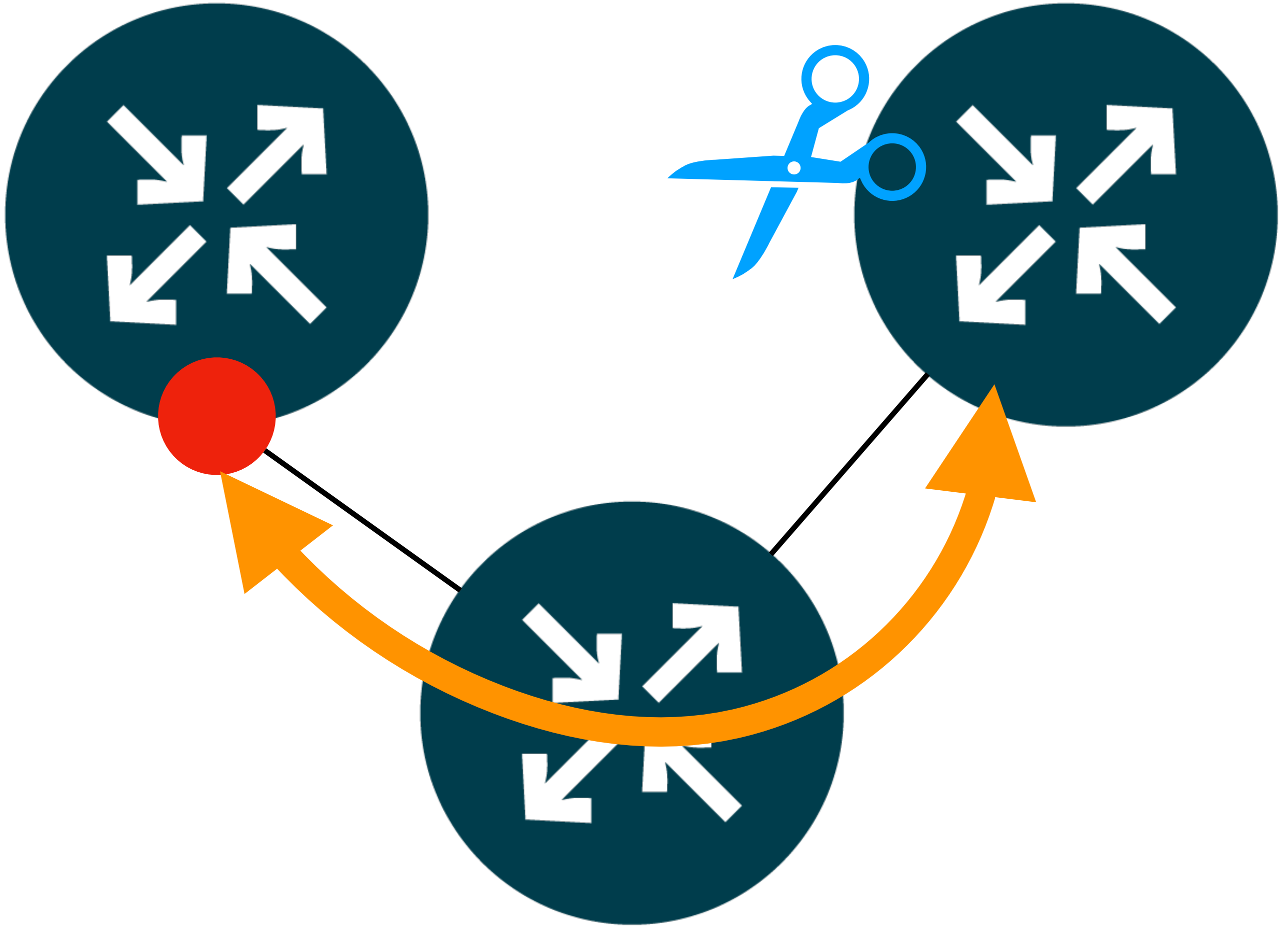
iBGP - Which IP addresses to use?



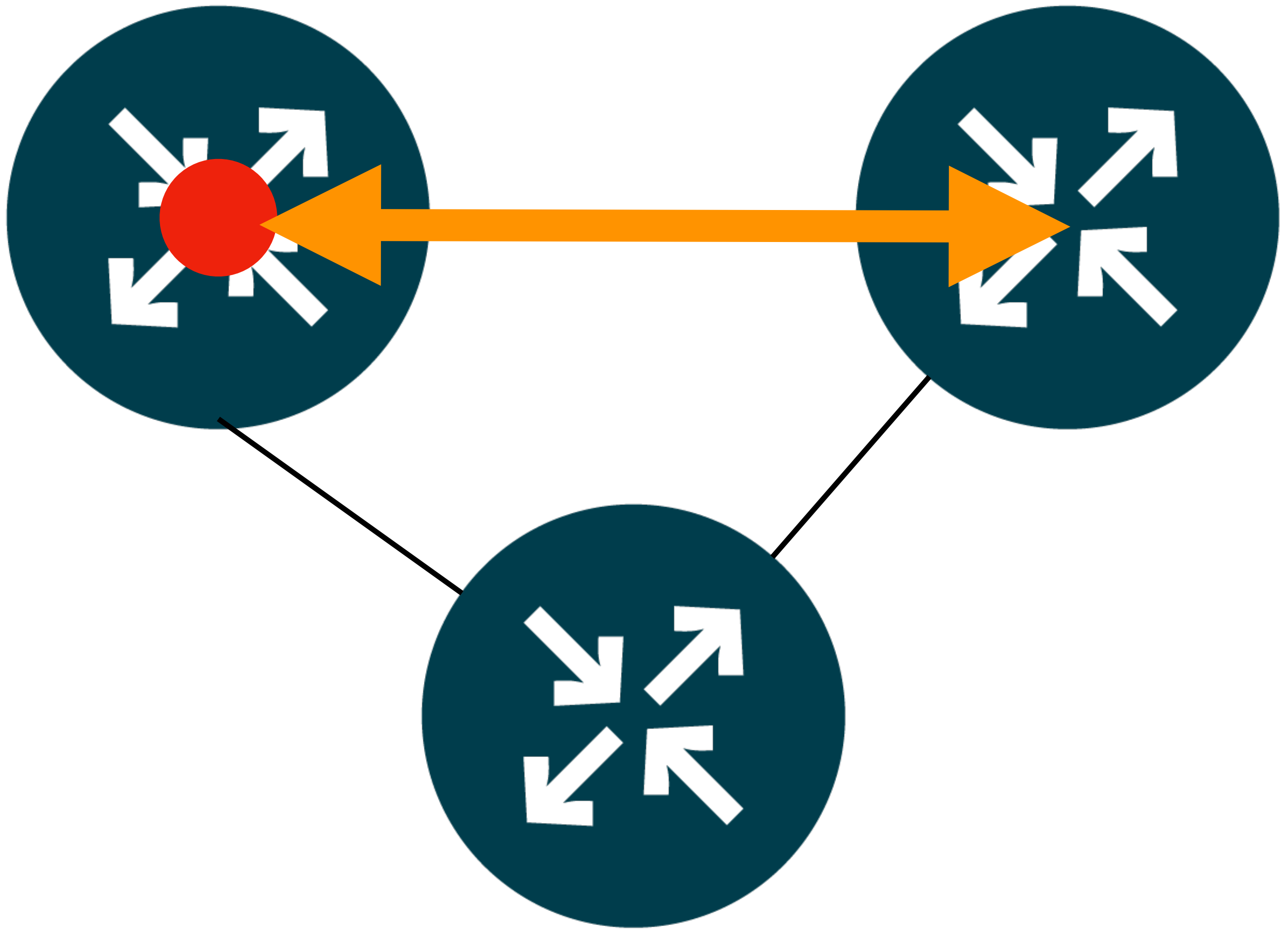
iBGP - Which IP addresses to use?



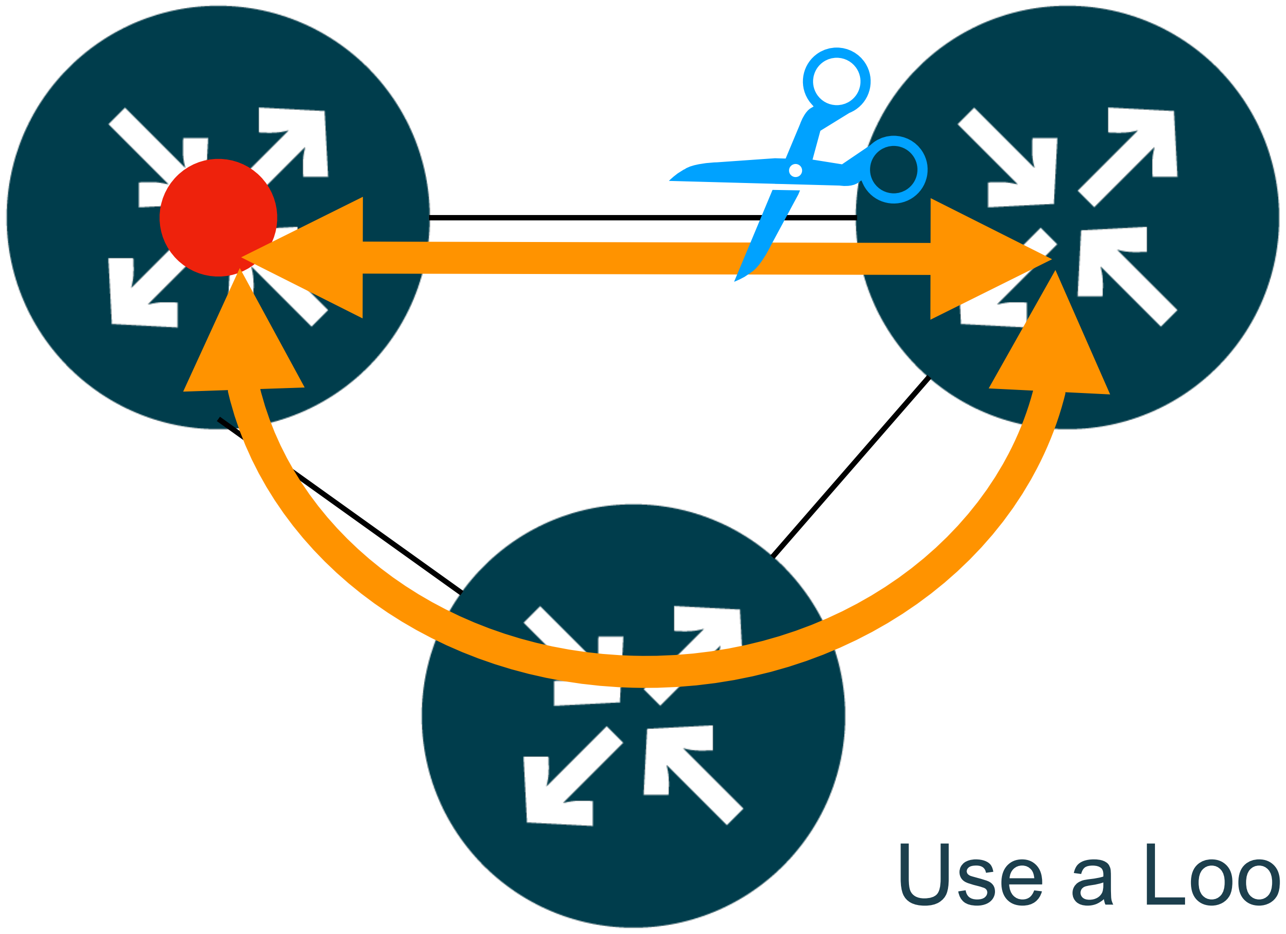
iBGP - Which IP addresses to use?



iBGP - Which IP addresses to use?

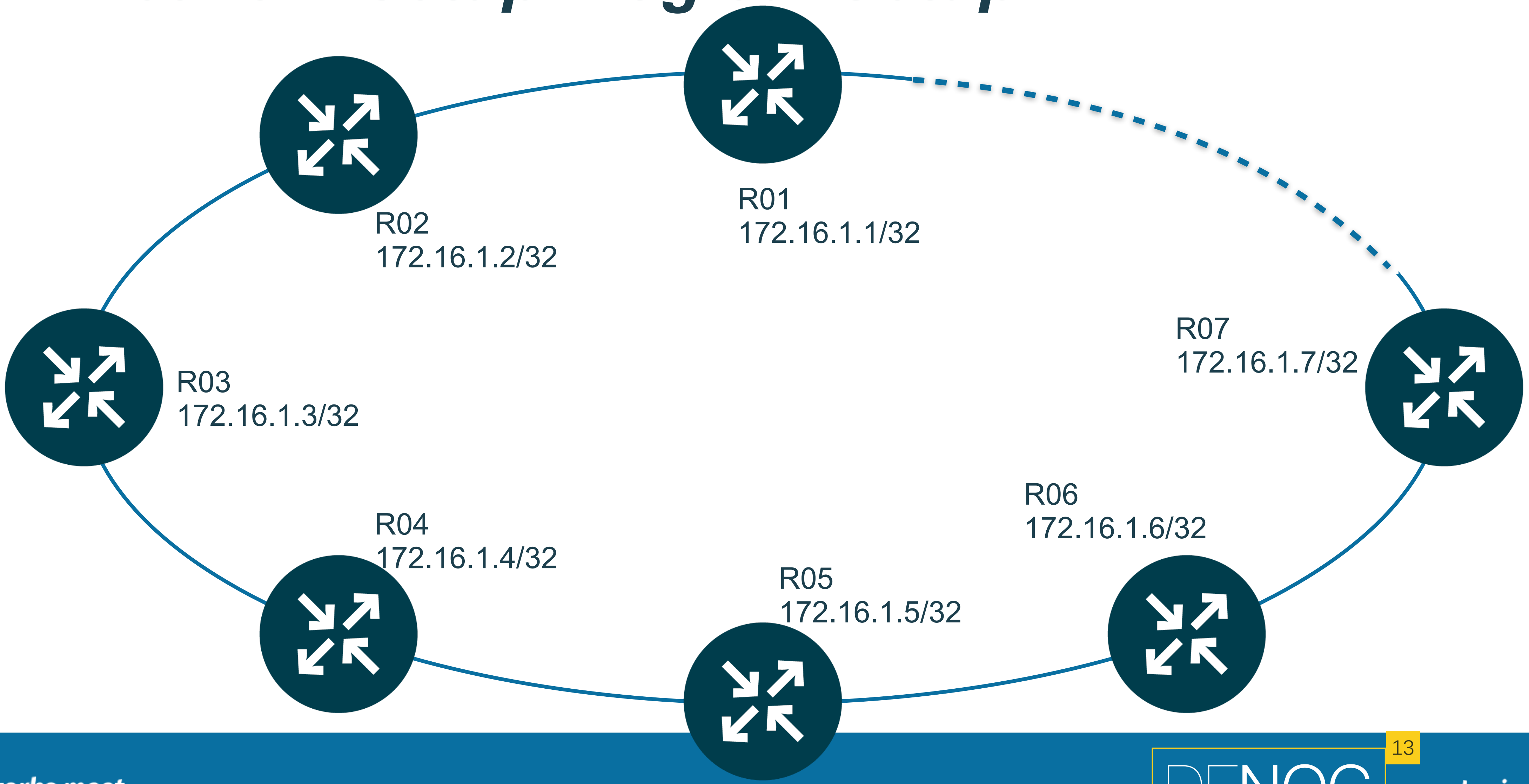


iBGP - Which IP addresses to use?

