



**RIPE NCC**  
RIPE NETWORK COORDINATION CENTRE

# Advanced IPv6

Training Course

RIPE NCC Learning & Development



# Schedule

09:00 - 09:30	<b>Coffee, Tea</b>
11:00 - 11:15	<b>Break</b>
13:00 - 14:00	<b>Lunch</b>
15:30 - 15:45	<b>Break</b>
17:30	<b>End</b>



# RIPE NCC Training Material

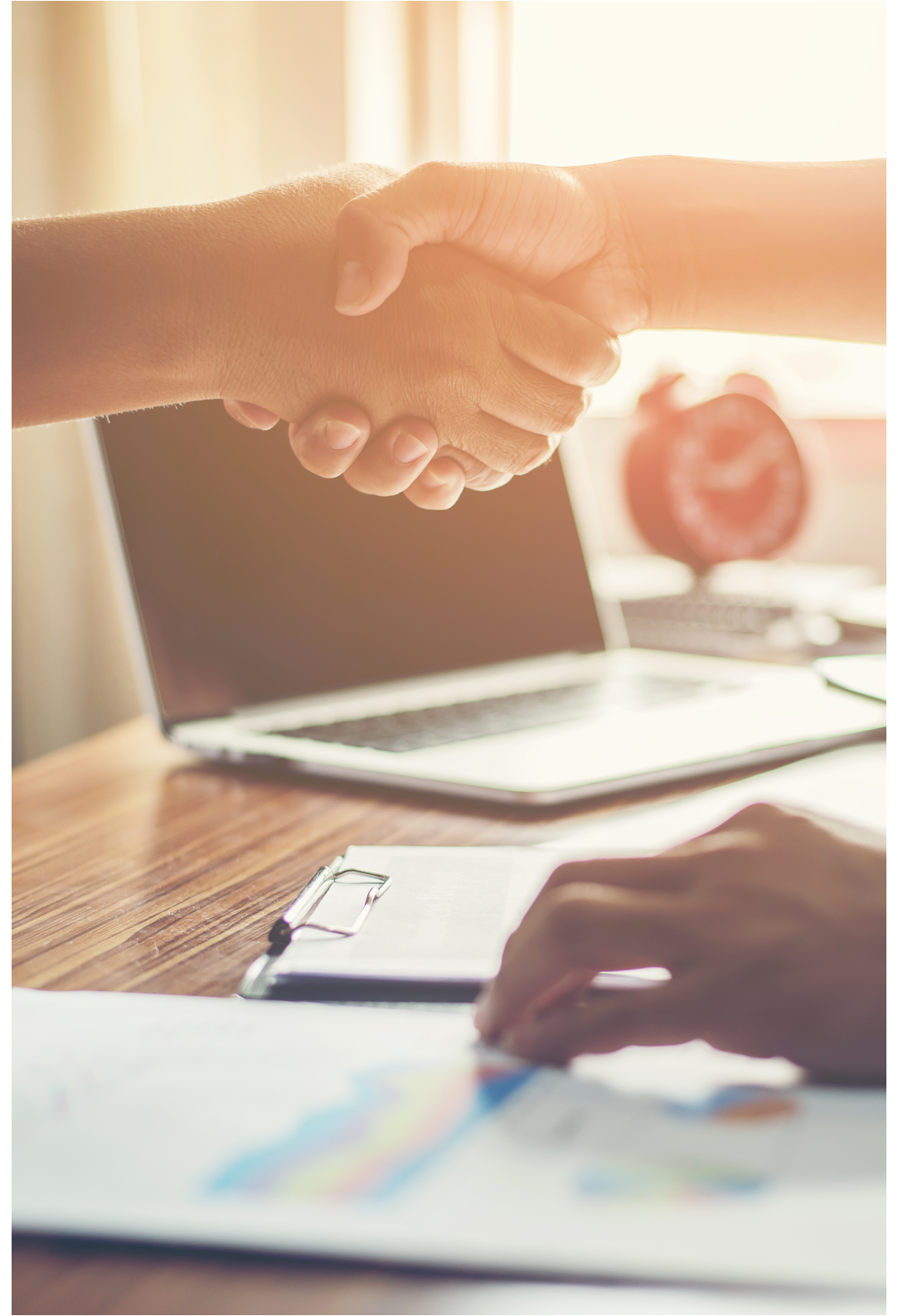
Please find your training material at the following link

<https://www.ripe.net/training-material>



# Introductions

- Name
- Number on the list
- Experience with:
  - IPv6
  - Cisco
  - OSPF
  - BGP
- Goals





# Course Overview

1 - Section



# Overview Day 1

- Introduction
- IPv6 Routing Basics
- Exercise: Enable IPv6
- OSPFv3
- Exercise: Configuring OSPFv3
- BGP
- Exercise: Configuring BGP
- Content
- Mobile Providers



# Overview Day 2

- Transition Mechanisms
- Exercise: NAT64/DNS64
- Host Configuration
- Exercise: SLAAC
- DHCPv6
- Exercise: DHCPv6
- Security
- IP Address Management



# IPv6 Routing Basics

2 - Section





# IPv6 Routing Basics

- IPv6 routing is the same as IPv4 routing
  - Longest matching prefix
  - Same structure and concepts
  - Some technical differences



# Longest Matching Prefix

Example routing table:

Route	Next-Hop
::/0	2001:db8:aaa:bbb::cdef:1
2001:db8::/32	2001:db8:bcd:aaa::1
2001:db8::/48	2001:db8:cde:bbb::1
2001:db8:1000::/36	2001:db8:fff:eee::1
2001:db8:1000::/48	2001:db8:def:bbb::1
2001:db8:2000::/48	2001:db8:def:bbb::2



# Longest Matching Prefix

Matches for a packet with destination:

2001:db8:2000:1a2b:02ab:9eff:fe01:f5b1

Route	Next-Hop
::/0	2001:db8:aaa:bbb::cdef:1
2001:db8::/32	2001:db8:bcd:aaa::1
2001:db8::/48	2001:db8:cde:bbb::1
2001:db8:1000::/36	2001:db8:fff:eee::1
2001:db8:1000::/48	2001:db8:def:bbb::1
2001:db8:2000::/48	2001:db8:def:bbb::2



# Longest Matching Prefix

Matches for a packet with destination:

2001:db8:2000:1a2b:02ab:9eff:fe01:f5b1

Route	Next-Hop
::/0	2001:db8:aaa:bbb::cdef:1
2001:db8::/32	2001:db8:bcd:aaa::1
2001:db8::/48	2001:db8:cde:bbb::1
2001:db8:1000::/36	2001:db8:fff:eee::1
2001:db8:1000::/48	2001:db8:def:bbb::1
<b>2001:db8:2000::/48</b>	<b>2001:db8:def:bbb::2</b>



# Longest Matching Prefix

Matches for a packet with destination:

2001:db8:2001:1a2b:02ab:9eff:fe01:f8b2

Route	Next-Hop
::/0	2001:db8:aaa:bbb::cdef:1
2001:db8::/32	2001:db8:bcd:aaa::1
2001:db8::/48	2001:db8:cde:bbb::1
2001:db8:1000::/36	2001:db8:fff:eee::1
2001:db8:1000::/48	2001:db8:def:bbb::1
2001:db8:2000::/48	2001:db8:def:bbb::2



# Longest Matching Prefix

Matches for a packet with destination:

2001:db8:2001:1a2b:02ab:9eff:fe01:f8b2

Route	Next-Hop
::/0	2001:db8:aaa:bbb::cdef:1
2001:db8::/32	2001:db8:bcd:aaa::1
2001:db8::/48	2001:db8:cde:bbb::1
2001:db8:1000::/36	2001:db8:fff:eee::1
2001:db8:1000::/48	2001:db8:def:bbb::1
2001:db8:2000::/48	2001:db8:def:bbb::2



# Longest Matching Prefix

Matches for a packet with destination:

2001:db8:1001:1a2b:02ab:92ff:fe01:f8b2

Route	Next-Hop
::/0	2001:db8:aaa:bbb::cdef:1
2001:db8::/32	2001:db8:bcd:aaa::1
2001:db8::/48	2001:db8:cde:bbb::1
2001:db8:1000::/36	2001:db8:fff:eee::1
2001:db8:1000::/48	2001:db8:def:bbb::1
2001:db8:2000::/48	2001:db8:def:bbb::2



# Longest Matching Prefix

Matches for a packet with destination:

2001:db8:1001:1a2b:02ab:92ff:fe01:f8b2

Route	Next-Hop
::/0	2001:db8:aaa:bbb::cdef:1
2001:db8::/32	2001:db8:bcd:aaa::1
2001:db8::/48	2001:db8:cde:bbb::1
<b>2001:db8:1000::/36</b>	<b>2001:db8:fff:eee::1</b>
2001:db8:1000::/48	2001:db8:def:bbb::1
2001:db8:2000::/48	2001:db8:def:bbb::2





# Summary

- IPv6 routing uses the same structure as IPv4 routing
  - Addresses are longer
  - Prefixes are longer