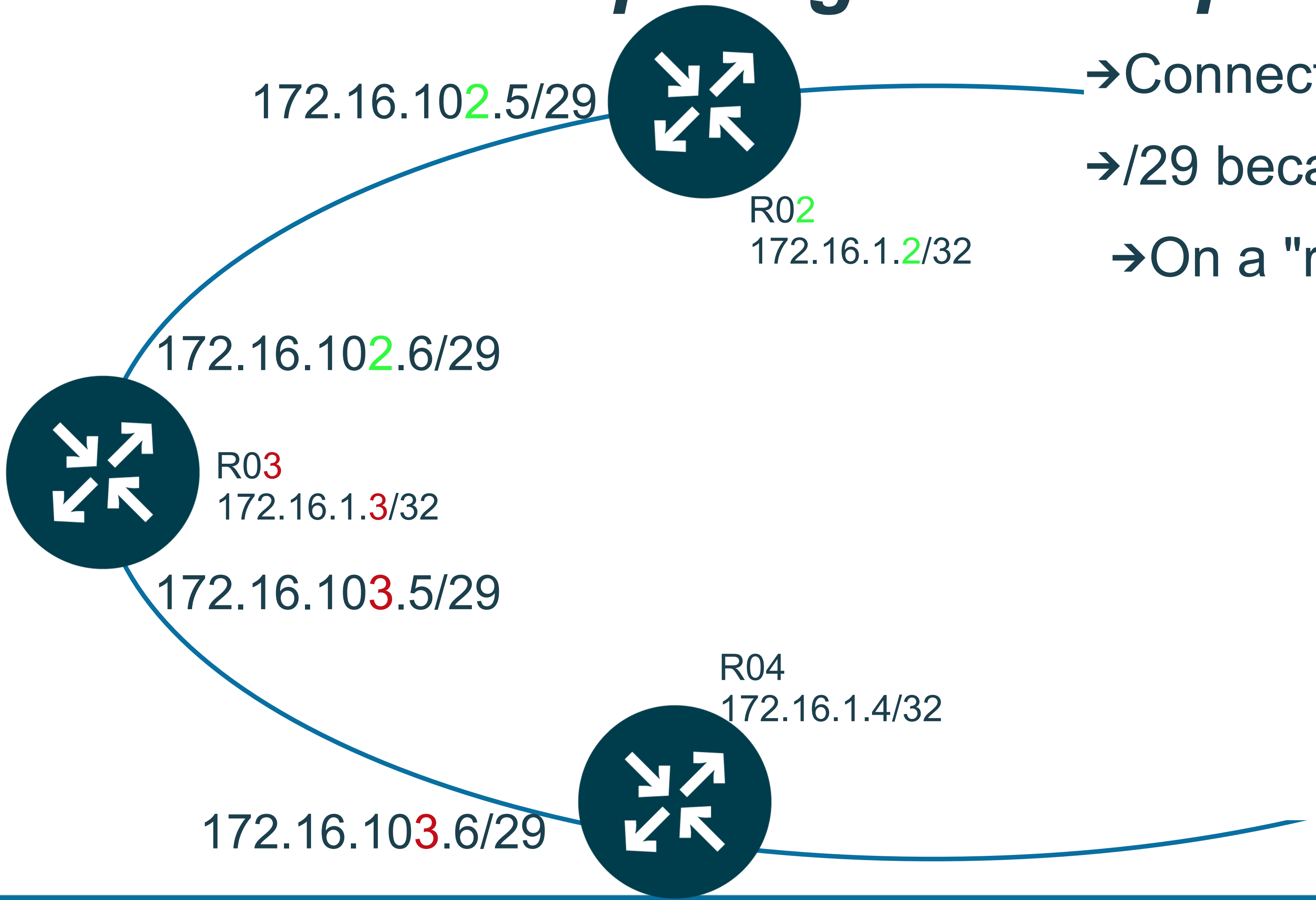


Use a Loopback address!

Network setup: Logical Setup

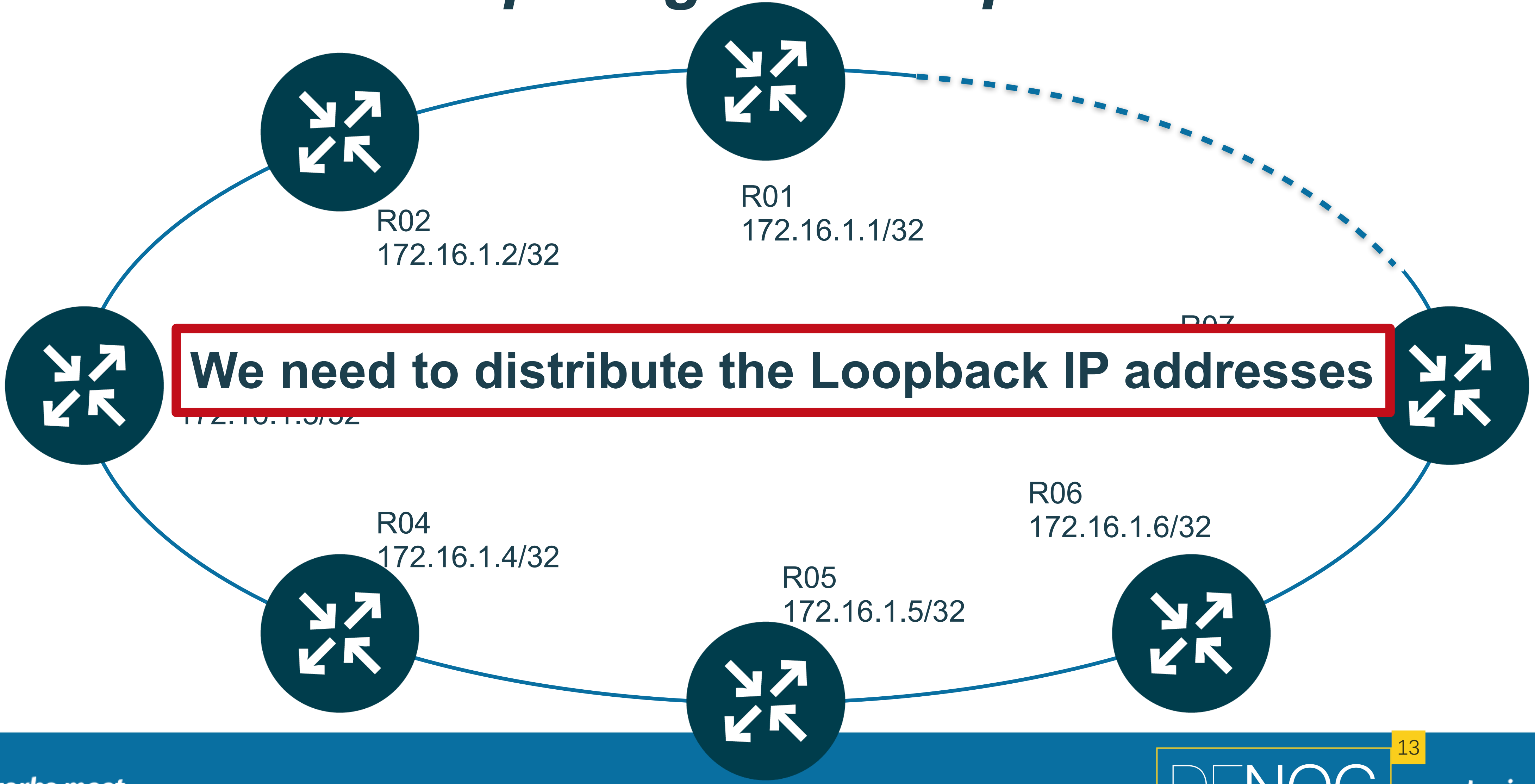


Network setup: Logical Setup



- →Connecting Interfaces
- →/29 because of Docker
  - →On a "real" router: /30

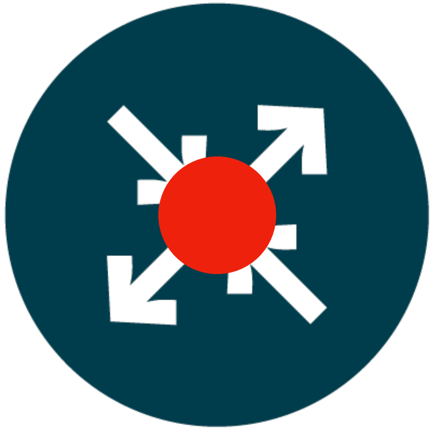
Network setup: Logical Setup



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 whats there

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

→Clean slate installation: Use IS-IS

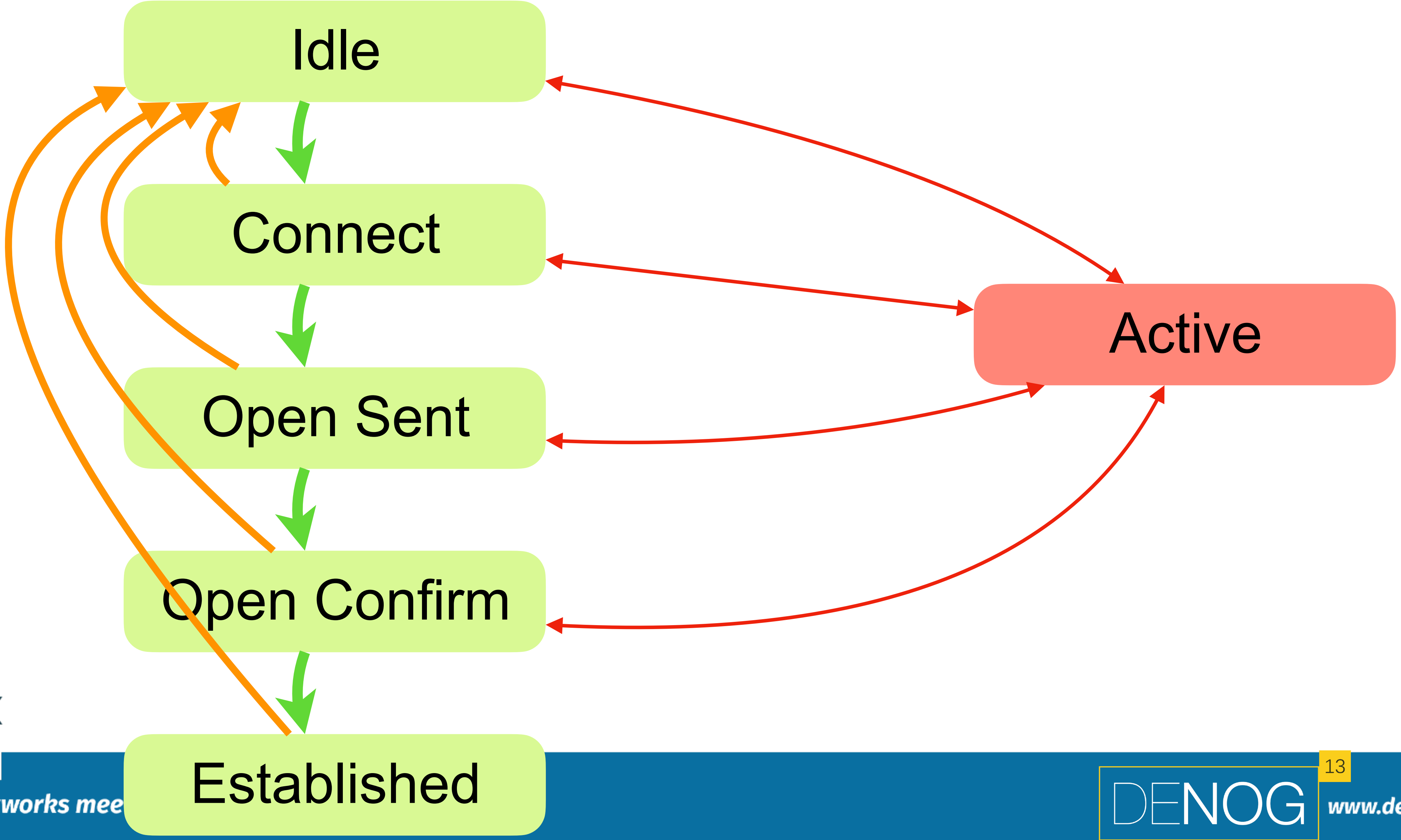
→Today: IS-IS is already set up in the lab

  →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

Life cycle of a BGP session (incomplete)



Experiment: Setup iBGP

experiment 01d + experiment 01e
prepare: ./1c-solution-isis -n XX

Save your config!

→write mem

iBGP - why fully meshed?

→BGP receives prefixes from external - eBGP

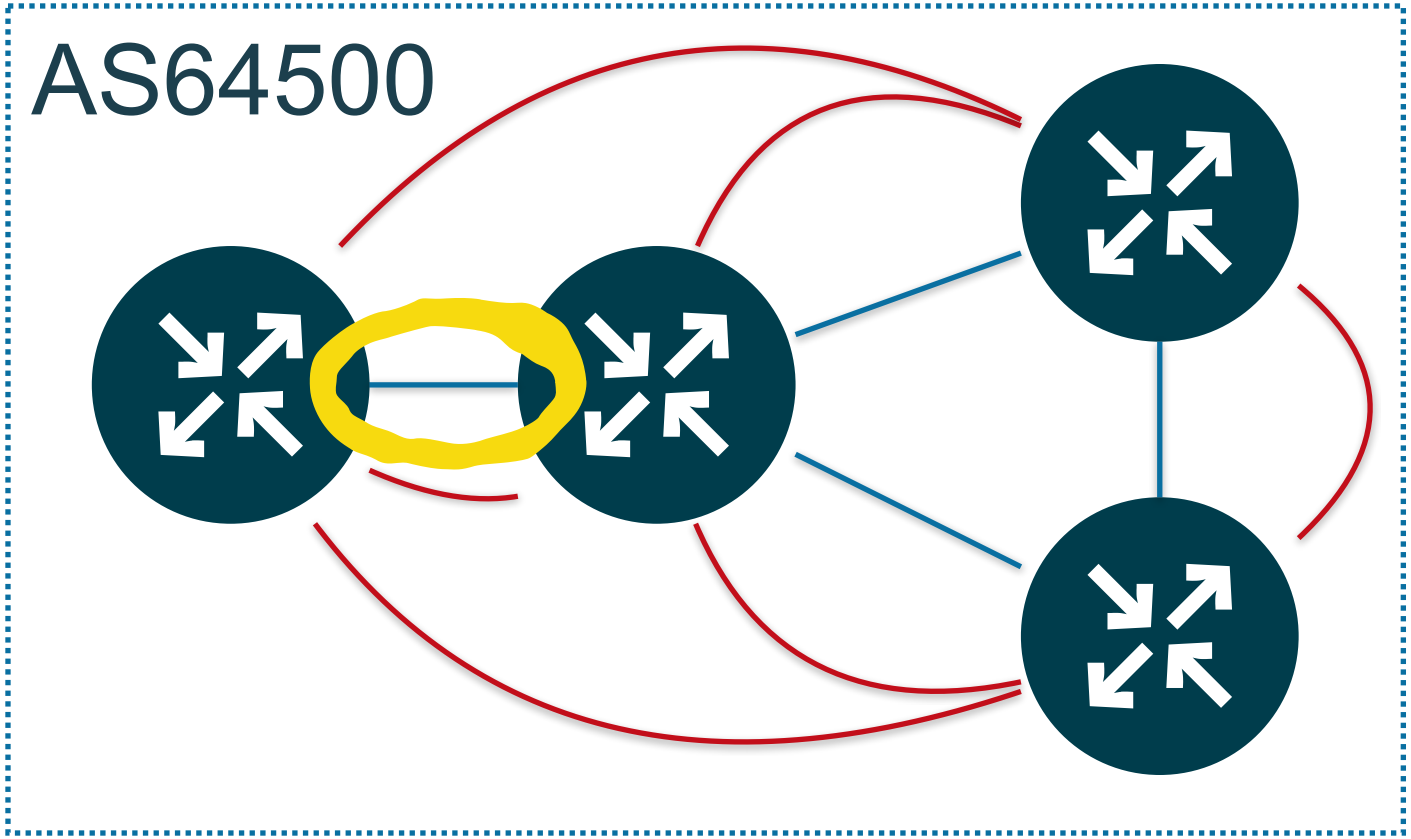
→BGP sends **all** prefixes to external (unless filtered)