# **Add IPv6 to Loopback & Links**

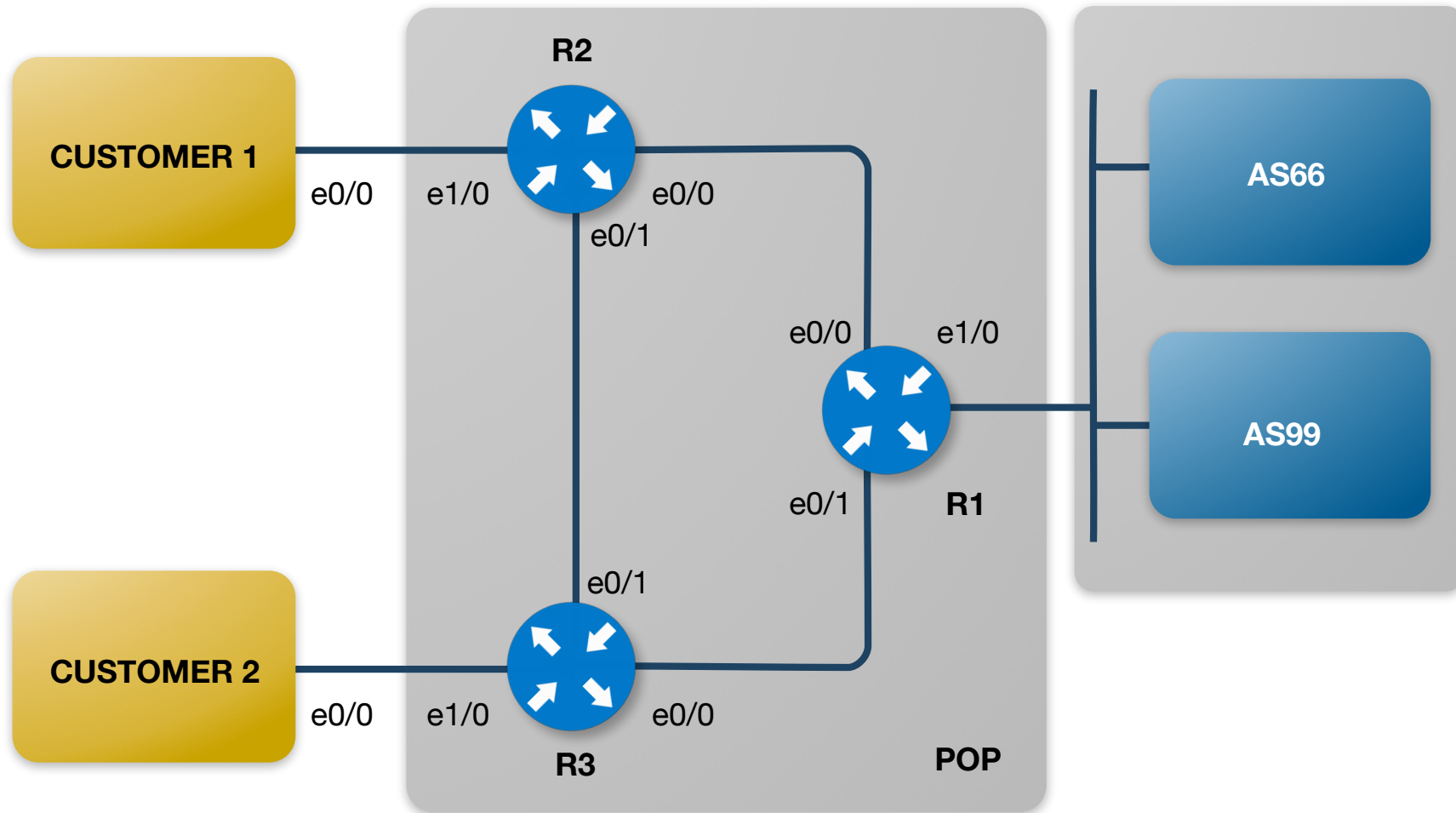
3 - Exercise



# Discover the IPv4 Network

- Make sure you have connectivity
- Go to: **workbench.ripe.net**
- Choose LAB1 or LAB2 (trainer will tell you)
- Your login is your number on the attendee list
- We will provide you with the password
  
- Read instructions carefully
- First discover, then configure

# Network Diagram





# Discover the IPv4 Network

- Routing Protocol
  - IGP (OSPF) is used for loopback addresses and point-to-point links
  - EGP (BGP) is used for the edge core routers
- R1 announces a default route via OSPF
- Keeps routing tables in the area smaller
  - All inter-area traffic must pass R1



# Adding IPv6 to the Network

- We will now add IPv6 to our existing network
- We will not change the network structure
- **We will not change IPv4 addresses**
  
- First step: Addressing Structure
- Find the addresses on the handout



# Addressing with IPv6

Where **X** is your number on the attendee list!

- Loopbacks:
  - There is a /32 (IPv4): 172.**X**.255.**Y**(router number)
  - Use a /128 (IPv6) 2001:ff**XX**::**Y**(router number)/128
- Point-to-point core:
  - There is a /30 (IPv4) from 10.**X**.0.0/24
  - Use a /127 from 2001:ff**XX**::/60 for core links
  - Use a /64 from 2001:ff**XX**::/60 for the customer links



# Interface IPv6 Settings Routers

- Disable Router Advertisements
  - On point-to-point interfaces
  - On LANs where unprepared devices are connected
- You'll configure "router advertisement suppression" on the appropriate interfaces.
- Otherwise they will suddenly be globally reachable over IPv6 without being configured, prepared and/or protected





# Basic IPv6 Settings

- Accessing the configuration mode:

```
configure terminal
```

- Before configuring IPv6 on your router interfaces, the basic IPv6 settings need to be set up on the router
- For R1, R2, R3 and the Customers

```
ipv6 unicast-routing  
ipv6 cef
```



# Interface IPv6 Settings Routers

- Use the information in the handouts
- Give the correct IPv6 addresses to the interfaces
- Start with the loopback interface
- Then configure the point-to-point links
- Fill in the appropriate interface name, IPv6 address and prefix length

```
interface xyz
  ipv6 address ...
  no ipv6 redirects
  ipv6 nd ra suppress all
```

# Interface IPv6 Settings Customers



- Use the information in the handouts
- Give the correct IPv6 addresses to the interfaces
- We don't configure loopback interfaces
- Then configure the point-to-point link
- Fill in the appropriate interface name, IPv6 address and prefix length (/64)
- We don't disable router advertisements

```
interface xyz
  ipv6 address ...
  no ipv6 redirects
```

# Interface IPv6 Settings Customers



- We will set a default route for the customers
- This is a manual configuration
- This is not needed if you use SLAAC

```
ipv6 route ::/0 2001:ffXX:0:ff01::b (customer 1)
```

```
ipv6 route ::/0 2001:ffXX:0:ff02::b (customer 2)
```

- **XX** is your number on the list



# Checking Your Configuration

- Check your own configuration
  - Can you ping your own IPv6 loopback address?
  - Can you ping your own side of the point-to-point link?



# Questions





# OSPFv3

4 - Section



# OSPF Characteristics

- OSPF = Open Shortest Path First
- Link State Protocol
- OSPFv3 is an implementation of OSPF for IPv6
- OSPFv2 (for IPv4) and OSPFv3 run independently on the router
- Most OSPFv3 functions are the same as OSPFv2

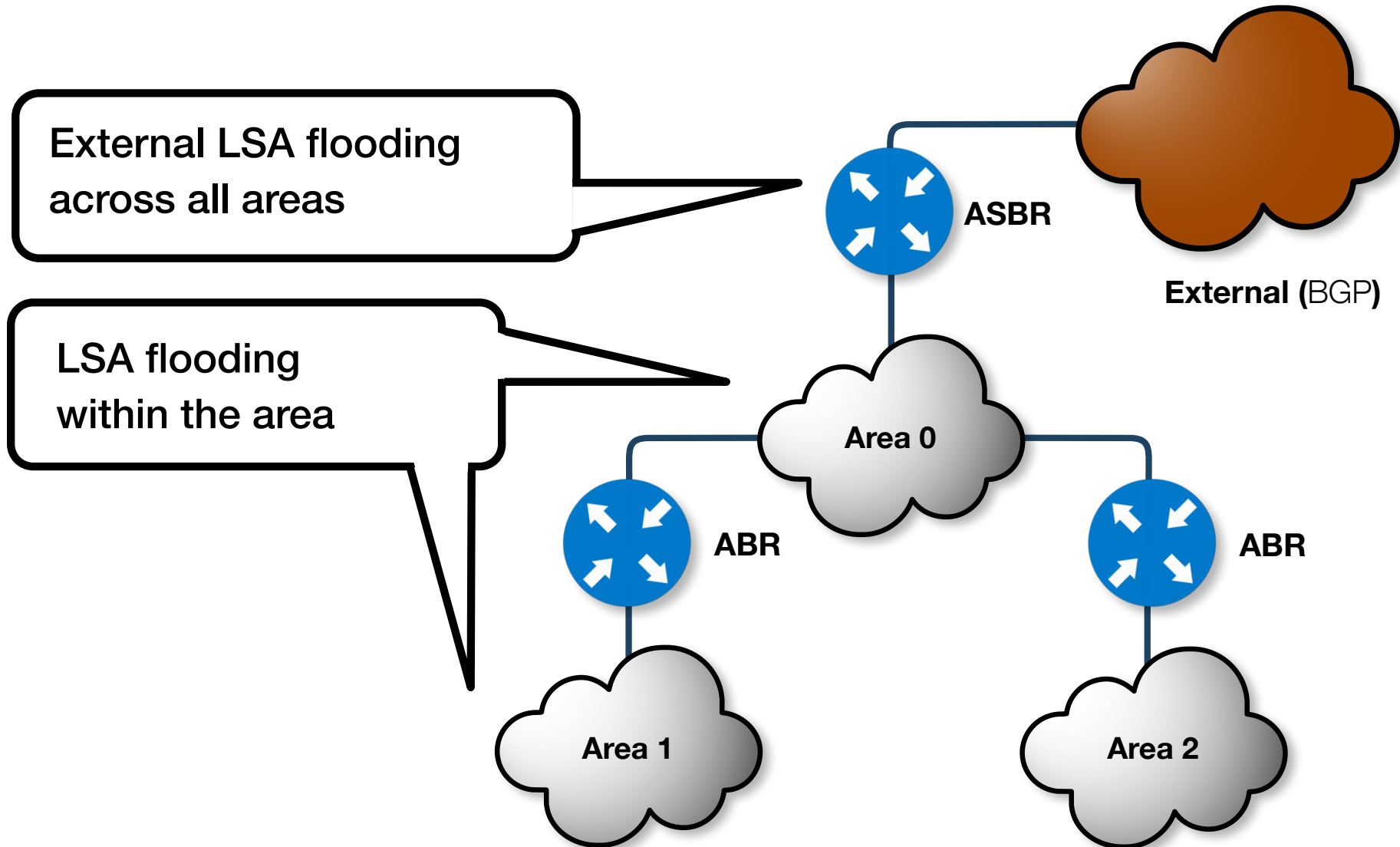




# OSPF Refresher

- Link state protocol
  - Every router has full insight into network topology of the area
  - Routes are sent to other routers using Link State Advertisements (LSAs)
- Role of Area Border Routers:
  - Limit the flooding of LSAs to isolate topology changes within the area
- DR/BDR selection
  - FF02::5 - all OSPF routers
  - FF02::6 - DR/BDR Routers