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

→BGP does **not** send prefixes received from internal to internal

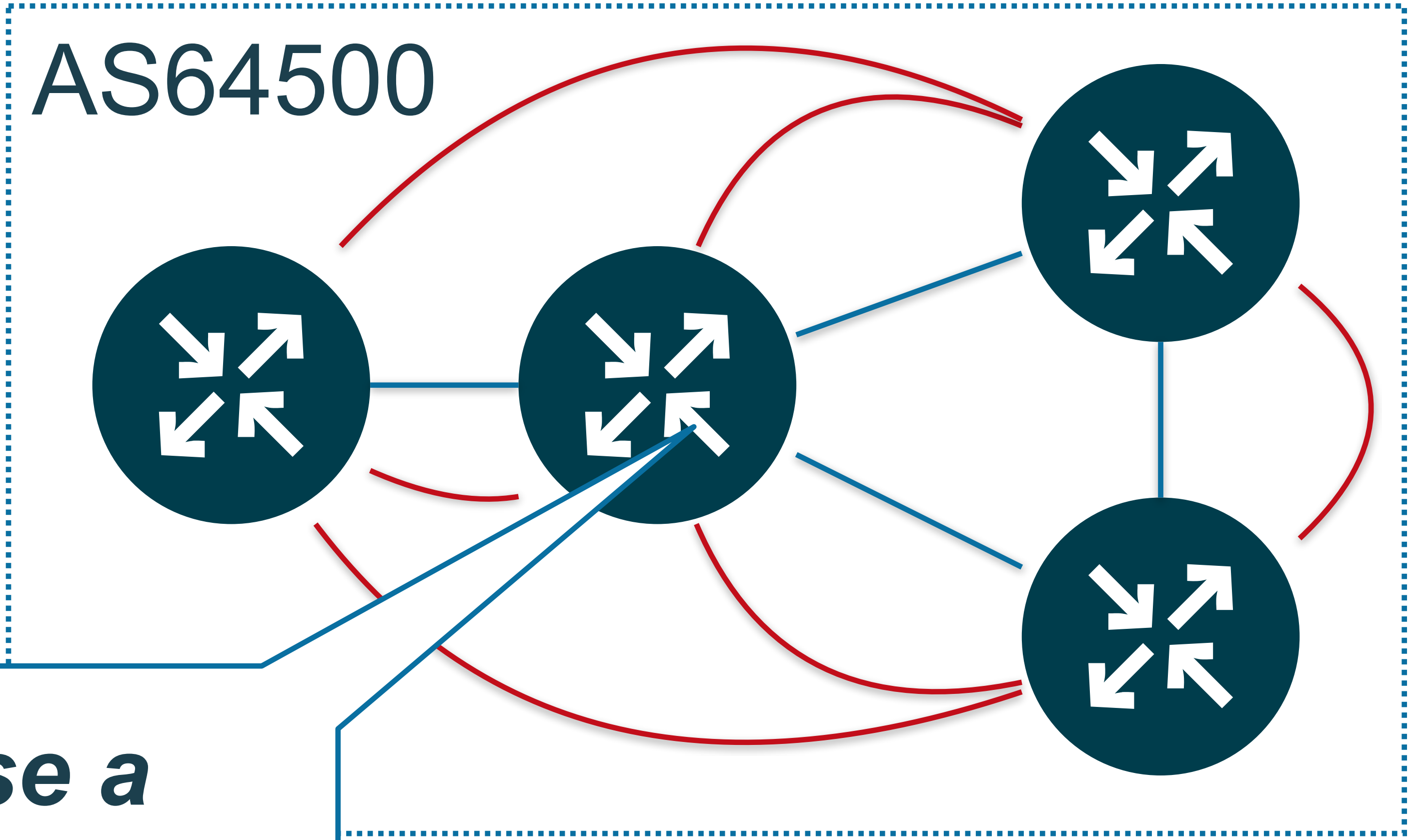
→**unless...**

Example Network: Fully meshed iBGP?



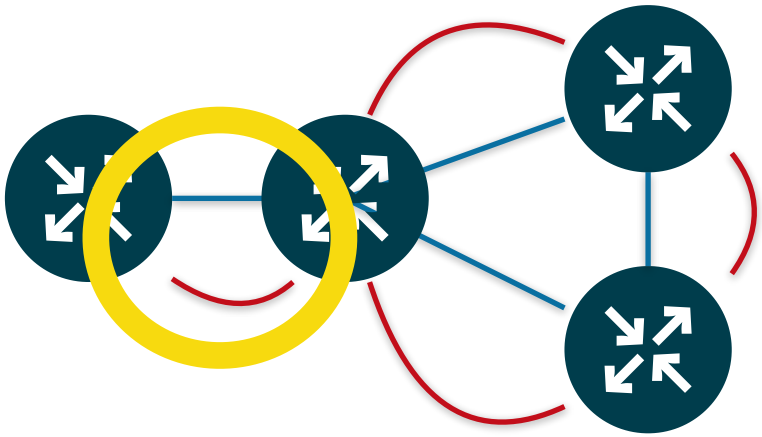
DE CIX

Example Network: ~~Fully meshed~~ iBGP?



Use a route reflector

iBGP: Route Reflector



→"Normal" prefix forwarding rule for iBGP

→do **not** 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

Open Questions?

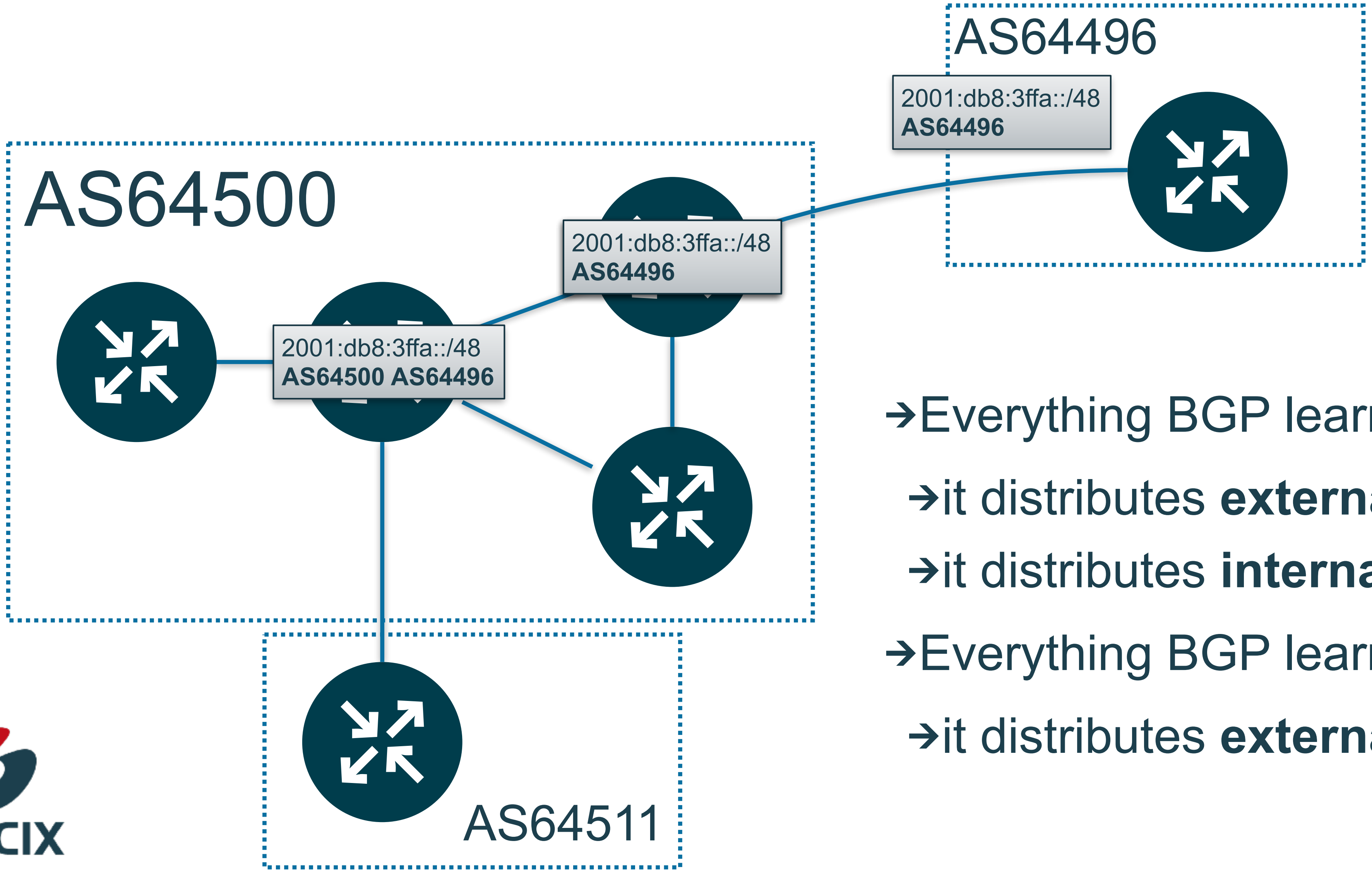
BGP for networks who peer

02a - Setup eBGP

Wolfgang Tremmel
academy@de-cix.net

Today's Training

- →IP Prefixes and AS Numbers
- →BGP: Introduction
- →iBGP and
- →eBGP
- →Becoming Multi-Homed
- →BGP Best Path Selection



→Everything BGP learns from **external**

→it distributes **external**

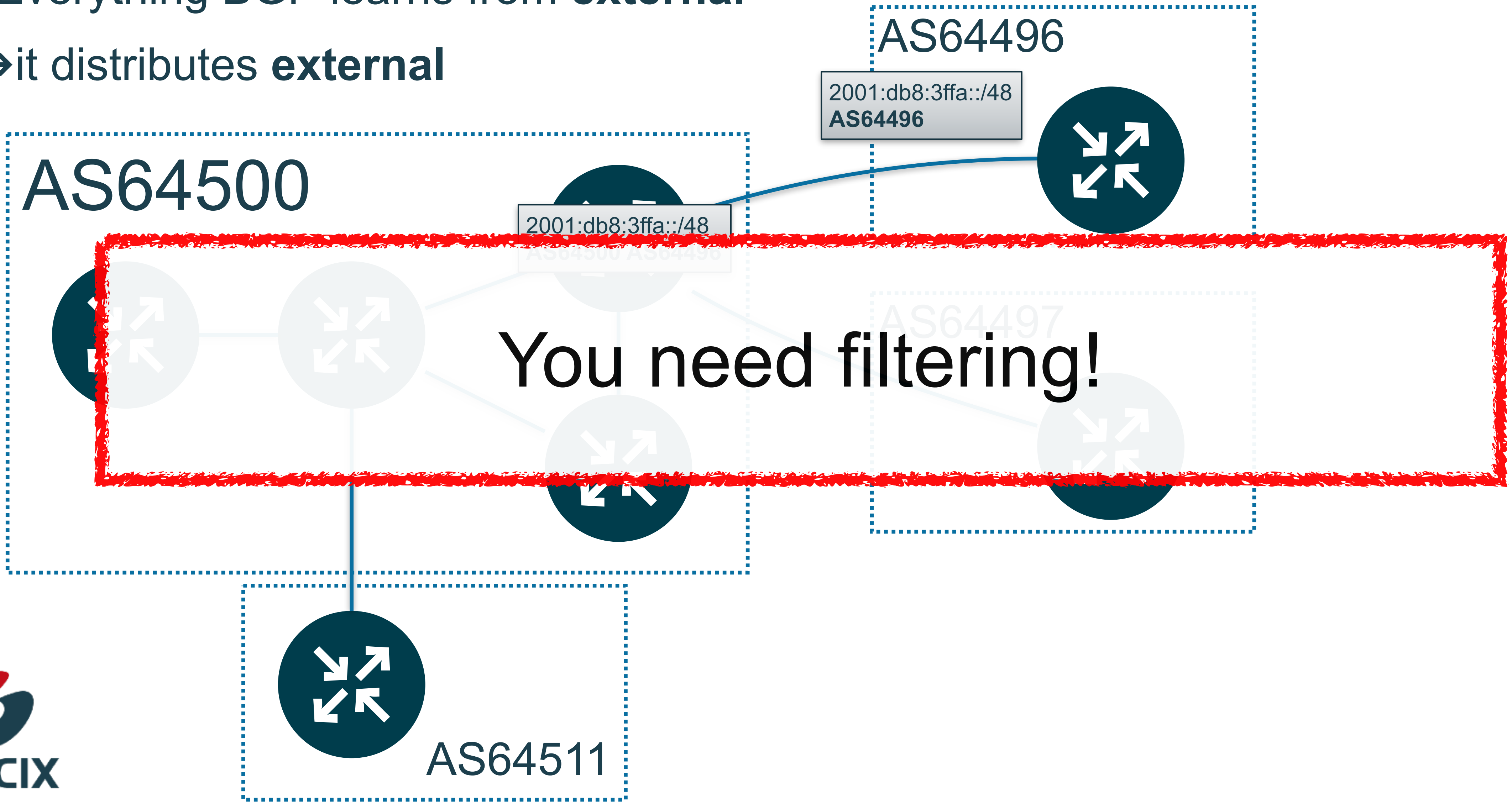
→it distributes **internal**

→Everything BGP learns from internal

→it distributes **external**

→Everything BGP learns from **external**

→it distributes **external**

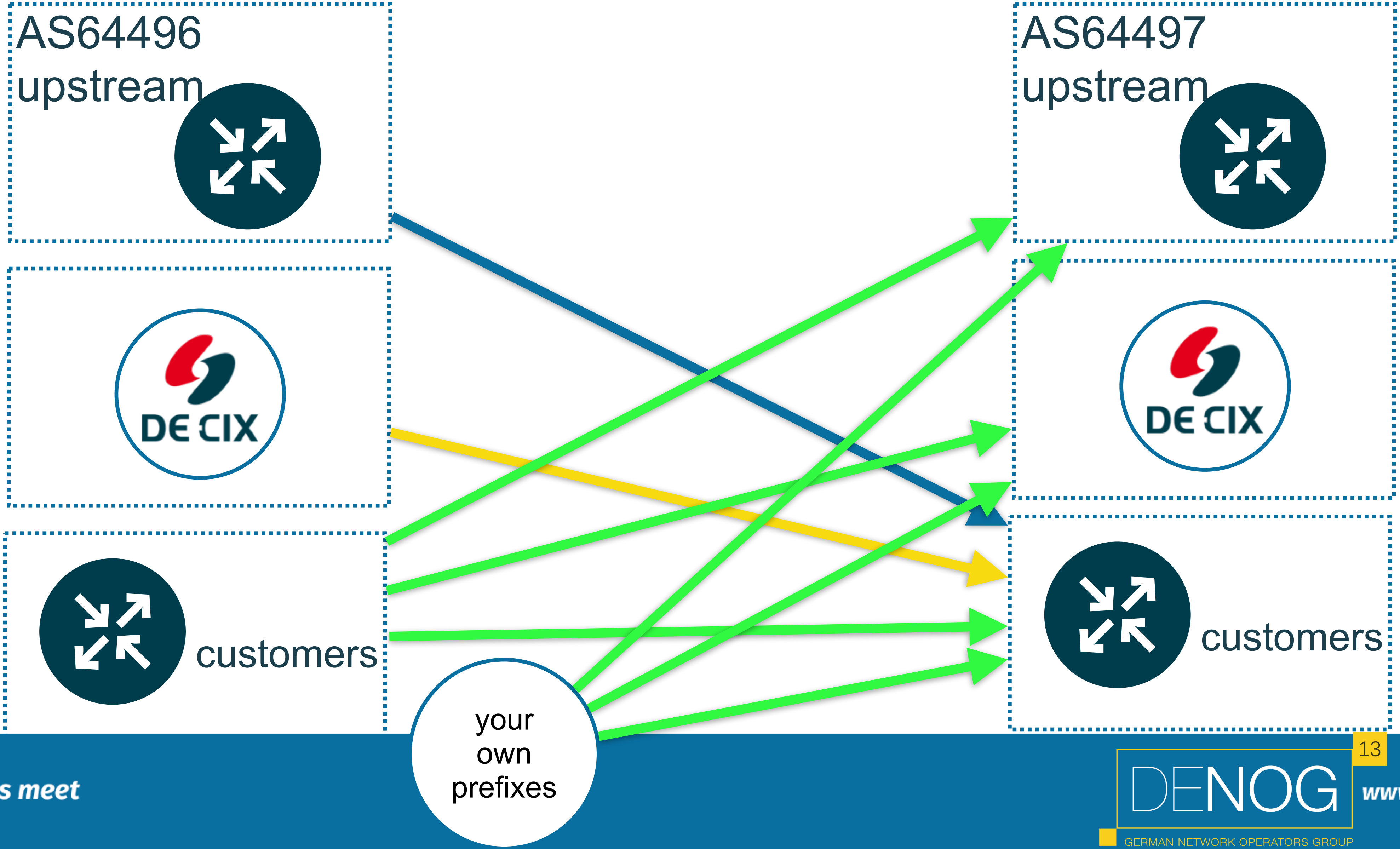


You need filtering!