# OSPF Refresher





# OSPF for IPv6

- Multiple instances of OSPFv3 can be run on a link
  - Unlike in OSPFv2
- OSPFv3 still uses 32-bit numbers as a router ID
  - If no IPv4 address is configured on any interface, the router ID command is required to set the 32-bit router ID (for IPv6-only networks)
- **We have the separation of the physical topology and the IPv6 address information**
  - New LSA Types - LSA Type 8 & 9



# OSPF for IPv6

- Router ID is a unique identifier for the router
  - Must be configured in the routing process
  - Is still a 32-bit number, written in 4 octets
  - It is used to sign routing updates
  
- But to make your life easy, you can use an IPv4 loopback address



# OSPF for IPv6

- OSPF for IPv4 (OSPFv2) can be configured:
  - on each subnet or,
  - on each link
- OSPF for IPv6 (OSPFv3) can be configured:
  - on each link
- Interface mode configuration will automatically activate the OSPF process on your running config



# OSPF for IPv6

- LSA types and functions in OSPF are almost the same as for OSPFv2
  - There are new LSAs introduced: LSA type 8 & 9 for IPv6
- OSPFv3 uses multicast addresses:
  - ff02::5 for All OSPFv3 Routers
  - ff02::6 for All OSPFv3 Designated Routers
- All OSPFv3 adjacencies are formed using link-local addresses
  - From fe80::/10 IPv6's link-local address scope



# Configuration of OSPF as IGP

- Example of OSPF for IPv4 per-subnet configuration

```
router ospf 1
  log-adjacency-changes
  passive-interface default
  network 172.16.1.1 0.0.0.0 area 1
  no passive-interface e0/0
  network 172.16.11.8 0.0.0.3 area 1
  no passive-interface e0/1
  network 172.16.11.0 0.0.0.3 area 1
```



# Configuration of OSPF as IGP

- Example of OSPF for IPv4 per-link configuration

```
router ospf 1
  log-adjacency-changes
  passive-interface e1/1
  passive-interface e1/0
  !
interface loopback 0
  ip ospf 1 area 1
  !
interface e0/0
  ip ospf 1 area 1
  !
interface e0/1
  ip ospf 1 area 1
  !
```



# Configuration of OSPF as IGP



- Example of OSPF for IPv6 per-link configuration

```
ipv6 router ospf 1
  log-adjacency-changes
  passive-interface e1/1
  passive-interface e1/0
  !
interface loopback 0
  ipv6 ospf 1 area 1
  !
interface e0/0
  ipv6 ospf 1 area 1
  !
interface e0/1
  ipv6 ospf 1 area 1
  !
```

# OSPFv3 Question

Which of the options below is NOT a difference between OSPFv2 and OSPFv3?

- a. Number of LSA types used
- b. The length of the Router-ID
- c. The separation of the physical topology and the IPv6 addresses
- d. Multicast addresses that are being used



# OSPFv3 Answer

Which of the options below is NOT a difference between OSPFv2 and OSPFv3?

- a. Number of LSA types used
- ✓ **The length of the Router-ID**
- c. The separation of the physical topology and the IPv6 addresses
- d. Multicast addresses that are being used





# Configuring OSPFv3

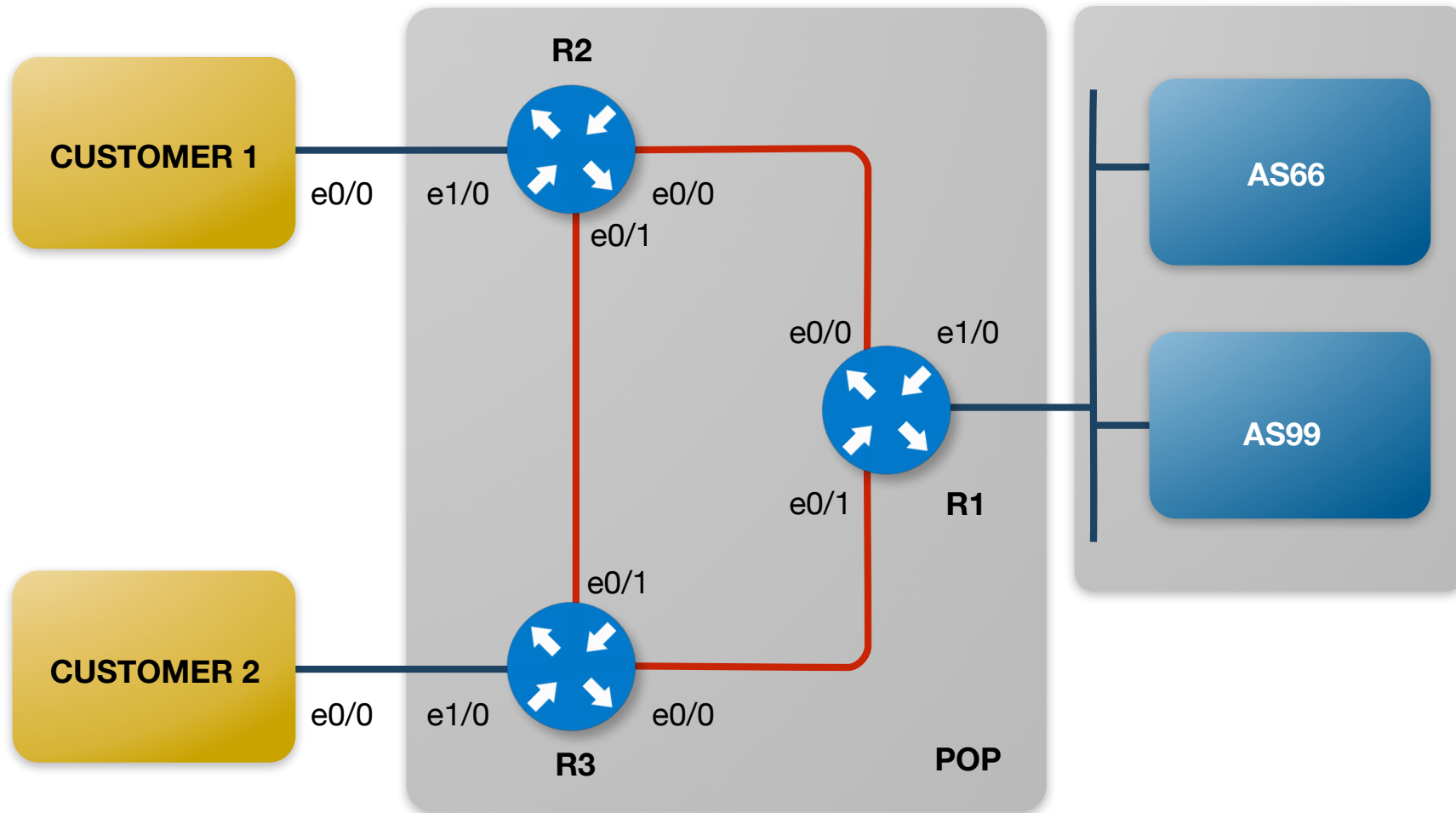
5 - Exercise



# Overview of IGP Configuration

- You have to configure OSPFv3 as IGP for IPv6
- Dual Stack will be used to ensure both IPv4 and IPv6 operation
- OSPFv2 is already set up

# Network Diagram





# Have a good look...

- At the IPv4 configuration....

```
show running-config | s router ospf
```



# OSPFv3 Global Settings

- Tell the router to do OSPFv3 and the process-id
- Log adjacency changes
- Set a router ID
- Define passive interface

```
ipv6 router ospf 1
  log-adjacency-changes
  router-id 172.X.255.Y    (Y is router number)
  passive-interface Ethernet1/0
  passive-interface Ethernet1/1
  redistribute connected
```

- On router 1 also add:

```
ipv6 router ospf 1
  default-information originate always
```





# OSPFv3 Interface Settings

- OSPFv3 interface settings

```
interface xyz
  ipv6 ospf network point-to-point
  ipv6 ospf 1 area 0
```

- Fill in the appropriate interface names and OSPF area



# Checking Your Configuration

- Check your own configuration
  - Can you ping the loopback on R3 from C1?
  - Can you ping the loopback on R2 from C2?



# Checking Your Configuration

- You should now have a running IPv6 core network!
- For every internal IPv4 route there should be a corresponding IPv6 route
- Try to ping and traceroute point-to-point connections and loopback addresses in your part of the network



# BGP

6 - Section



# BGP Overview

- Routing Protocol used to exchange routing information between networks
  - Exterior Gateway Protocol
- It is based on Path Vector Protocol
  - Similar to Distance Vector
- Each border router sends to its neighbors the full route to one destination, not just the distance



# Autonomous System

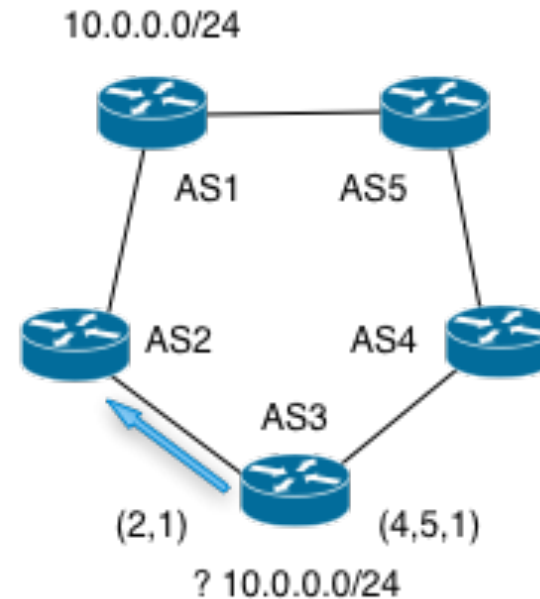
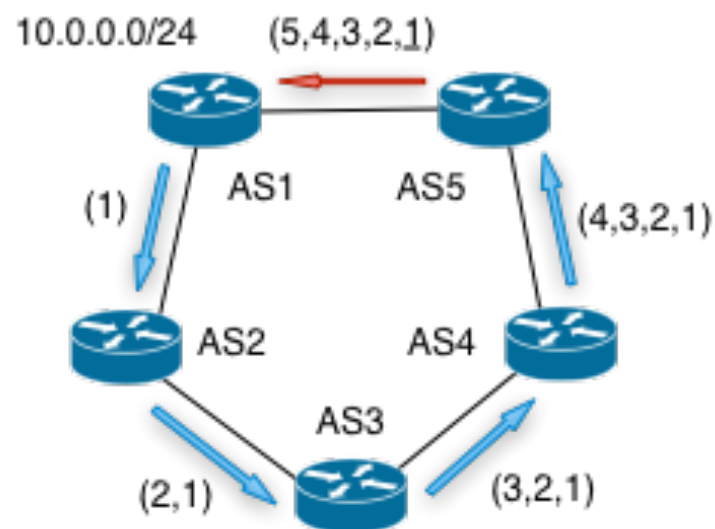
- Collection of networks with the same routing policy
- Usually under single ownership and administrative control
  - Single routing policy
- Identified by 16 or 32 bit numbers
  - 16bit: 0 - 65,535
  - 32bit: 65,536 - 4,294,967,295





# AS Path

- Sequence of ASes a route has traversed
  - Loop detection
  - Path selection ( AS-PATH length )





# BGP Modes

- **eBGP**: Between BGP speakers in a different AS



- **iBGP**: Between BGP speakers within the same AS







# BGP Messages

- OPEN
  - opens the tcp session
- KEEPALIVE
  - keeps the session running
- NOTIFICATION
  - error handling
- UPDATE
  - actual route updates (NLRI, AS-path, AS-path attributes)



# NLRI

- Network Layer Reachability Information
  - Used to advertise feasible routes
- NLRI Composed of:
  - Network Prefix
  - Mask Length



# BGP Path Attributes

- Well known
  - They are known by all the routers and passed to BGP neighbors
  - Mandatory and are included in the UPDATE messages
- Optional
  - May not be supported by all BGP implementations
  - The transitive bit determines if an optional attribute is passed to BGP neighbors

# BGP Question

What allows BGP to support IPv6 (and the other address families)?

- a. MD5 Hashing
- b. IPSec
- c. Multiprotocol Extensions
- d. DS-LITE



# BGP Answer

What allows BGP to support IPv6 (and the other address families)?

- a. MD5 Hashing
- b. IPSec
- Multiprotocol Extensions**
- d. DS-LITE





# Multiprotocol BGP (MP-BGP)

- Extension to the BGP protocol
- Carries routing information about other protocols:
  - Multicast
  - MPLS VPN
  - IPv6
- Multi-Protocol NLRI exchange is negotiated at session set up (OPEN Message)



# MP-BGP

- New features in OPEN Message:
  - **BGP Capabilities Advertisement:**
    - Address Family Identifier (**AFI**)
    - Subsequent Address Family Identifier (**SAFI**)
    - Multiprotocol Reachable Network Layer Reachability Information



# AFI / SAFI

- Address Family Identifier (AFI)
  - Identifies Address Type
  - AFI = 1 (IPv4)
  - AFI = 2 (IPv6)
- Subsequent Address Family Identifier (SAFI)
  - Sub category for AFI Field
  - Address Family Identifier (AFI)
    - Sub-AFI = 1 (NLRI is used for unicast)
    - Sub-AFI = 2 (NLRI is used for multicast RPF check)
    - Sub-AFI = 3 (NLRI is used for both unicast and multicast RPF check)
    - Sub-AFI = 4 (label)
    - Sub-AFI = 128 (VPN)





# MP-BGP Capabilities Negotiation

- BGP routers establish peering sessions through the OPEN message
- OPEN message contains optional parameters
- BGP session is terminated if OPEN parameters are not recognised
- A new optional parameter: CAPABILITIES containing its capabilities:
  - Multiprotocol extension (AFI/SAFI)
  - Route Refresh
  - Outbound Route Filtering



# Managing Multiple Protocols

- Independent operation
  - One RIB per protocol
  - Distinct policies per protocol (IP address specific route maps and prefix lists must be adjusted)
  - Make separate route maps for IPv4 and IPv6
  - Prefix lists are always separate
  - It is common to use a **\_v4** and a **\_v6** suffix to names



# Configuring MP-BGP & Customers

7 - Exercise



# eBGP

## 7.1 - Exercise

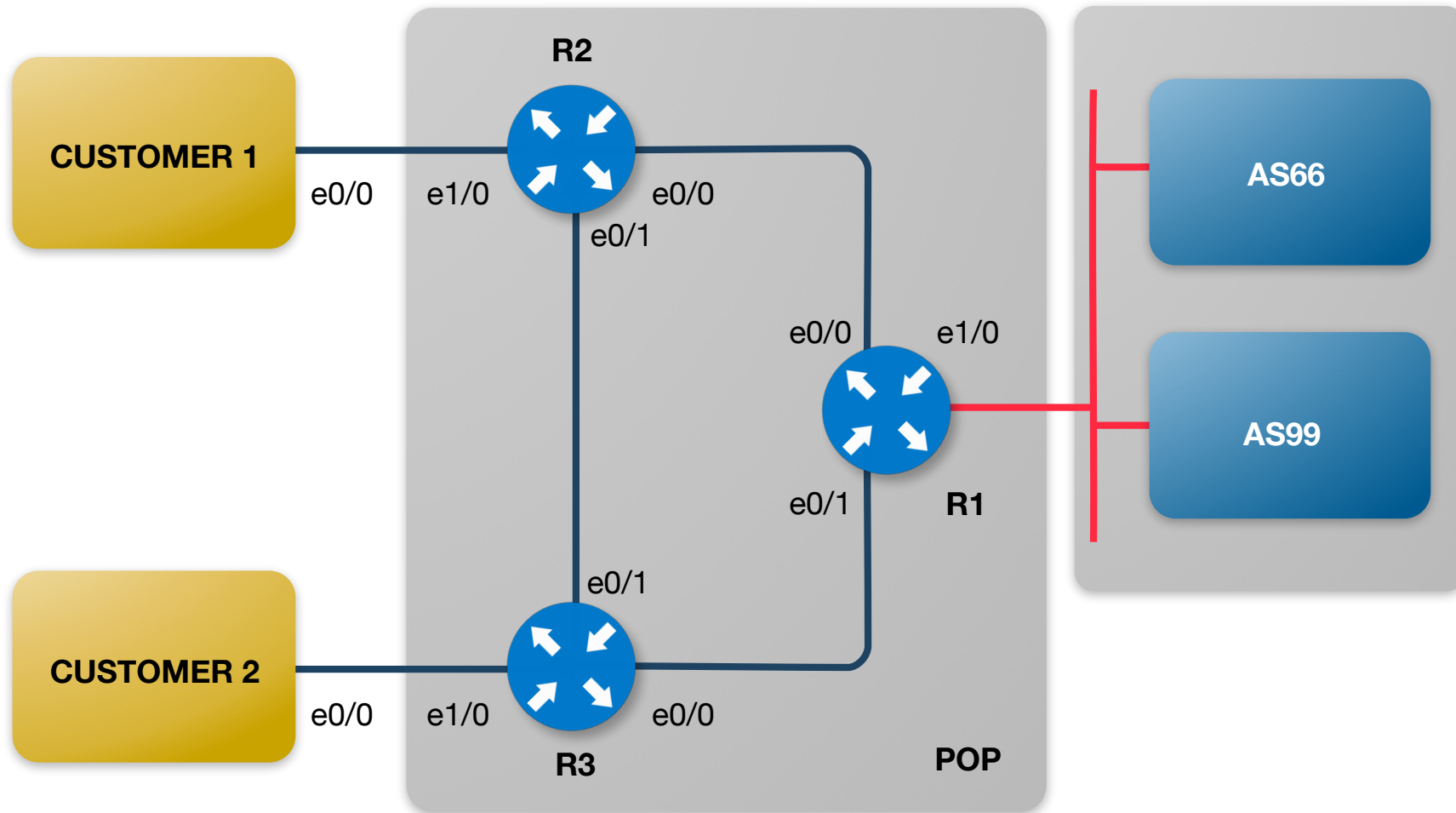


# BGP Configuration R1

- Cisco defaults to address-family IPv4
- This must be disabled before configuring IPv6
- Your AS Number is 1 + your number on the participants list (e.g. 109)

```
router bgp 1XX  
no bgp default ipv4-unicast
```

# Network Diagram



# Set the Route and Prefix list on R1



```
address-family ipv6
  network 2001:ffXX::/32
(exit) X 2
ipv6 route 2001:ffXX::/32 Null0
ipv6 prefix-list filter_v6 seq 5 permit 2001:ffXX::/32
```



# BGP Configuration R1

- Now we are going to set up BGP to our upstreams
- We use the same settings for IPv6 as we have for IPv4
- Only configure R1

```
router bgp 1XX  
neighbor 2001:ff69::66 remote-as 66  
neighbor 2001:ff69::99 remote-as 99
```



# BGP Configuration R1



- And activate the external session in the correct address family

```
address-family ipv6
  redistribute static
  neighbor 2001:ff69::66 activate
  neighbor 2001:ff69::99 activate
  neighbor 2001:ff69::66 prefix-list filter_v6 out
  neighbor 2001:ff69::99 prefix-list filter_v6 out
```