- →You have multiple sources of prefixes
  - →upstream provider(s)
  - →peering(s)
  - →customer(s)
  - →your own prefixes!
- →And destinations to which you announce prefixes
  - →upstream providers
  - →peerings
  - →customers

your
own
prefixes

Prefixes
Sources and Destinations



Easy filtering for beginners

→Deny everything outgoing

→Allow everything incoming

```
route-map upstream-out deny 100
!
route-map upstream-in permit 100
!
```

→Open filters step by step to allow certain prefixes through

```
ip prefix-list my-networks permit 198.51.100.0/24
!
route-map upstream-out permit 50
 match ip address prefix-list my-networks
!
route-map upstream-out deny 100
```

About: Route Maps (in terms of Cisco and FRR)

→Each route-map has a name (and there is no check against typos)

→Each route-map consists of a ordered list of statements

→Just like a BASIC program (with line numbers)

→Each statement has a result of either permit or deny

```
route-map my-great-filter permit 10
```

About: Route Maps Statements

→Each statement has a result of either permit or deny

→also zero to many "*match" clauses*

→no *match clause* = **always true**

→more then one *match clause* are "**and**"ed together

→If match(es) evaluate true, route-map is terminated and result returned

```
route-map my-great-filter permit 10
 match ip address prefix-list my-list
```

About: Route Maps Statements

→route-maps also can have none to many set-statements

→if match-statements evaluate true (or if there are no match statements)

  →all set-statements are executed

  →route-map terminates and result is returned

```
route-map my-great-filter permit 10
 match ip address prefix-list my-list
 set local-preference 1000
```

Example: Filter for receiving prefixes

```
route-map upstream-in deny 10
 match ip address prefix-list ipv4-unwanted
 match ipv6 address prefix-list ipv6-unwanted

route-map upstream-in deny 20
 match as-path 100

route-map upstream-in permit 1000
 set local-preference 10
```

We start with simple filters for eBGP

→Configure filters (route-map) for in and out

```
route-map upstream-in permit 100
route-map upstream-out deny 100
```

→The **in** filter lets everything through

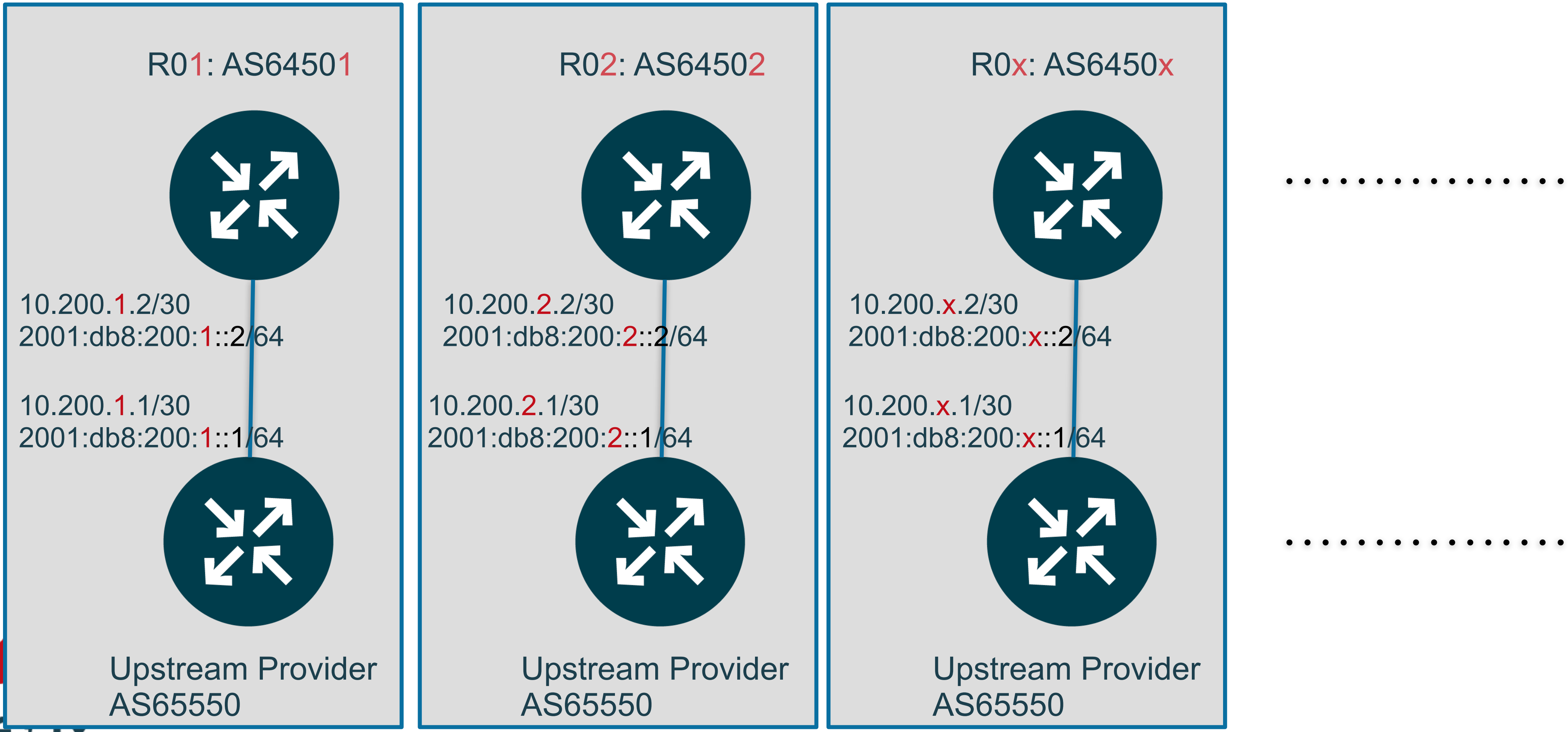
→no match statement = always true

→result "permit" is returned for every prefix

→The **out** filter blocks everything

→result "deny" is returned for every prefix

Network setup: Logical Setup



Configure eBGP: Peer Group

→we group common commands in a *peer group*

→we might have multiple upstreams with multiple AS numbers,

→ so we keep the remote AS in the neighbor config

→Remember our filters? **upstream-in** for in, **upstream-out** for out

```
router bgp 6450x
 neighbor upstream peer-group
 address-family ipv4 unicast
  neighbor upstream route-map upstream-in in
  neighbor upstream route-map upstream-out out
  neighbor upstream soft-reconfiguration inbound
  neighbor upstream activate
```

Configure eBGP: Neigbor(s)

→We have a peer-group, so we only need what is unique to each neighbor

→statements configured in the peer-group are inherited by each member

→In this case, this is only the AS number

→Neighbor IP address is different for each router

```
router bgp 6450X
  neighbor 10.200.X.1 remote-as 65550
  neighbor 10.200.X.1 peer-group upstream
```

Experiment: Configure eBGP

experiment 02a - Setup eBGP
start exabgp!

Summary

→BGP uses TCP

→eBGP is BGP between Autonomous Systems

→BGP distributes prefixes

  →from external to internal

  →from internal to external

  →from external to external

    →Filtering!

Comparing iBGP and eBGP

	iBGP	eBGP
Distributes Prefixes	learned from external	learned from external learned from internal
Neighbor AS	same as own AS	different to own AS
Next-Hop IP address	unchanged!	set to IP of receiver
more differences?	Yes	Yes

BGP for networks who peer

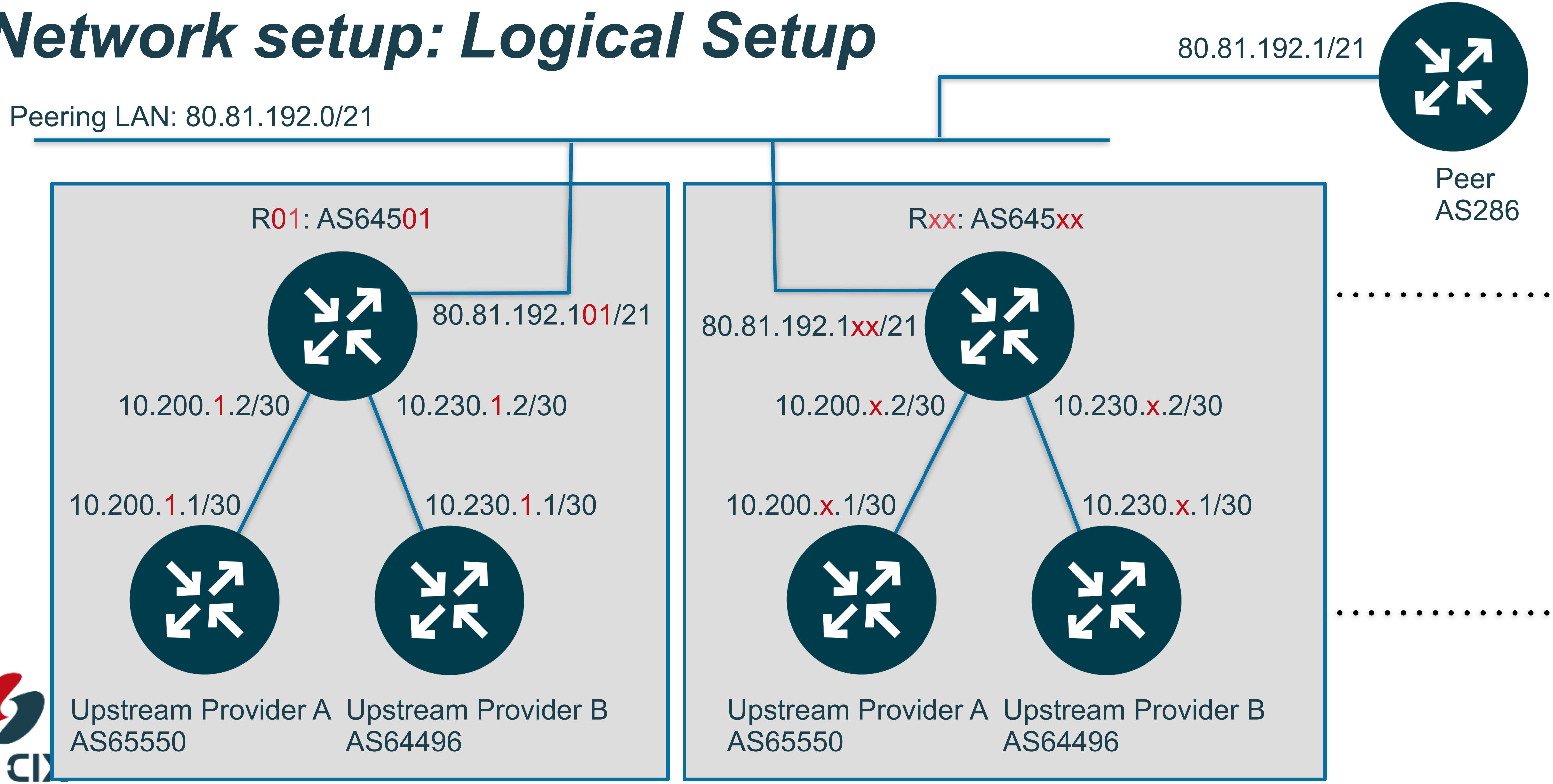
02b - Become Multi-Homed

Wolfgang Tremmel

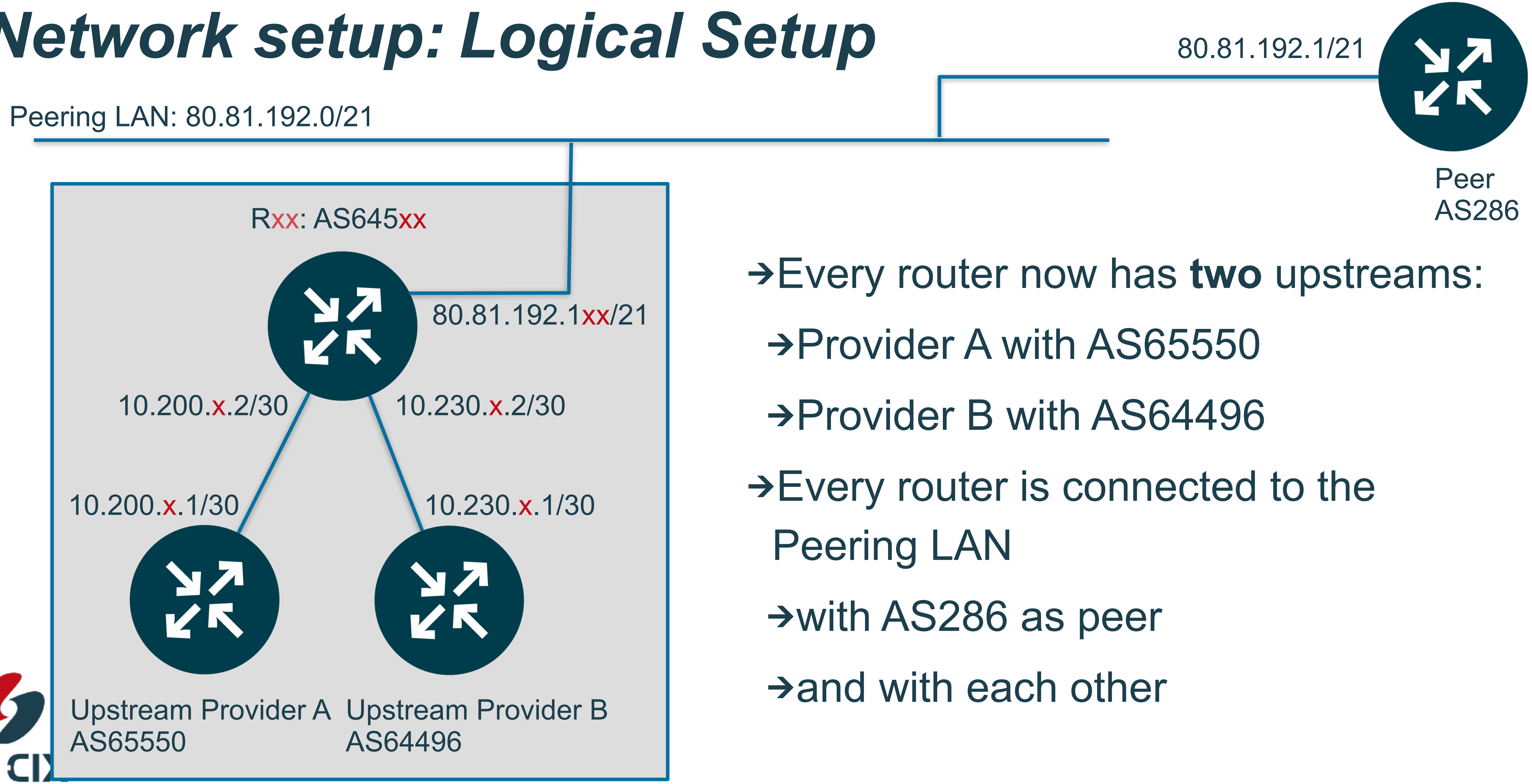
academy@de-cix.net

DE CIX

Network setup: Logical Setup



Network setup: Logical Setup



- →Every router now has **two** upstreams:
  - →Provider A with AS65550
  - →Provider B with AS64496
- →Every router is connected to the Peering LAN
  - →with AS286 as peer
  - →and with each other

Experiment: Configure eBGP

experiment 02b - become multi homed (2 upstreams)

BGP - becoming multihomed

Adding multiple upstreams and peering

Wolfgang Tremmel
wolfgang.tremmel@de-cix.net

Today's Training

- →IP Prefixes and AS Numbers
- →BGP: Introduction
- →iBGP and eBGP
- →Becoming Multi-Homed
- →BGP Best Path Selection

Why do we do this?