# Filtering

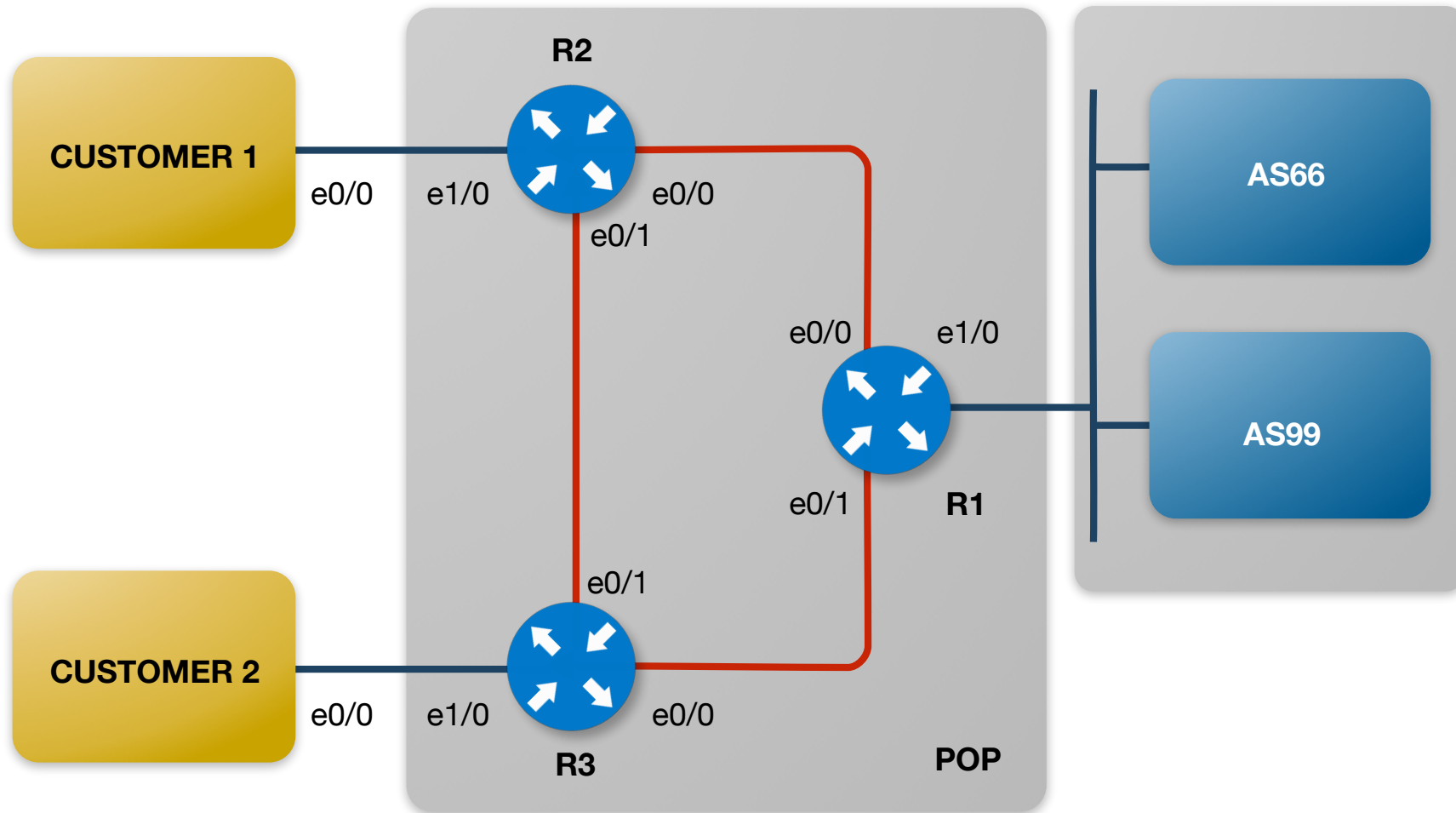
- We filter the routes we announce
  - Why?
  - Why in this way?
  - What are the differences in IPv4 and IPv6 ?



**iBGP**

7.2 - Exercise

# Network Diagram





# BGP Configuration R1

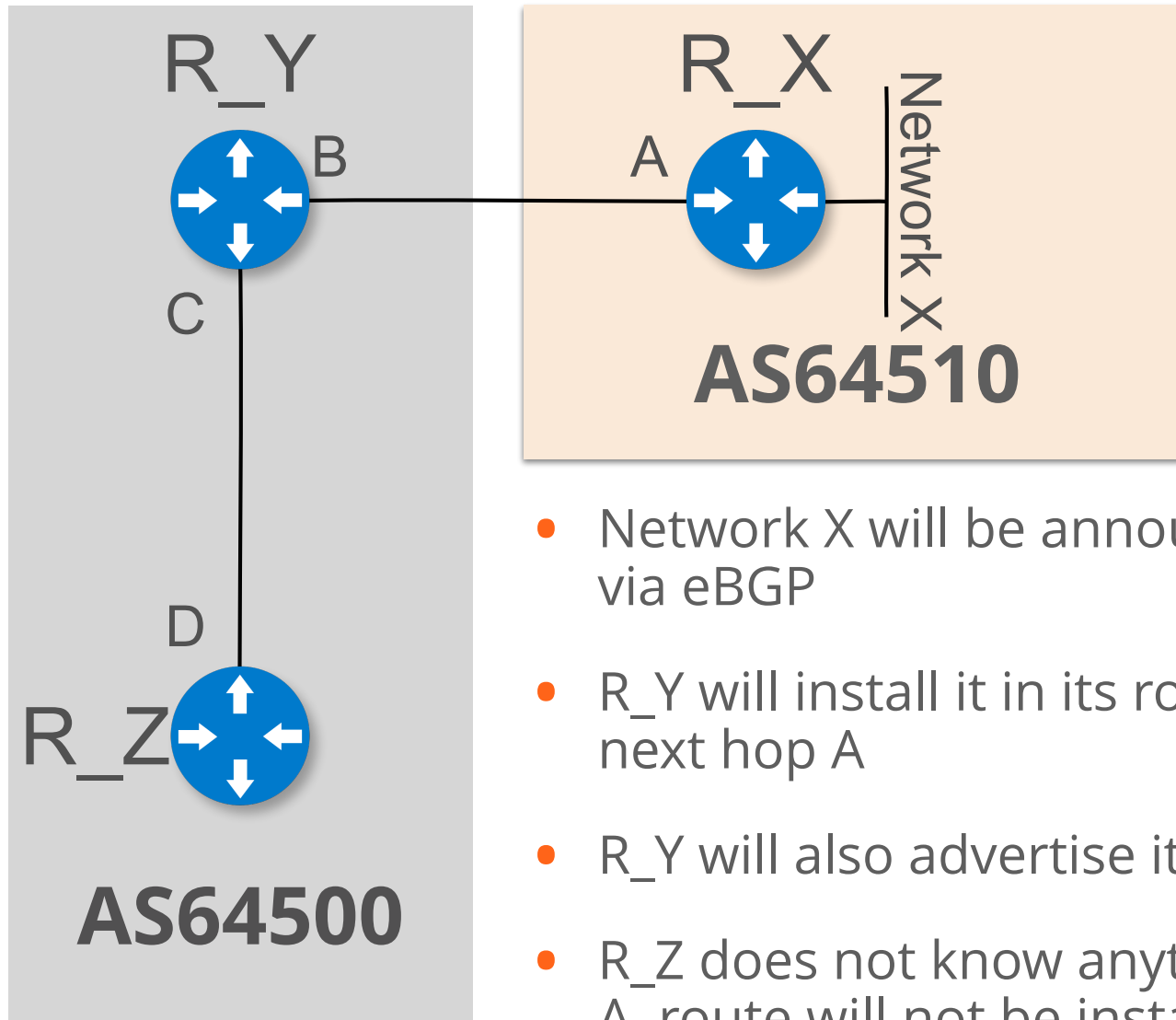
- Now we are going to set up BGP on top of our IPv4 core
- We use the same settings for IPv6 as we have for IPv4

```
neighbor 2001:ffXX::2 remote-as 1XX
neighbor 2001:ffXX::2 update-source 1o0
neighbor 2001:ffXX::3 remote-as 1XX
neighbor 2001:ffXX::3 update-source 1o0
```

```
address-family ipv6
  redistribute static
  neighbor 2001:ffXX::2 activate
  neighbor 2001:ffXX::3 activate
  neighbor 2001:ffXX::2 next-hop-self
  neighbor 2001:ffXX::3 next-hop-self
```



# Why do we use next-hop-self?



- Network X will be announced to R\_Y by R\_X via eBGP
- R\_Y will install it in its routing table with the next hop A
- R\_Y will also advertise it to R\_Z via iBGP
- R\_Z does not know anything about next hop A, route will not be installed
- Solution; **IGP or next-hop-self**



# BGP Configuration R2

- Now we are going to set up BGP on top of our IPv4 core
- We use the same settings for IPv6 as we have for IPv4

```
router bgp 1XX  
  no bgp default ipv4-unicast
```

```
neighbor 2001:ffXX::1 remote-as 1XX  
neighbor 2001:ffXX::1 update-source lo0  
neighbor 2001:ffXX::3 remote-as 1XX  
neighbor 2001:ffXX::3 update-source lo0
```

```
address-family ipv6  
  redistribute static  
  neighbor 2001:ffXX::1 activate  
  neighbor 2001:ffXX::3 activate  
  neighbor 2001:ffXX::1 next-hop-self  
  neighbor 2001:ffXX::3 next-hop-self
```



# BGP Configuration R3

- Now we are going to set up BGP on top of our IPv4 core
- We use the same settings for IPv6 as we have for IPv4

```
router bgp 1XX
  no bgp default ipv4-unicast
```

```
neighbor 2001:ffXX::1 remote-as 1XX
neighbor 2001:ffXX::1 update-source lo0
neighbor 2001:ffXX::2 remote-as 1XX
neighbor 2001:ffXX::2 update-source lo0
```

```
address-family ipv6
  redistribute static
  neighbor 2001:ffXX::1 activate
  neighbor 2001:ffXX::2 activate
  neighbor 2001:ffXX::1 next-hop-self
  neighbor 2001:ffXX::2 next-hop-self
```