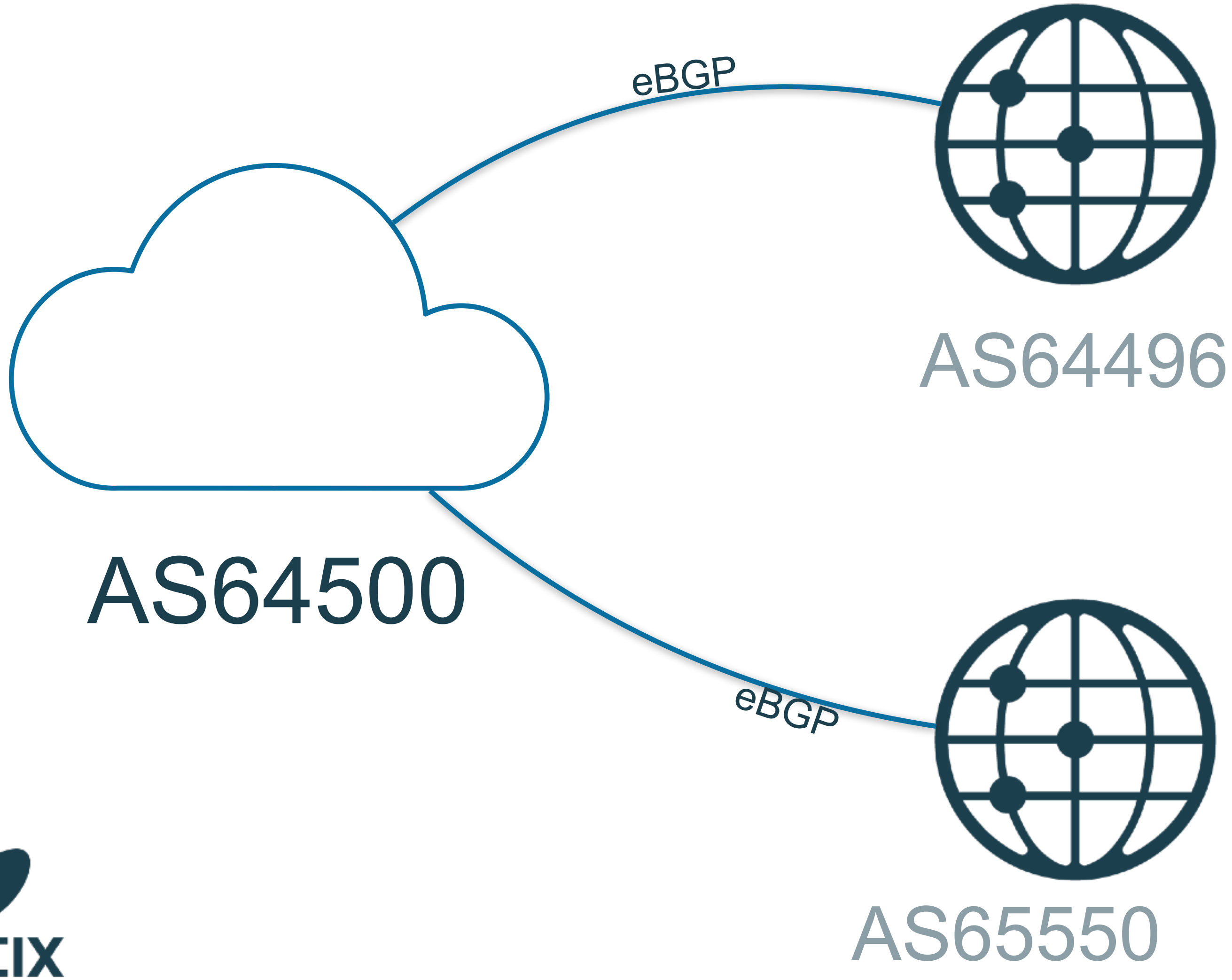
	With multiple Upstreams and Peering	Without an AS or with just one upstream
Redundancy	In case of a problem with one upstream you have a second one	If your upstream has a problem, you have a problem
Control	You have full control over your outgoing traffic	Your upstream ISP controls your traffic
Cost	You can optimize your traffic for cost	You just pay your one upstream ISP
Peering	You can setup your own peering policy and have full control	Your upstream ISP makes all decisions

Multi_{homed}

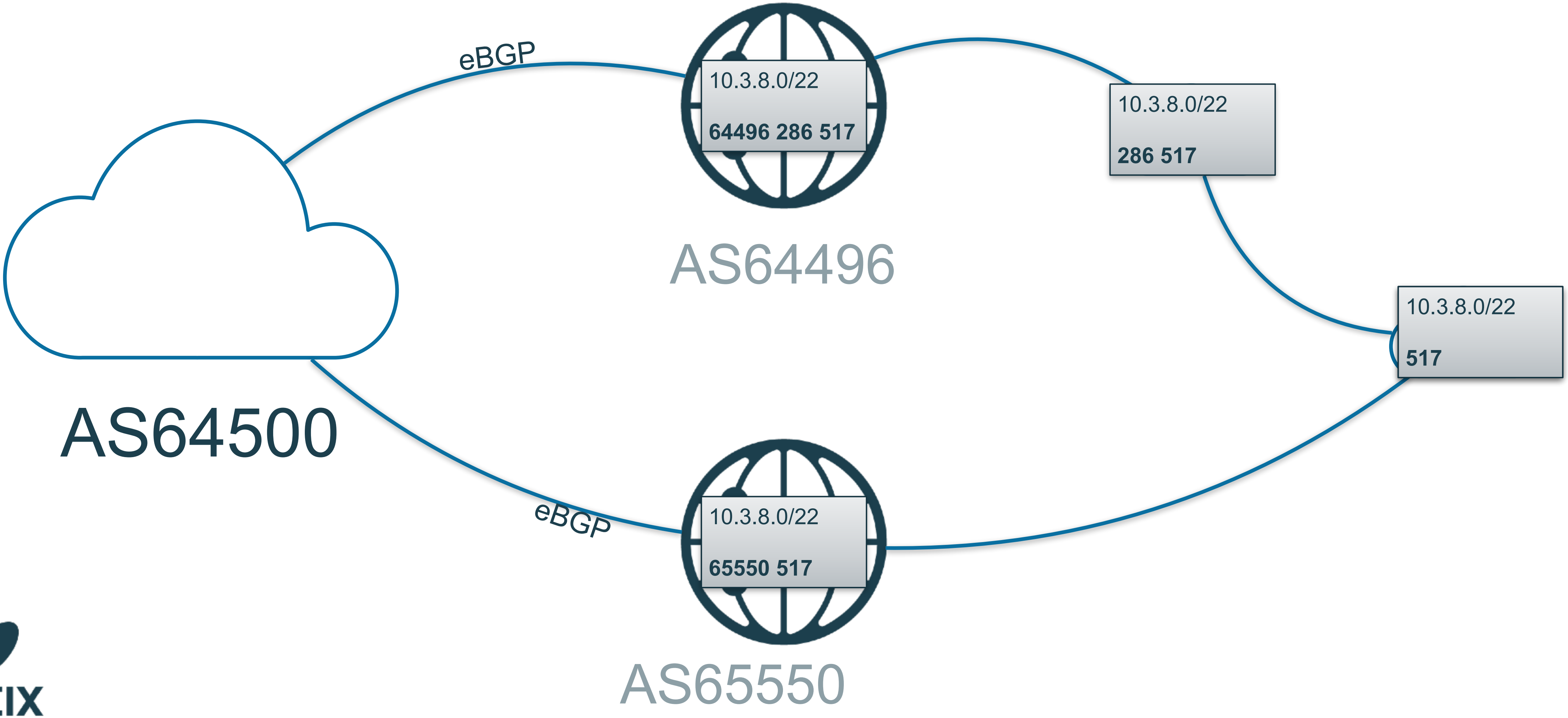
→ Multiple Upstreams
- For redundancy
- For cost optimization

→ Peering
- For even better performance
- For even more resilience

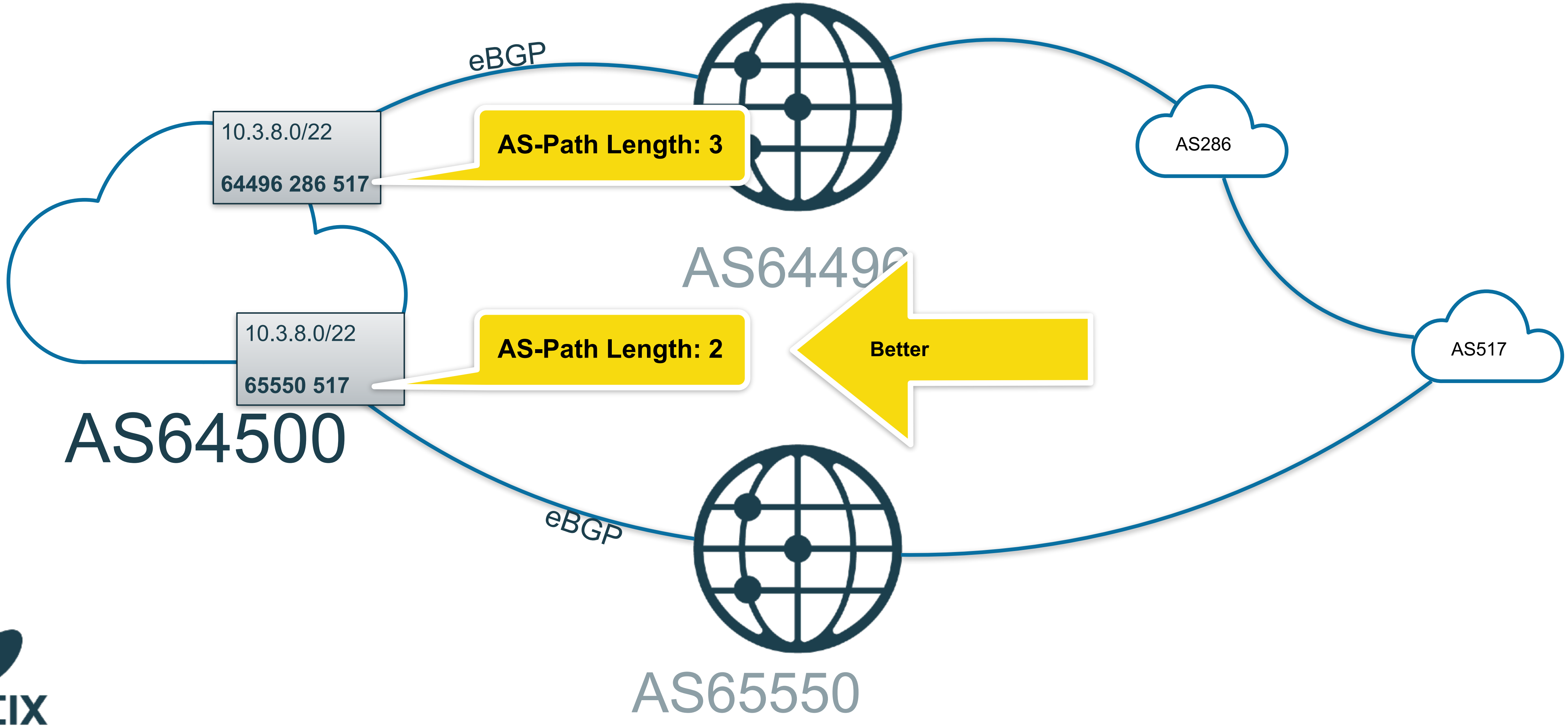
Let's get started.... with two upstreams



Let's get started.... with two upstreams



Let's get started.... with two upstreams



The BGP Routing Algorithm

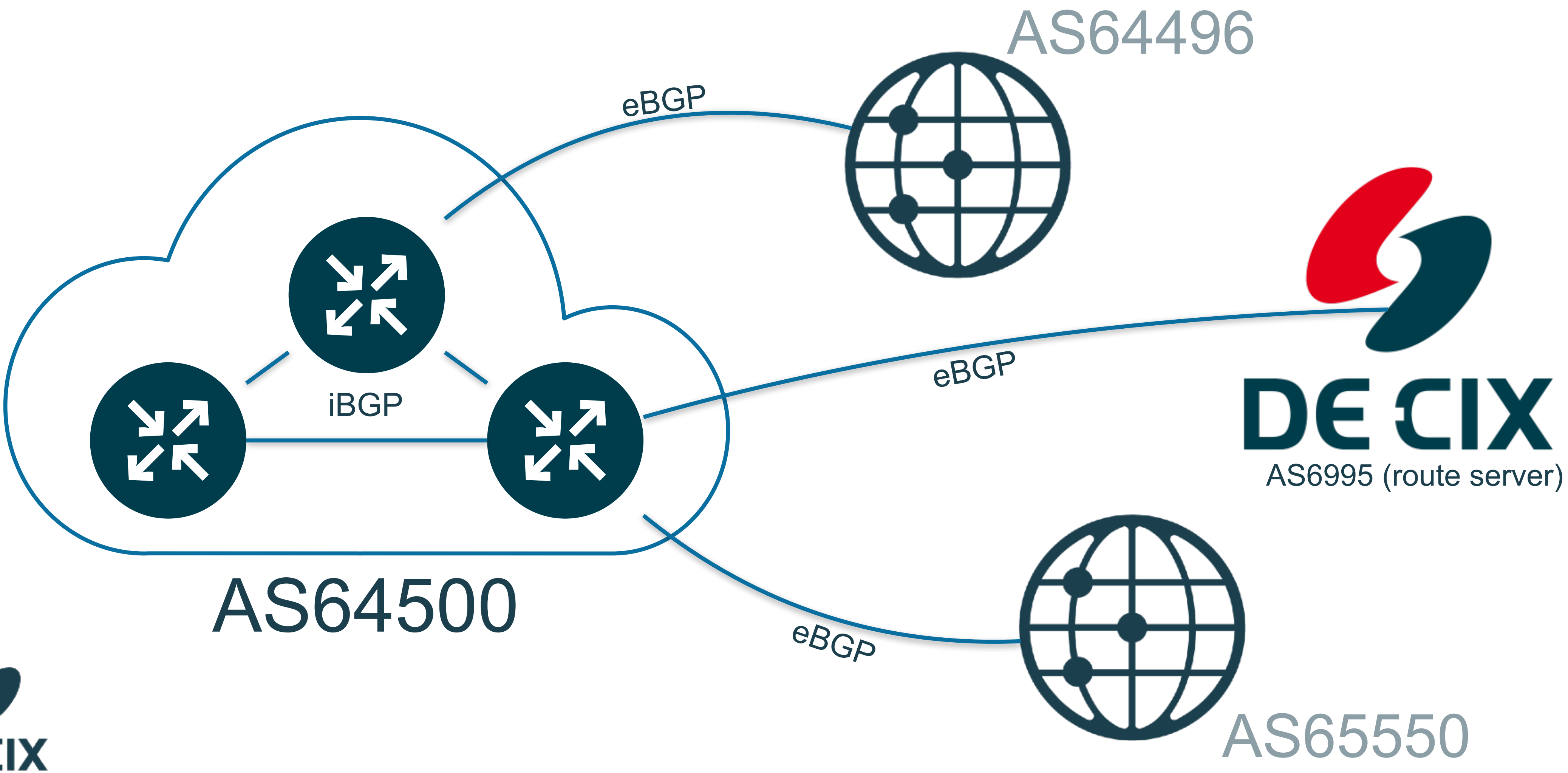
1	NextHop reachable?	Continue if "yes"
2		
3		
4		
5		
6		
7		
8		
9		
10		

AS-Path Length: 3

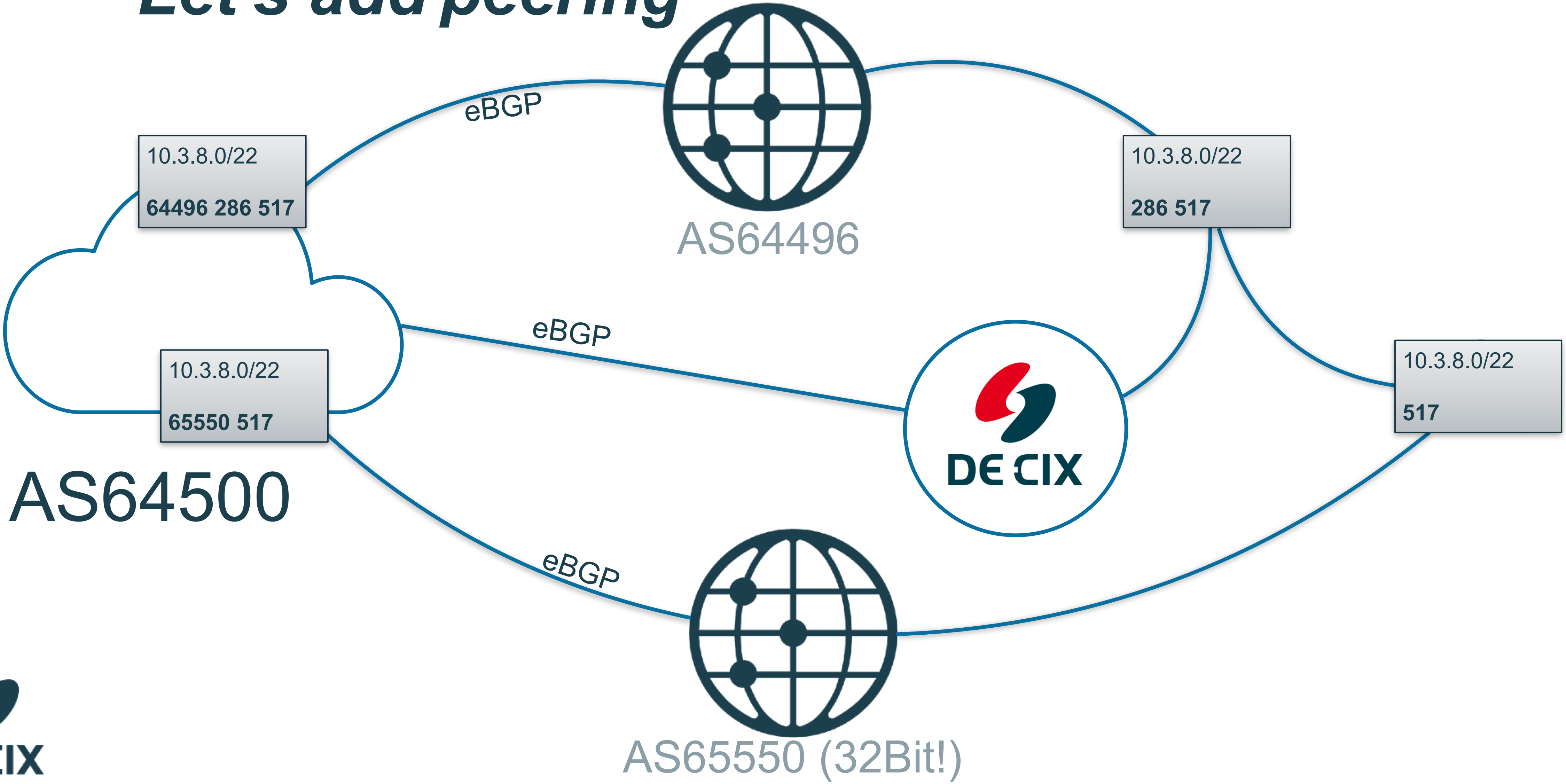
AS-Path Length: 2

Better

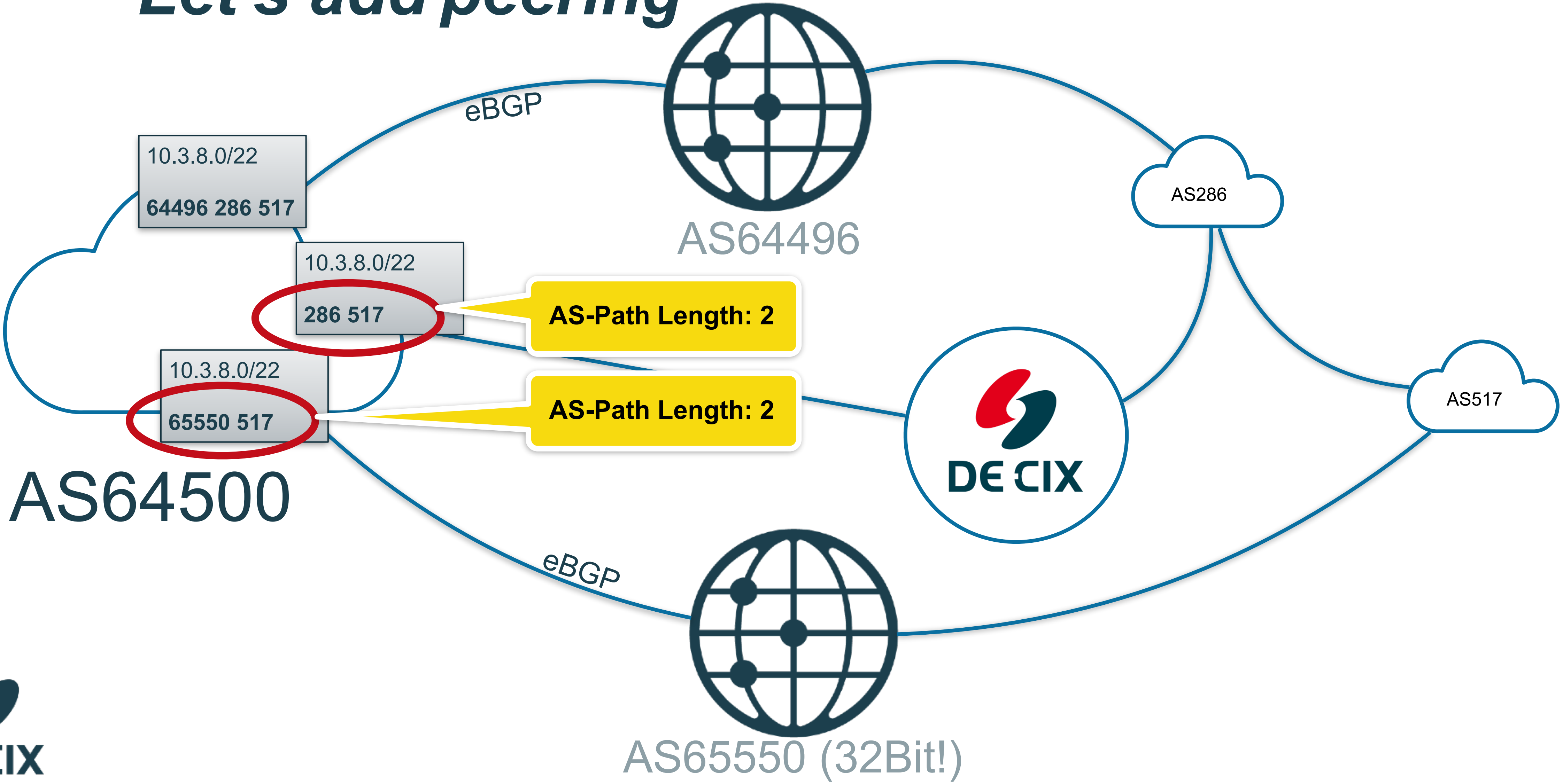
Let's continue...



Let's add peering



Let's add peering



The BGP Routing Algorithm

1	NextHop reachable?	Continue if "yes"
2		
3	AS Path Length	shorter wins
4		
5		
6		
7		
8		
9		
10		

AS-Path Length: 2

AS-Path Length: 2

Local Preference

- → Higher wins
- → Integer value (32bit, 0-4294967295)
- → Propagated via iBGP inside an Autonomous System
- → Set using a route-map when receiving prefixes
- → Typical values:
  - Customer prefixes: 10000
  - Peering prefixes: 1000
  - Upstream prefixes: 10

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4		
5		
6		
7		
8		
9		
10		

Local Preference - how to set

→ High level:

```
if (prefix received from customer)
  then set local-preference of prefix = 10000
else if (prefix received from peer)
  then set local-preference of prefix = 1000
else
  set local-preference of prefix = 10
```

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4		
5		
6		
7		
8		
9		
10		

→ Our experiment

```
route-map peering-in permit 100
  set local-preference 1000
route-map upstream-in permit 100
  set local-preference 10
```

Experiment: Configure eBGP

experiment 02b - become multi homed (add peering)

Summary

- → When connecting to multiple upstreams ISPs and peering, you need to define a routing policy
- → This policy changes attributes of **received** prefixes
- → This policy defines how your **outgoing** traffic is routed
- → *Local Preference* can be used to influence this
- → Otherwise *AS Path Length* is used to find the best path
- → BGP has a complex route selection algorithm

BGP route selection algorithm

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4		
5		
6		
7		
8		
9		
10		

BGP - Best Path Selection

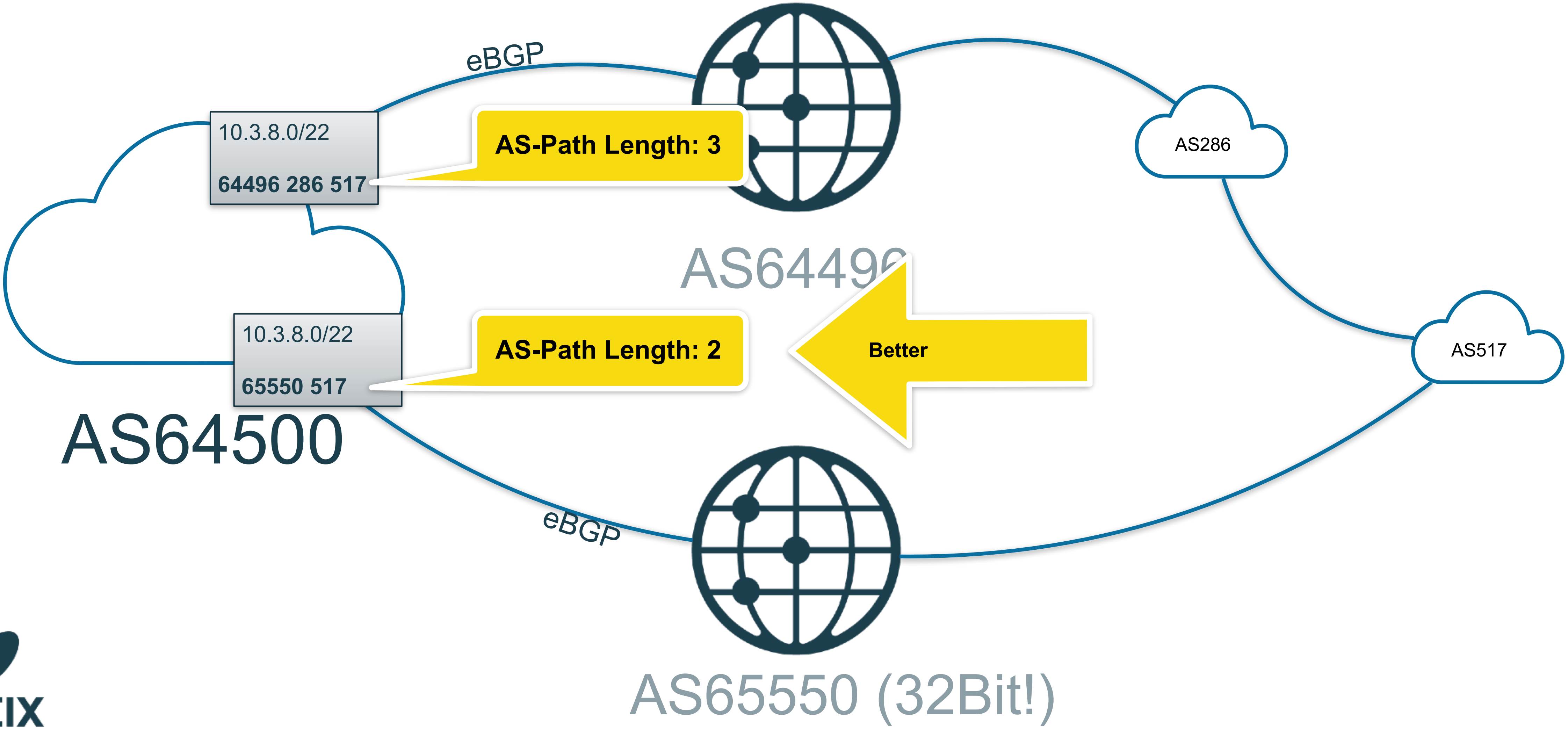
Beyond LocalPref and AS-Path Length

Wolfgang Tremmel
wolfgang.tremmel@de-cix.net

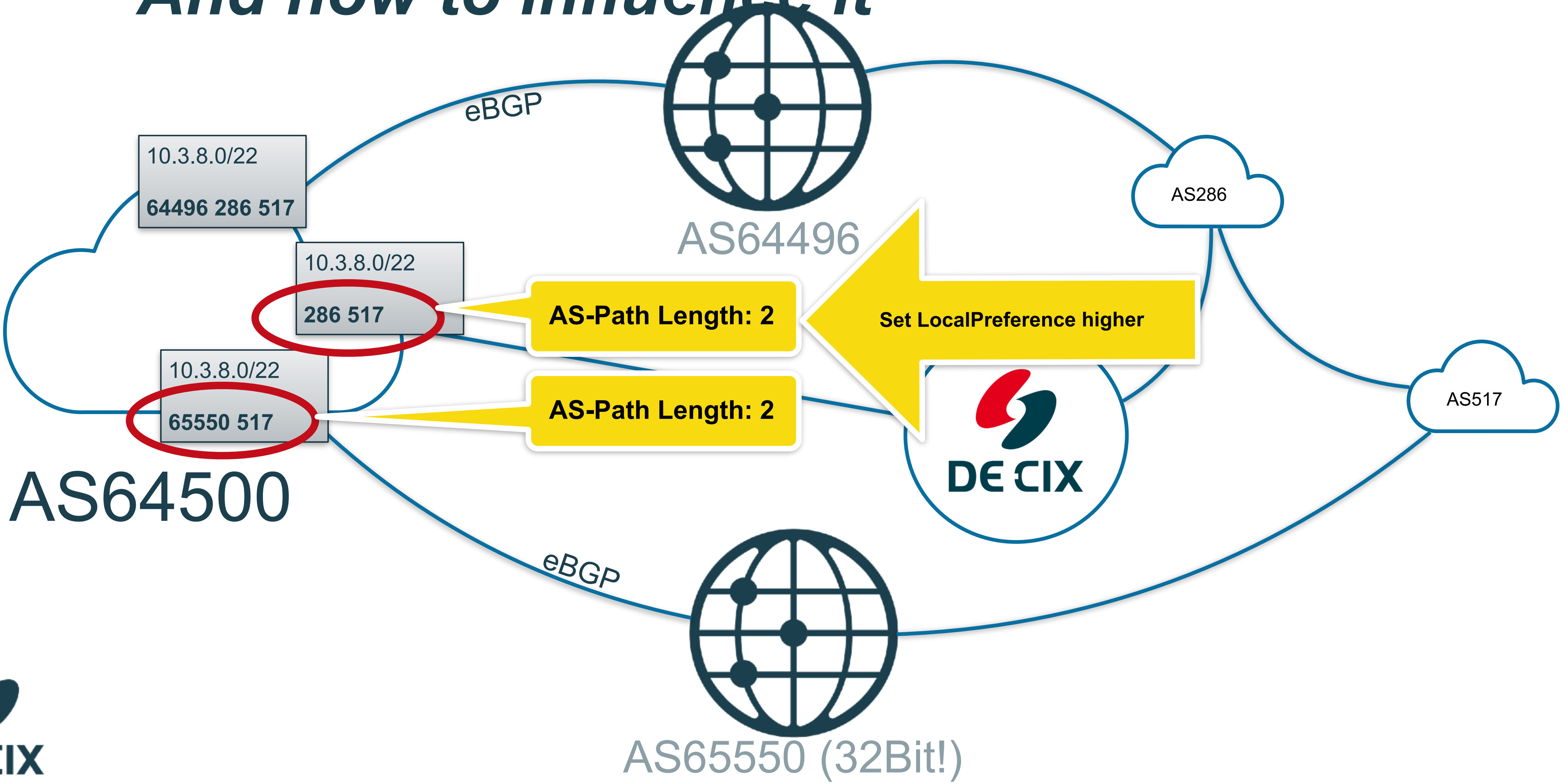
Today's Training

- →IP Prefixes and AS Numbers
- →BGP: Introduction
- →iBGP and eBGP
- →Becoming Multi-Homed
- →BGP Best Path Selection

We talked about path selection



And how to influence it

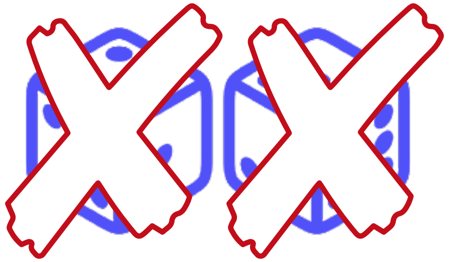


BGP Route Selection Algorithm

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4		
5		
6		
7		
8		
9		
10		

BGP Route Selection Algorithm: Motivation

- → Only one single path for each destination is needed (and wanted)
- → Decision must be based on attributes
- → And must not be random, but deterministic
- → Some of the criteria will sound strange
- → Some are really outdated
- → So we will focus on the most important ones
- → But all will be covered.