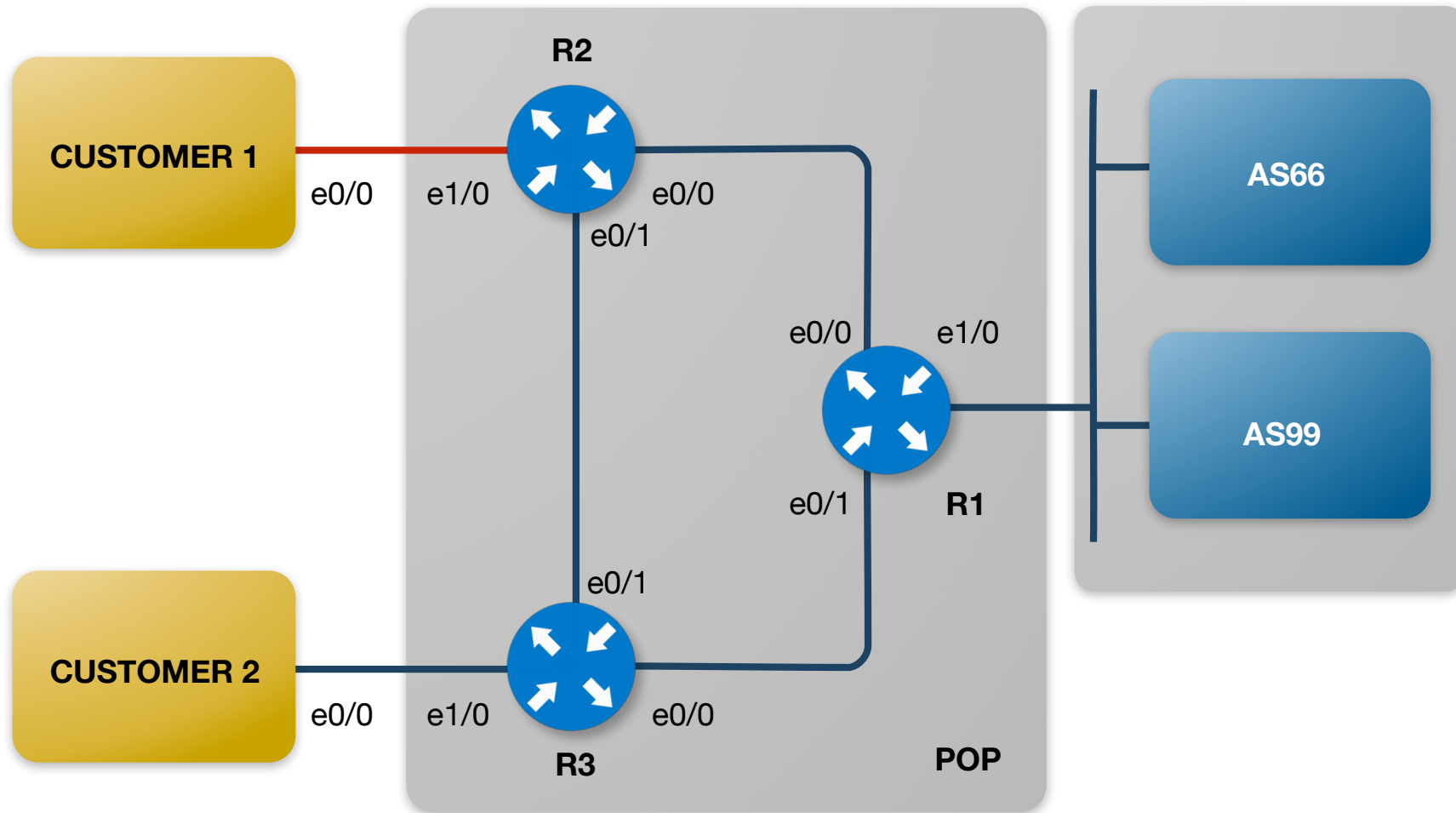




# BGP Customer1

7.3 - Exercise

# Network Diagram





# BGP Configuration Customer 1

- We will remove the default route for the customers

```
no ipv6 route ::/0 2001:ffXX:0:ff01::b
```



# BGP Configuration Router 2

- The AS number for customer is 2 + your number on the participants list (e.g. 209)
- Add BGP session for Customer 1

```
router bgp 1XX  
  neighbor 2001:ffXX:0:ff01::a remote-as 2XX
```

```
address-family ipv6  
  neighbor 2001:ffXX:0:ff01::a activate
```

# BGP Configuration Router 2



- Now add customer prefix to the prefix list
  - Customer 1 prefix: 2001:ffXX:ff01::/48

```
ipv6 prefix-list customer1_v6 seq 5 permit 2001:ffXX:ff01::/48
```

```
router bgp 1XX  
  address-family ipv6  
    neighbor 2001:ffXX:0:ff01::a prefix-list customer1_v6 in
```

```
clear bgp ipv6 unicast 2001:ffXX:0:ff01::a soft in
```



# BGP Configuration Customer 1

- The AS number for customer is 2 + your number on the participants list (e.g. 209)
- Configure BGP session with R2

```
router bgp 2XX
  no bgp default ipv4-unicast
  neighbor 2001:ffXX:0:ff01::b remote-as 1XX
```

```
address-family ipv6
  redistribute static
  neighbor 2001:ffXX:0:ff01::b activate
```

# BGP Configuration Customer 1



- Now add the prefix, prefix list and static route
  - Customer1 prefix: 2001:ffXX:ff01::/48

```
address-family ipv6
  network 2001:ffXX:ff01::/48
(exit) X 2
ipv6 route 2001:ffXX:ff01::/48 Null0
ipv6 prefix-list my_v6 seq 5 permit 2001:ffXX:ff01::/48
```

```
router bgp 2XX
  address-family ipv6
    neighbor 2001:ffXX:0:ff01::b prefix-list my_v6 out
```

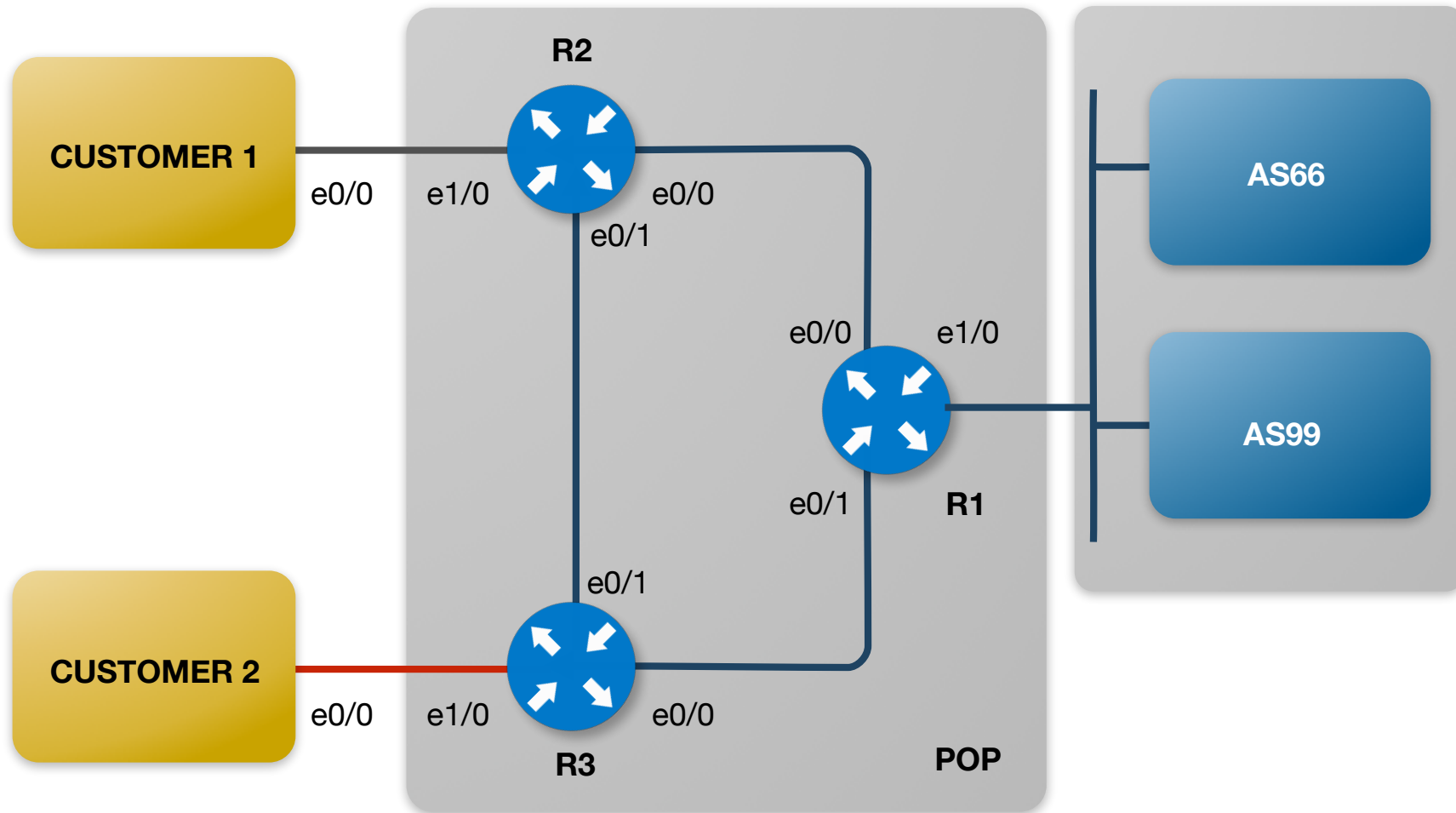


# Challenge: BGP Customer 2

7.4 - Exercise



# Network Diagram



# BGP Configuration Customer 2



- Configure BGP session between Customer 2 router and provider R3
- The AS number for customer is 3 + your number on the participants list (e.g. 309)
- Add the prefix, prefix list and static route
  - Customer2 prefix: 2001:ffXX:ff02::/48



# Summary

- We now added IPv6 to...
  - Links/interfaces
  - IGP (OSPF)
  - EGP (BGP)
  - Customers
- How difficult was it?
- Any surprises?