1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4		
5		
6		
7		
8		
9		
10		

Experiment: best path selection

Experiment 3.01: Local Preference
Experiment 3.02: AS Path Length

BGP Route Selection: Origin Type

- Origin Type is a "historical" attribute
- Three possible values:
  - IGP - route is generated by BGP network statement
  - EGP - route is received from EGP
  - incomplete - redistributed from another protocol
- ***This rule is not really important***

Exterior **G**ateway **P**rotocol

Predecessor of BGP which is no longer used

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4		
5		
6		
7		
8		
9		
10		

BGP Route Selection: Origin Type Examples

```
show ip bgp

Origin codes: i - IGP, e - EGP, ? - incomplete
* i1.0.4.0/22     206.130.10.8    634     200       0 6939 i
* i1.0.137.0/24   80.81.194.12 5000       200       0 9318 23969 ?
```

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4	Origin Type	IGP over EGP over Incomplete
5		
6		
7		
8		
9		
10		

BGP Route Selection: Origin Type Examples

```
show ip bgp 1.0.4.0/22

Path #22: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.10 0.11
  Advertised to peers (in unique update groups):
    46.31.120.208
  6939 4826 38803 56203
    206.130.10.8 from 206.130.10.252 (206.130.10.252)
      Origin IGP, metric 634, localpref 200, vali
import-candidate, import suspect
      Received Path ID 0, Local Path ID 1, version
      Community: 51531:35214 65101:0 65102:200 65103:
      Origin-AS validity: not-found
```

1		NextHop reachable?	Continue if "yes"
2		Local Preference	higher wins
3		AS Path Length	shorter wins
4		Origin Type	IGP over EGP over Incomplete
5			
6			
7			
8			
9			
10			

BGP Route Selection: Origin Type Examples

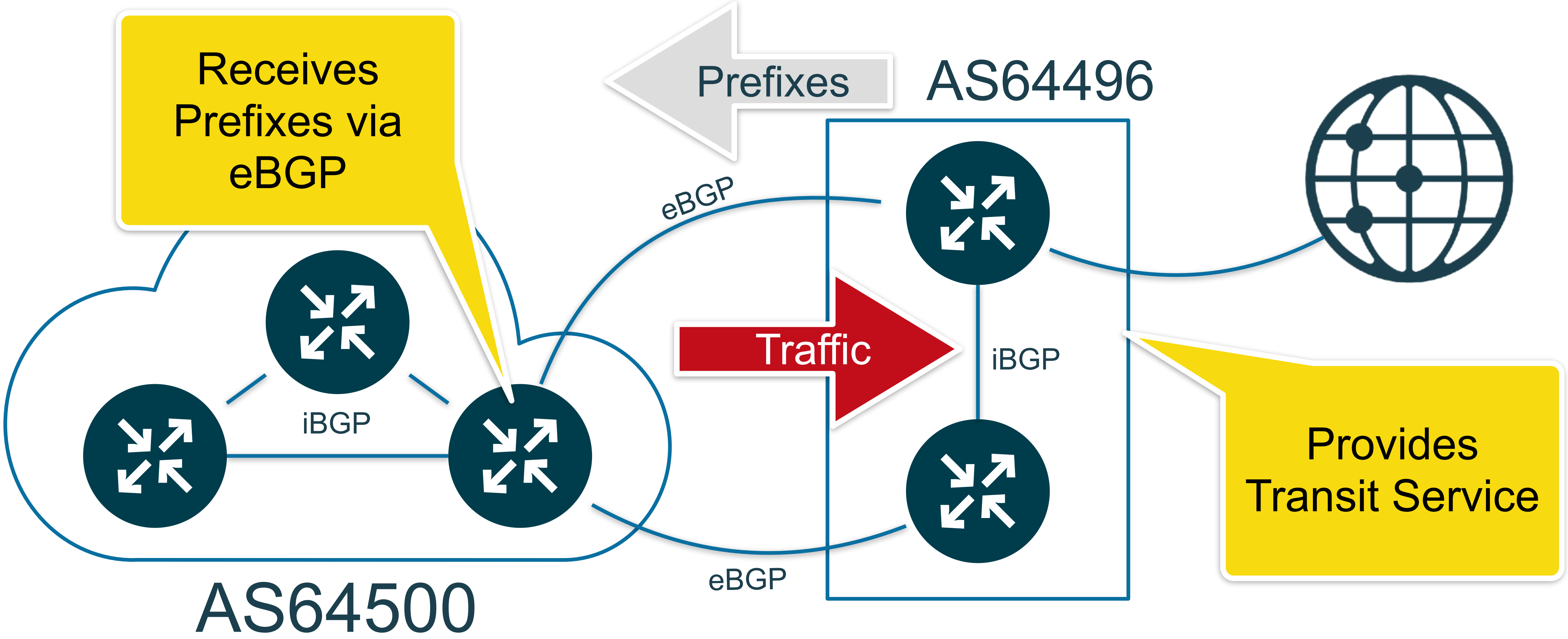
```
show ip bgp 1.0.137.0/24

Path #6: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.10 0.11
  Advertised to peers (in unique update groups):
    46.31.120.208
  9318 38040 23969
    80.81.1[illegible] from 80.81.192.157 (80.81.192.157)
      Origin incomplete, metric 5000, localpref 200,
import-candidate, import suspect
      Received Path ID 0, Local Path ID 1, version 332265
      Community: 9318:120 9318:8300 9318:8330 9318:9020
65103:276 65104:150
      Origin-AS validity: not-found
```

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4	Origin Type	IGP over EGP over Incomplete
5		
6		
7		
8		
9		
10		

Experiment: best path selection

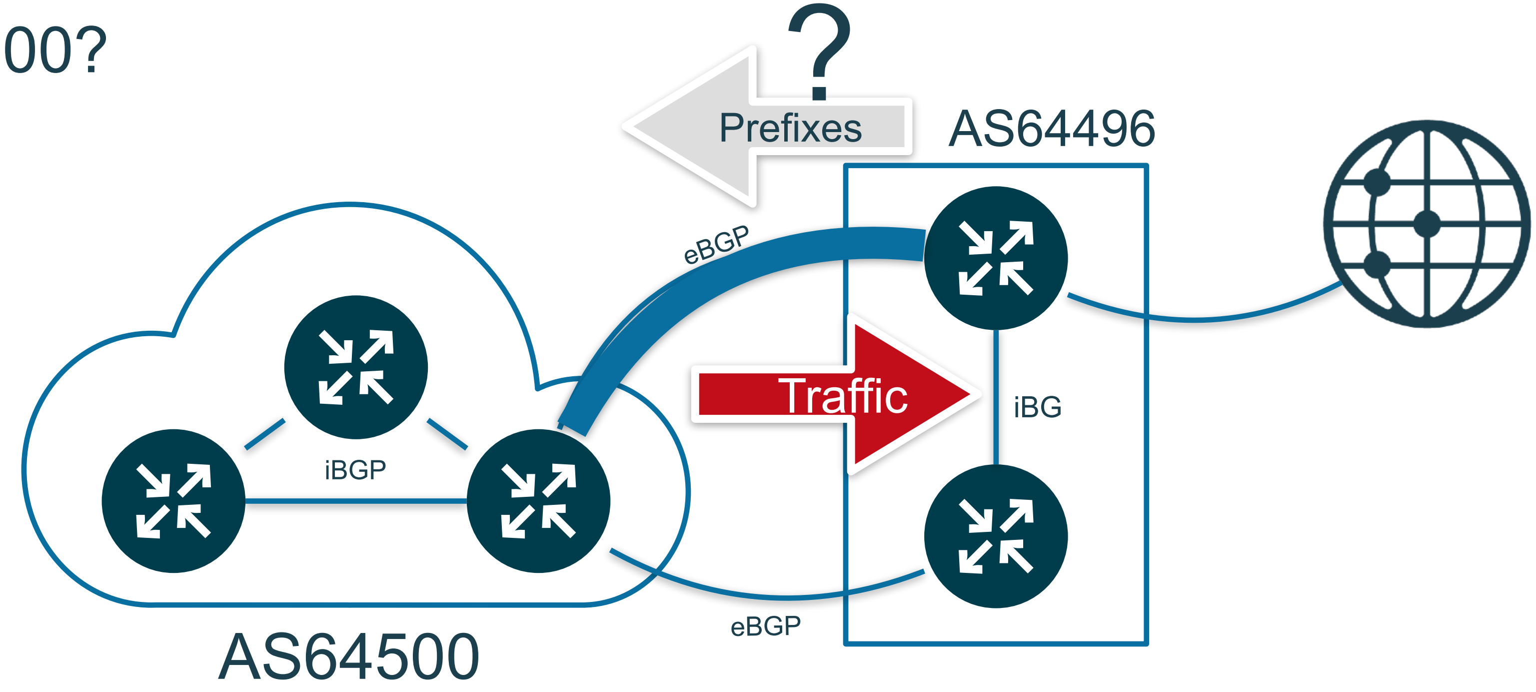
Experiment 3.03: Origin Type

Consider the following network



Consider the following network

- There are two circuits
- AS64496 wants one of them preferred
- How to tell AS64500?



BGP Route Selection Algorithm:

How to tell your neighbor where you prefer traffic?

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4	Origin Type	IGP over EGP over Incomplete
5		
6		
7		
8		
9		
10		

BGP Route Selection Algorithm: MED

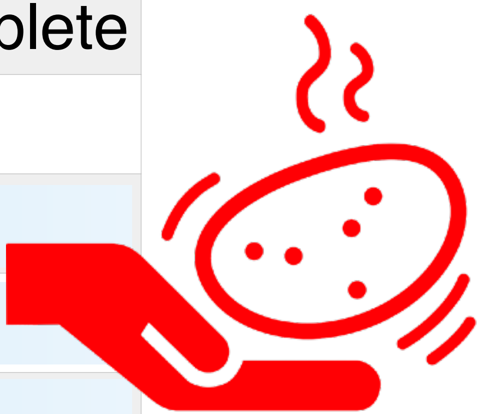
- → MED = **M**ulti-**E**xit **D**iscriminator
- → Only compared if next-hop AS is the same
- → 32bit value (0..4294967294)
- → Lower wins
- → Optional (does not have to be there)
- → A missing MED can be treated as "best" (=0, default) or "worst" (=4294967294)
- → Option "always-compare-med" **not recommended**!
- → And of course you can override whatever you receive

Experiment: best path selection

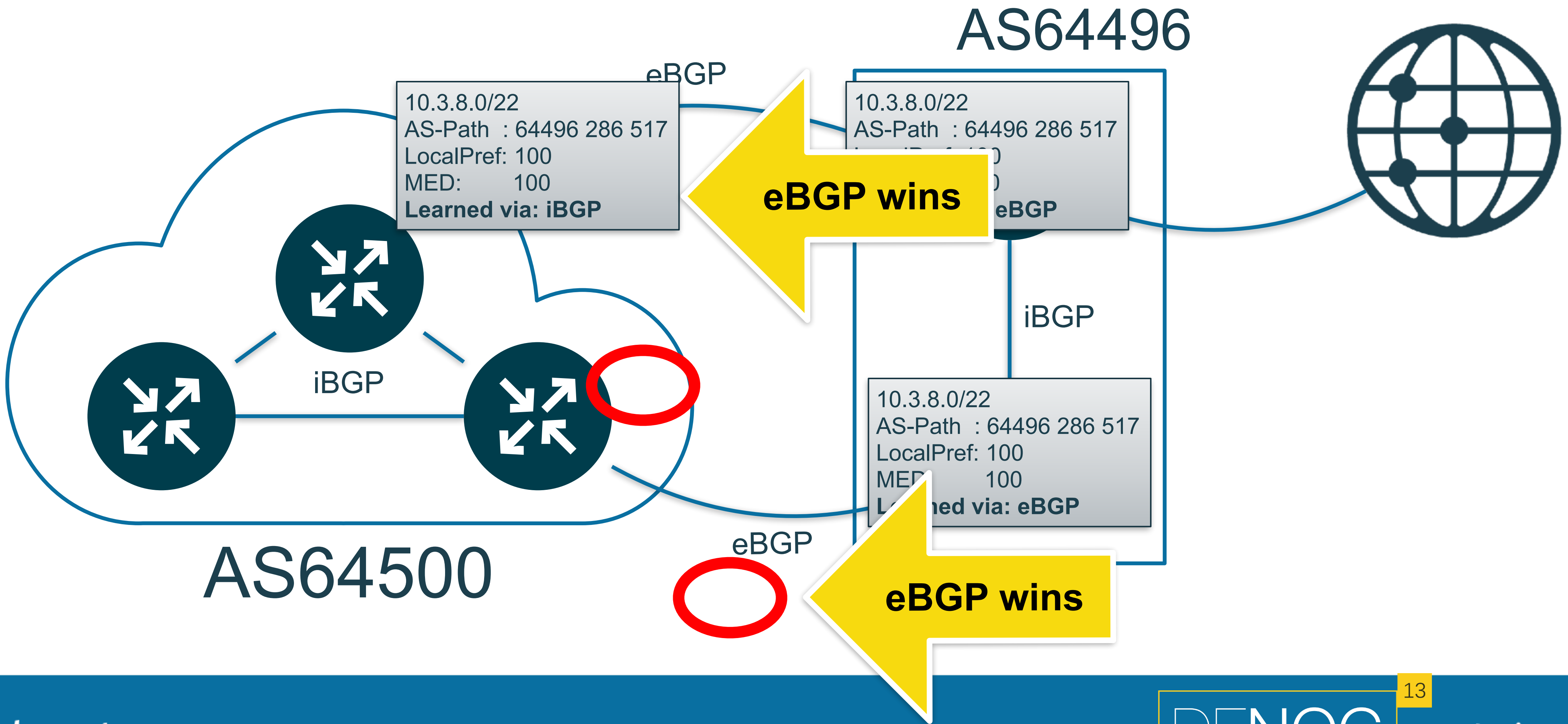
Experiment 3.04a: MED (same first AS)
Experiment 3.04b:MED (different first AS)

BGP Route Selection : Hot Potato Rules

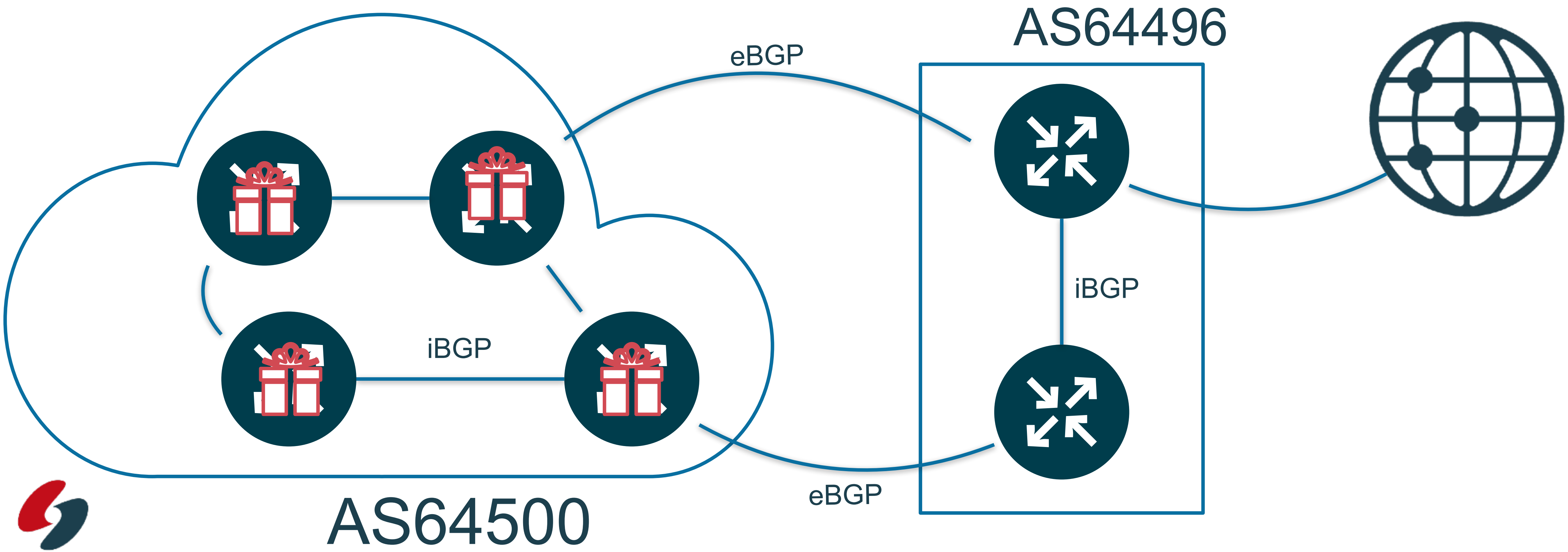
1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4	Origin Type	IGP over EGP over Incomplete
5	MED	lower wins
6		
7		
8		
9		
10		



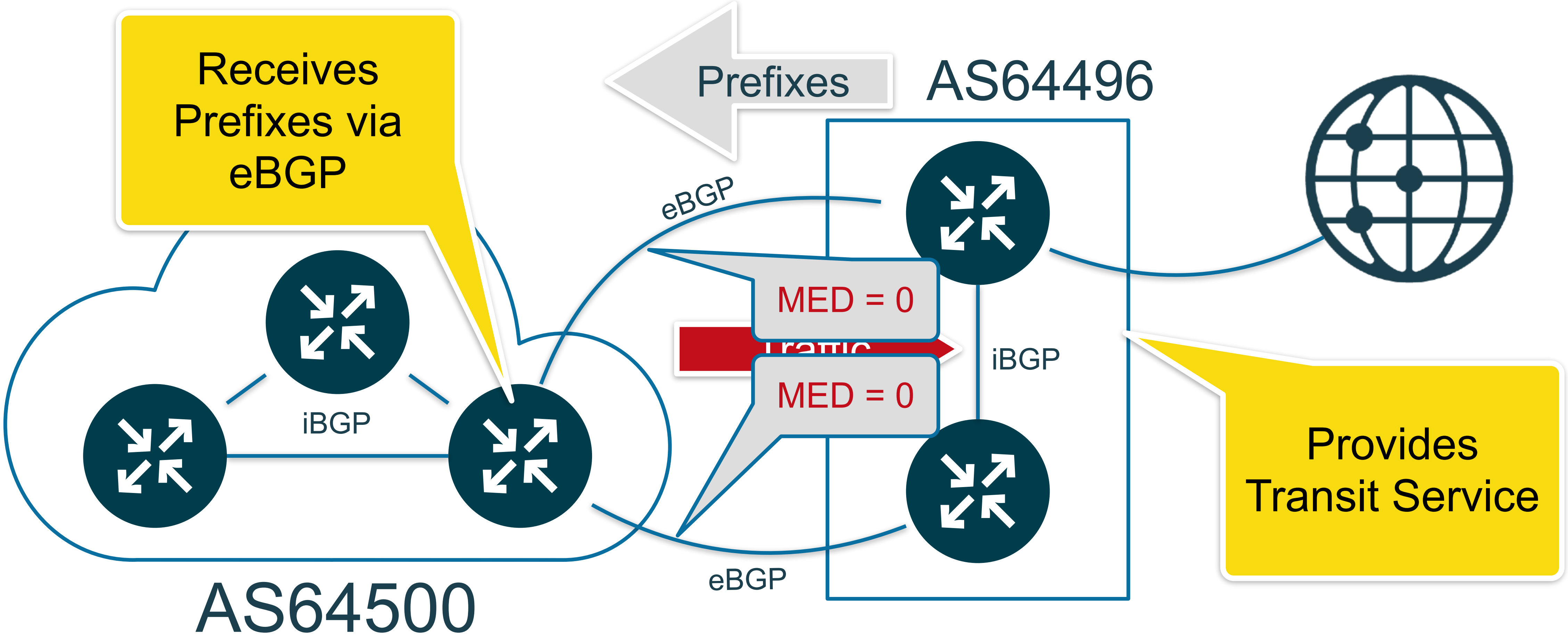
BGP Route Selection : eBGP wins



BGP Route Selection : nearest exit wins



Let's go back to our sample network

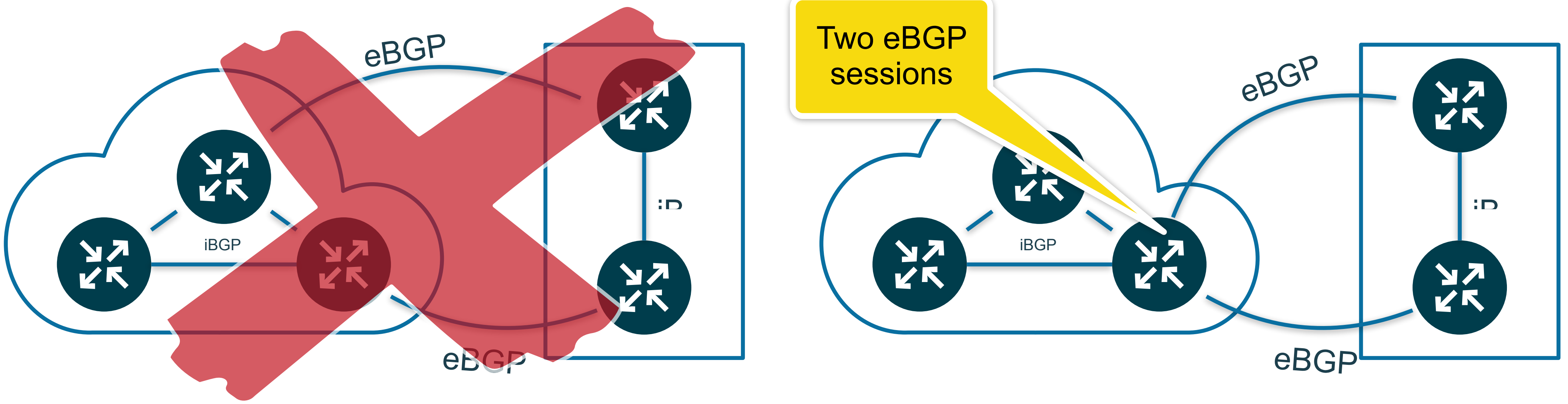


BGP Route Selection : Age / Stability

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4	Origin Type	IGP over EGP over Incomplete
5	MED	lower wins
6	eBGP, iBGP	eBGP wins
7	Exit	nearest wins
8		
9		
10		

BGP Route Selection : Age / Stability

- Exact phrasing is (Cisco):
  "When both paths are external, prefer the path that was received first"
- So this applies only if a router has two (or more) eBGP sessions
- Which happens quite often when connecting to Internet Exchanges



Experiment: best path selection

Experiment 3.05: older wins + rest

DE CIX

BGP Route Selection : Last Resort

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4	Origin Type	IGP over EGP over Incomplete
5	MED	lower wins
6	eBGP, iBGP	eBGP wins
7	Exit	nearest wins
8	Age of route	older wins
9		
10		

BGP Route Selection : Last Resort

- Router ID: lower wins
- Neighbor IP: lower wins
- Rules of last resort
- ...because at the end one and only one best path has to be selected
- Usually path selection stops before it gets to these two rules....

1		NextHop reachable?	Continue if "yes"
2		Local Preference	higher wins
3		AS Path Length	shorter wins
4		Origin Type	IGP over EGP over Incomplete
5		MED	lower wins
6		eBGP, iBGP	eBGP wins
7		Exit	nearest wins
8		Age of route	older wins
9		Router ID	lower wins
10		Neighbor IP	lower wins

BGP Route Selection : Summary

1	NextHop reachable?	Continue if "yes"
2	Local Preference	higher wins
3	AS Path Length	shorter wins
4	Origin Type	IGP over EGP over Incomplete
5	MED	lower wins
6	eBGP, iBGP	eBGP wins
7	Exit	nearest wins
8	Age of route	older wins
9	Router ID	lower wins
10	Neighbor IP	lower wins

Is that all? No!

BGP Multipath

→Some times you do not only want one best path

→But multiple "equal good" for load sharing

  →This can be enabled with the "maximum-path" config option

→To be considered as "equal good", a candidate must have the following equal to the best path:

  →local preference

  →as-path (can be relaxed)

  →origin

  →MED

  →neighbor AS (can be switched off)

1		NextHop reachable?	Continue if "yes"
2		Local Preference	higher wins
3		AS Path Length	shorter wins
4		Origin Type	IGP over EGP over Incomplete
5		MED	lower wins
6		eBGP, iBGP	eBGP wins
7		Exit	nearest wins
8		Age of route	older wins
9		Router ID	lower wins
10		Neighbor IP	lower wins

Any final questions?

?

Links and further reading

Links visited during the webinar

→ RFCs

RFCs are Internet standards issued by the Internet Engineering Task Force (IETF)

→ RFC4632: Classless Inter-domain routing (CIDR)

→ RFC4291: IPv6 addressing architecture

→ RFC827: Exterior Gateway Architecture (EGP) (historical, obsolete)

→ RFC1930: Guidelines for creation, selection, and registration of an Autonomous System (AS)

→ RFC6793: BGP Support for Four-Octet Autonomous System (AS) Number Space

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

Links visited during the webinar

- AS Numbers
  - Giving AS numbers to the RIRs: iana.org
  - Requesting an AS number, links for:
    - ARIN
    - Lacnic
    - APNIC
    - RIPE NCC
    - Afrinic
  - Private AS numbers: RFC6996, see also http://www.iana.org/assignments/as-numbers/
  - AS numbers for documentation purposes: see RFC5398

Links visited during the webinar

→ RFCs

RFCs are Internet standards issued by the Internet Engineering Task Force (IETF)

- → RFC4632: Classless Inter-domain routing (CIDR)
- → RFC4291: IPv6 addressing architecture
- → RFC827: Exterior Gateway Architecture (EGP) (historical, obsolete)
- → RFC1930: Guidelines for creation, selection, and registration of an Autonomous System (AS)
- → RFC6793: BGP Support for Four-Octet Autonomous System (AS) Number Space

→ AS Numbers

- → Giving AS numbers to the RIRs: iana.org
- → Requesting an AS number, links for:
  - → ARIN
  - → Lacnic
  - → APNIC
  - → RIPE NCC
  - → Afrinic

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

Links visited during the webinar

- RFCs

  RFCs are Internet standards issued by the Internet Engineering Task Force (IETF)

  - RFC827: Exterior Gateway Architecture (EGP) (historical, obsolete)
  - RFC1930: Guidelines for creation, selection, and registration of an Autonomous System (AS)
  - RFC4271 - A Border Gateway Protocol 4 (BGP-4)
  - RFC4632: Classless Inter-domain routing (CIDR)
  - RFC6793: BGP Support for Four-Octet Autonomous System (AS) Number Space

- AS Numbers
  - Giving AS numbers to the RIRs: iana.org
  - Requesting an AS number, links for:
    - ARIN
    - Lacnic
    - APNIC
    - RIPE NCC
    - Afrinic

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

Links visited during the webinar

→ RFCs

RFCs are Internet standards issued by the [Internet Engineering Task Force (IETF)]

- → [RFC4632]: Classless Inter-domain routing (CIDR)
- → [RFC4291]: IPv6 addressing architecture
- → [RFC827]: Exterior Gateway Architecture (EGP) (historical, obsolete)
- → [RFC1930]: Guidelines for creation, selection, and registration of an Autonomous System (AS)
- → [RFC6793]: BGP Support for Four-Octet Autonomous System (AS) Number Space

→ AS Numbers

- → Giving AS numbers to the RIRs: iana.org
- → Requesting an AS number, links for:
  - → ARIN
  - → Lacnic
  - → APNIC
  - → RIPE NCC
  - → Afrinic

DE CIX

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

Links visited during the webinar

→ RFCs

RFCs are Internet standards issued by the Internet Engineering Task Force (IETF)

- → RFC827: Exterior Gateway Architecture (EGP) (historical, obsolete)
- → RFC1930: Guidelines for creation, selection, and registration of an Autonomous System (AS)
- → RFC4271 - A Border Gateway Protocol 4 (BGP-4)
- → RFC4632: Classless Inter-domain routing (CIDR)
- → RFC6793: BGP Support for Four-Octet Autonomous System (AS) Number Space

→ AS Numbers

- → Giving AS numbers to the RIRs: iana.org
- → Requesting an AS number, links for:
  - → ARIN
  - → Lacnic
  - → APNIC
  - → RIPE NCC
  - → Afrinic

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

Links visited during the webinar

- RFCs
  RFCs are Internet standards issued by the Internet Engineering Task Force (IETF)
  - RFC4271 - A Border Gateway Protocol 4 (BGP-4)
    - see 5.1.5 for a definition of *Local Preference*
    - see 9.1 for the BGP best path selection algorithm

- BGP Best Path Selection by vendor
  - Cisco
  - Juniper
  - Mikrotik
  - Nokia
  - BIRD
  - FRRouting

- If you peering at any Internet Exchange - please use PeeringDB

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

Links and further reading

- Definition of terms (all from RFC4271):
  - *Next Hop* is defined in Section 5.1.3
  - *AS Path* is defined in Section 5.1.2
  - *Local Preference:* Section 5.1.5
  - Origin: Section 5.1.1
  - *Multi Exit Discriminator (MED):* Section 5.1.4
- Best Path Selection process: Section 9.1
- BGP Route Selection Algorithm by vendor:
  - Cisco
  - Juniper
  - Mikrotik
  - Nokia
  - BIRD
  - Quagga

BGP Best Path Selection Algorithm

Bold items were covered in this webinar

1		NextHop reachable?	Continue if "yes"
2		Local Preference	higher wins
3		AS Path	shorter wins
4		**Origin Type**	**IGP over EGP over Incomplete**
5		**MED**	**lower wins**
6		**eBGP, iBGP**	**eBGP wins**
7		**Exit**	**nearest wins**
8		**Age of route**	**older wins**
9		**Router ID**	**lower wins**
10		**Neighbor IP**	**lower wins**

Can be switched off

Can be set/changed receiving

BGP Best Path Selection Algorithm

Local Preference is...

→ a 32bit integer value (0-4294967295)
→ Propagated via iBGP inside an Autonomous System
→ Usually set using rules when receiving prefixes
  - According to your routing policy
→ Typical values
  - 10000 (high value) for customer prefixes
  - 1000 (medium value) for prefixes received via peering
  - 100 (low value) for prefixes received via upstream
→ Rules to adjust local preference can be as complex as your router software allows it to be.

AS Path is...

→ an ordered list of AS numbers...
→ ...with the originator AS at the rightmost side
→ automatically built when prefixes are sent via eBGP
→ length of the path is used for selection (shorter wins)

BGP Best Path Selection Algorithm

Origin Type is...

→ a historic, but mandatory attribute

→ set by originator AS and forwarded unchanged

→ can have the values (in order of preference):

- IGP - prefix was originated via a network statement
- EGP - prefix was learned from Exterior Gateway Protocol (RFC904, historic)
- incomplete - prefix was learned by another protocol

Multi Exit Discriminator (MED) is...

→ a 32Bit value, lower wins

→ optional, if it is not there it's either treated as zero (best) or as 2^32-1 (worst)

→ non-transitive (set by an eBGP speaker and only sent to the next-hop AS)

→ usually set using rules when sending prefixes (according to the sender's routing policy)

→ only compared between eBGP speakers if next-hop AS is the same

Router ID is...

→ also called **BGP Identifier**

→ a 4 byte, unsigned integer (mostly it's the IPv4 loopback address of a router)

→ unique within one AS

→ set at startup and stays unchanged

→ the same for all BGP sessions

Neighbor IP is...

→ the last tie-breaker in the BGP Best Path Selection

→ the IP address of the eBGP speaker a prefix was learned from