# Questions





# Content

8 - Section



# Definition

- This section is mostly about websites but it can also apply to:
  - SMTP
  - IMAP4
  - SSH
  - Remote Desktops
  - Cloud Services



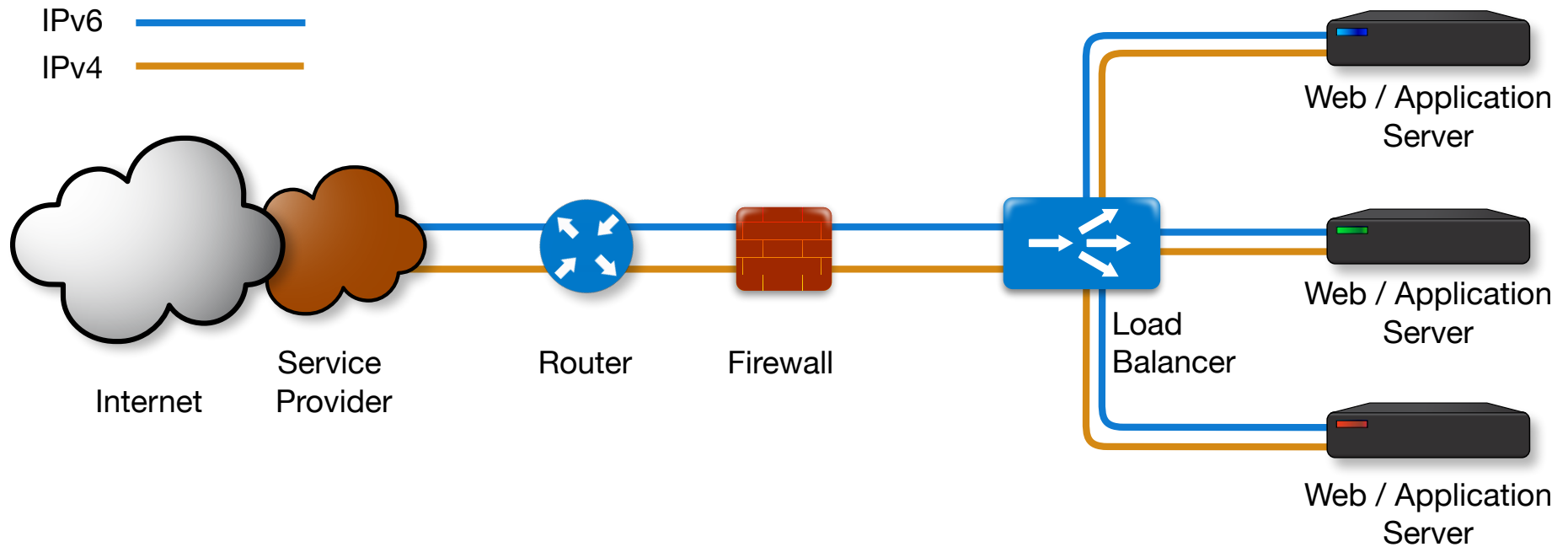
# Options

- Multiple ways to make content dual stack
  - Native dual stack
  - Dual stack load balancer
  - IPv6-to-IPv4 (reverse) proxy
  - NAT64



# Native Dual Stack

- If possible this is the preferred option



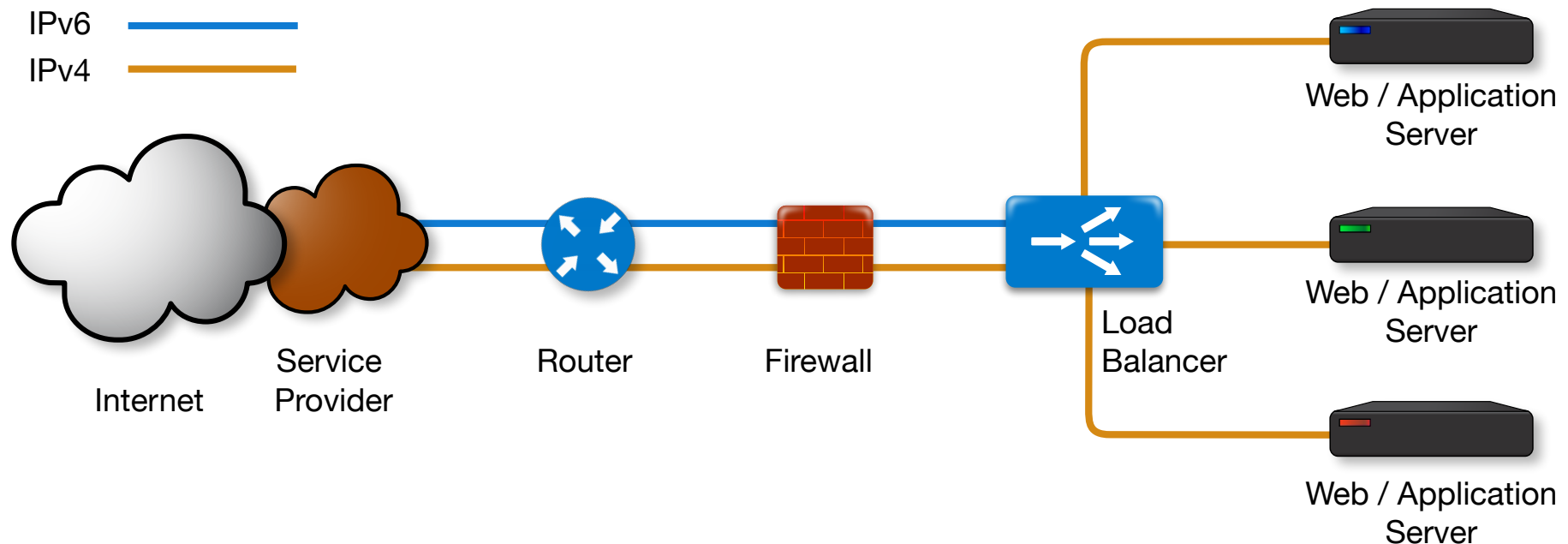
- Cleanest option: no mixing of IPv4 and IPv6
- Needs a fully dual stacked network
- All addresses fully visible where possible



# Load Balancer with NAT or Proxy



- If web servers can't handle IPv6

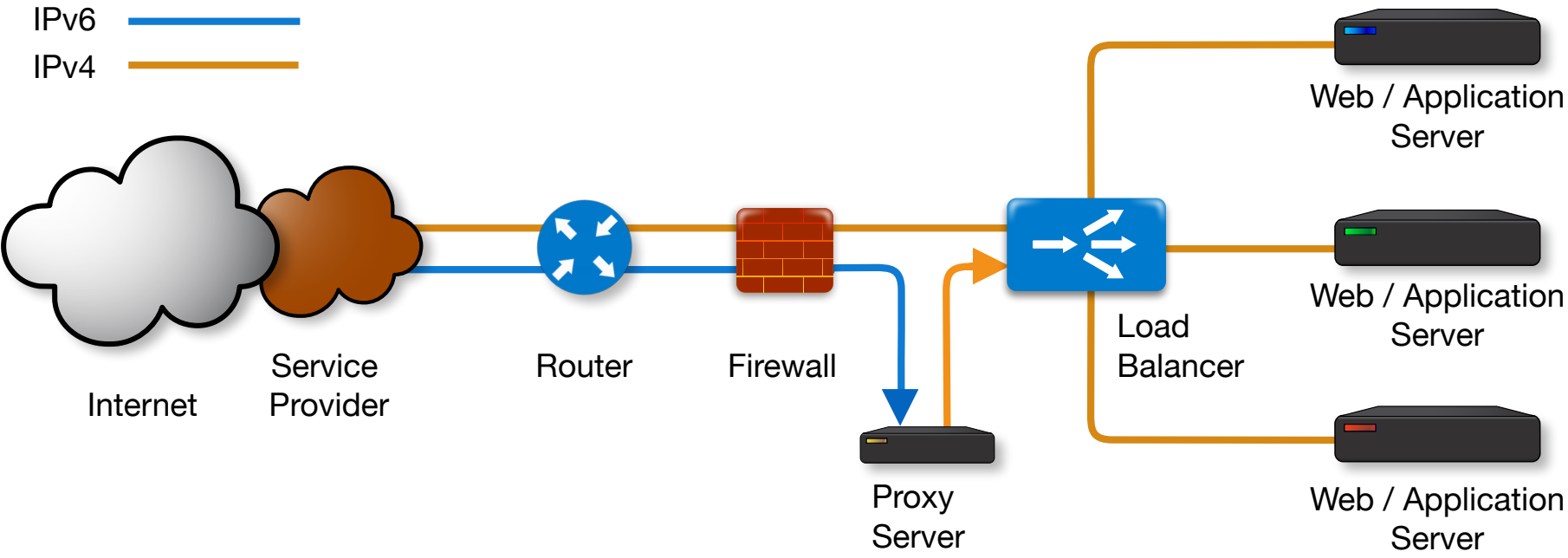


- Needs a fully dual stacked network up to the load balancer
- Web servers might not see IPv6 addresses



# IPv6-to-IPv4 Proxy

- If the load balancer or part of the network can't handle IPv6



- Web servers might not see IPv6 addresses



# Proxy Protocol Level

- You can proxy on
  - Layer 4 (TCP)
  - Layer 7 (HTTP/HTTPS)
- Both have advantages and disadvantages



# Proxy on Layer 4

- Very easy to configure
  - Doesn't need to know about the protocol
  - Doesn't need to be configured with host name
  - Don't need SSL/TLS keys on the proxy server
- Just map an IPv6 address+port to an IPv4 address+port
- Cannot provide information to the servers



# Proxy on Layer 4

- This example shows haproxy
  - Note the confusing notation in the config file
  - IPv6 address = 2001:db8:abc:123::cafe port 25

```
listen smtp1
  bind 2001:db8:abc:123::cafe:25
  mode tcp
  server smtp1 192.0.2.1:25
```



# Proxy on Layer 7

- Bit more work to configure
  - Needs to know about the protocol or application
  - Might need to be configured with host name
  - Needs SSL/TLS keys on the proxy server
- Can provide information to the servers
  - Like X-Forwarded-For header



# Proxy on Layer 7

- This example shows haproxy

```
listen website1
  bind 2001:db8:abc:123::cafe:80
  mode http
  option forwardfor
  server website1 192.0.2.1:80
```

- With SSL

```
listen website1-ssl
  bind 2001:db8:abc:123::cafe:443 ssl
  crt /etc/haproxy/website-ssl.pem
  mode http
  option forwardfor
  server website1 192.0.2.1:443 ssl
```

# Happy Eyeballs Question

What is Happy Eyeballs?

- a. An algorithm for determining which protocol is better to reach a network
- b. A method for deploying IPv6 servers
- c. A new protocol used instead of dual-stack
- d. A medical term for the eye disorder





# Happy Eyeballs Answer

What is Happy Eyeballs?

- ✓ **An algorithm for determining which protocol is better to reach a network**
- b. A method for deploying IPv6 servers
- c. A new protocol used instead of dual-stack
- d. A medical term for the eye disorder





# Happy Eyeballs

- Makes dual-stacked websites more responsive to users
- If there is both A and AAAA
  - First IPv6 is used with a 300 ms head start
  - If that fails, IPv4 is used
- Implemented by all browsers
- Instable connections can cause problems with cookies



# Happy Eyeballs v2 - RFC8305

- New enhancements over Happy Eyeballs v1
  - Asynchronous DNS queries
  - Sorting of resolved destination addresses
  - Asynchronous connection attempts
  - IPv4 Literals & Broken IPv6 Destination Addresses

# IPv6 in the Root Servers and TLDs



- All 13 root servers have IPv6 AAAA records
- There are 1487 TLDs
  - 1465 of them are IPv6 capable (98.5%)
- More than 26% of top 1M domains have IPv6 address
- Over 9 billion websites have AAAA records

Source: <http://bgp.he.net/ipv6-progress-report.cgi>



# Mobile Providers

9 - Section



# IPv6 in Mobile Networks

- IPv4 runout has a high and urgent impact on mobile internet providers
- Everyone has a smartphone
- Certain apps and protocols have problems with double NAT
- Do apps support IPv6?

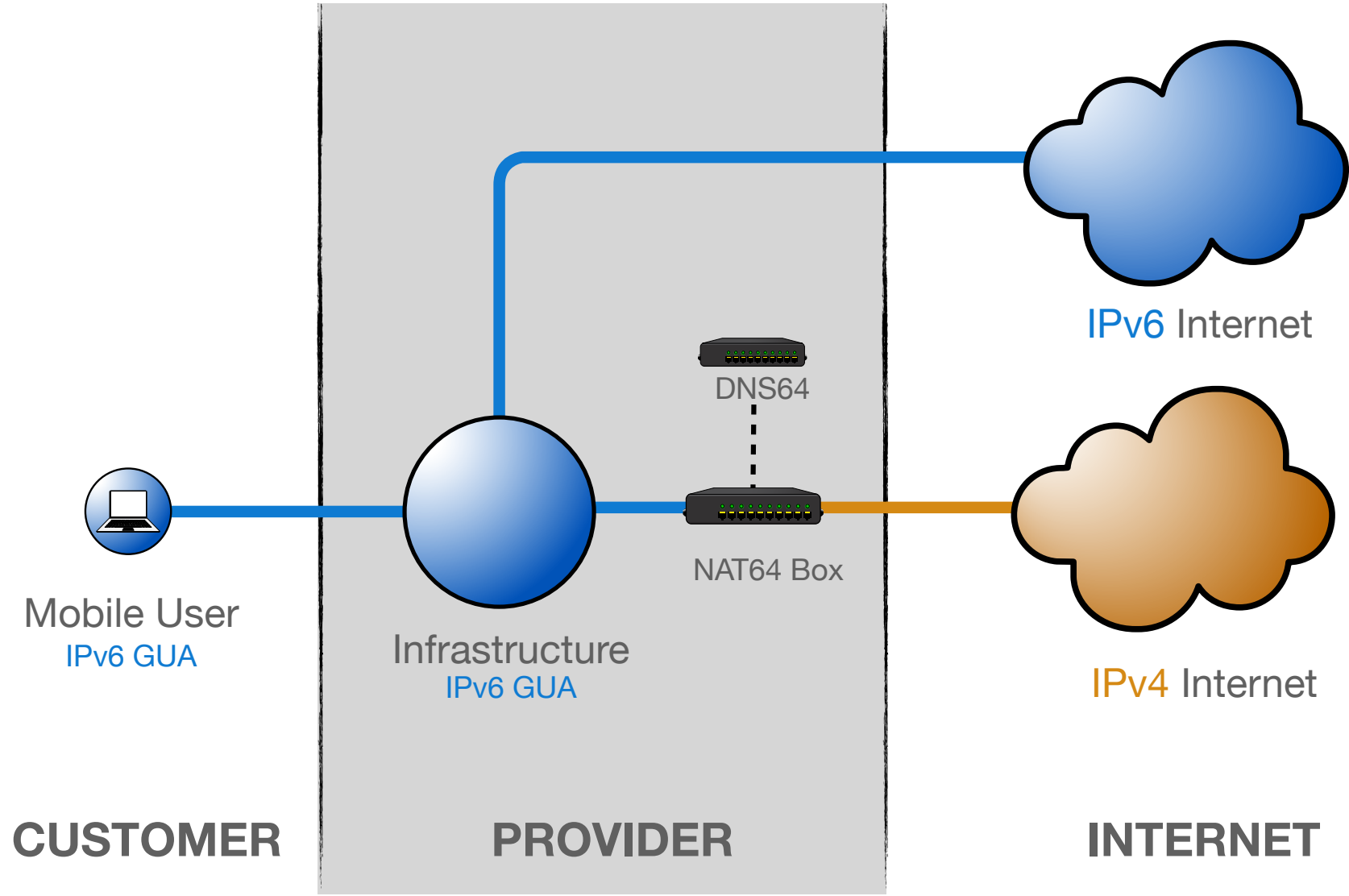


# Multiple Solutions

- Dual Stack users:
  - Public IPv4 and IPv6 GUA
  - Private IPv4 and IPv6 GUA
  - **Easier to implement, hard to maintain**
  
- IPv6 only users:
  - NAT64
  - IPv6 only
  - **Ultimate goal, needs considerable planning and work**



# NAT64/DNS64



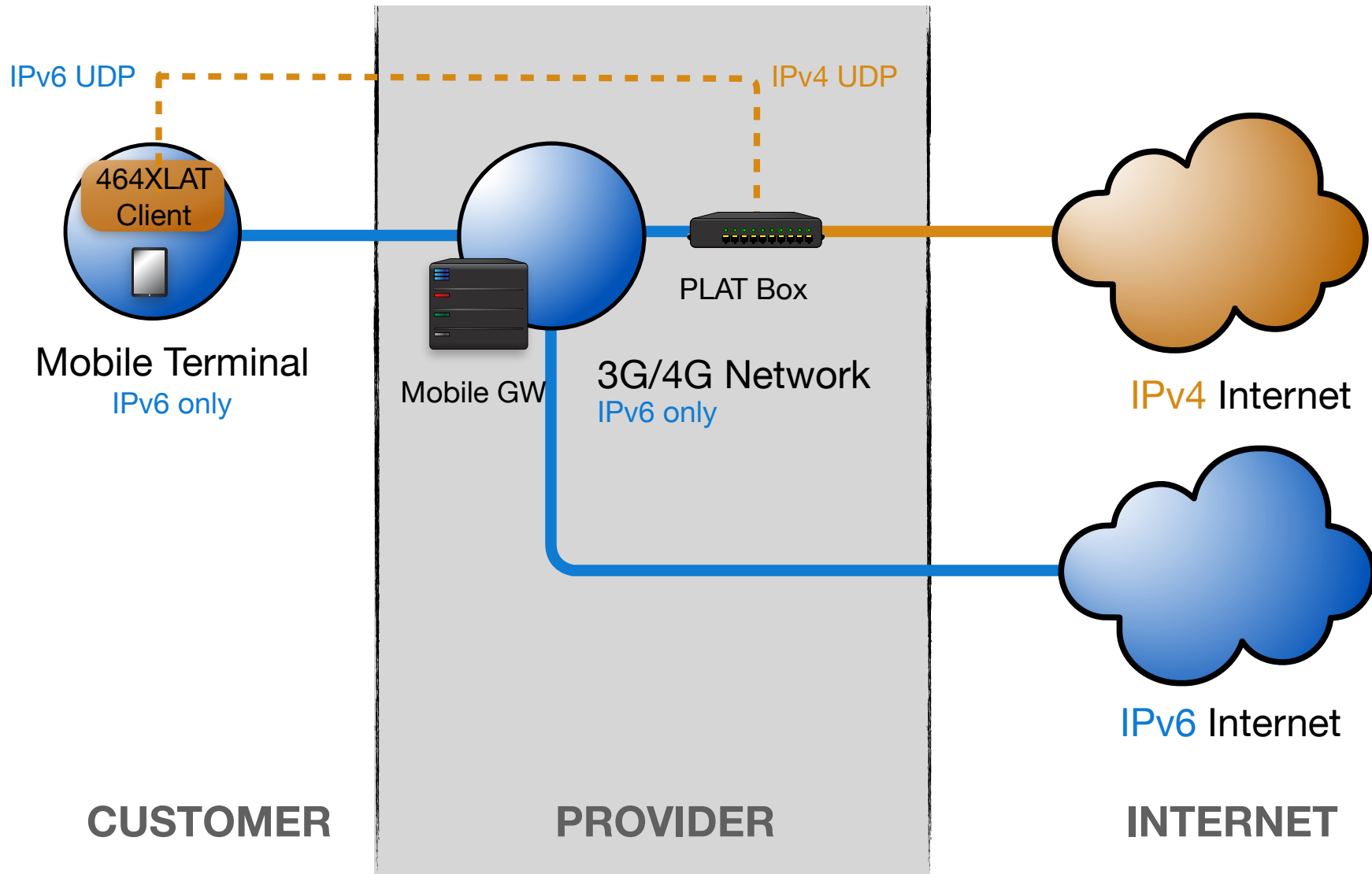




# 464XLAT

- Extension to NAT64 to access IPv4-only applications (legacy applications)
- Handset pretends there is an IPv4 address (CLAT) and sends IPv4 packets in UDP over IPv6
  - 464XLAT has been supported since Android 4.4, so virtually every Android phone has it nowadays.

# 464XLAT





# Apple Approach

- Apple announced they will not support 464XLAT on the iPhone
  - Except for tethering - *since iOS12*
- Instead they urge app developers to make apps work over IPv6-only
- That way, operators can use just NAT64 without 464XLAT

# 3G & 4G



- Stateful DHCPv6-based address configuration is not supported by 3GPP specifications\*
- Only **SLAAC** & **Stateless DHCPv6** supported
- 3GPP relies on RAs
- Only one /64 prefix is allowed

RFC 6459 IPv6 in 3GPP EPS

<https://datatracker.ietf.org/doc/html/rfc6459>

RFC 7066 IPv6 for 3GPP Cellular Hosts - Appendix-A

<https://datatracker.ietf.org/doc/html/rfc7066#appendix-A>

# 3G



- 3G networks work on PDP context
  - On the UE, a PDP context = network interface
- 3G networks support IPv6 since the beginning
- **IPv4v6** is supported since 3GPP Release 9
  - Dual-stack can also be provided by 2 different PDP contexts one per each protocol. But it means 2 different licenses and double resource consumption in the network
- SLAAC is supported

# 4G



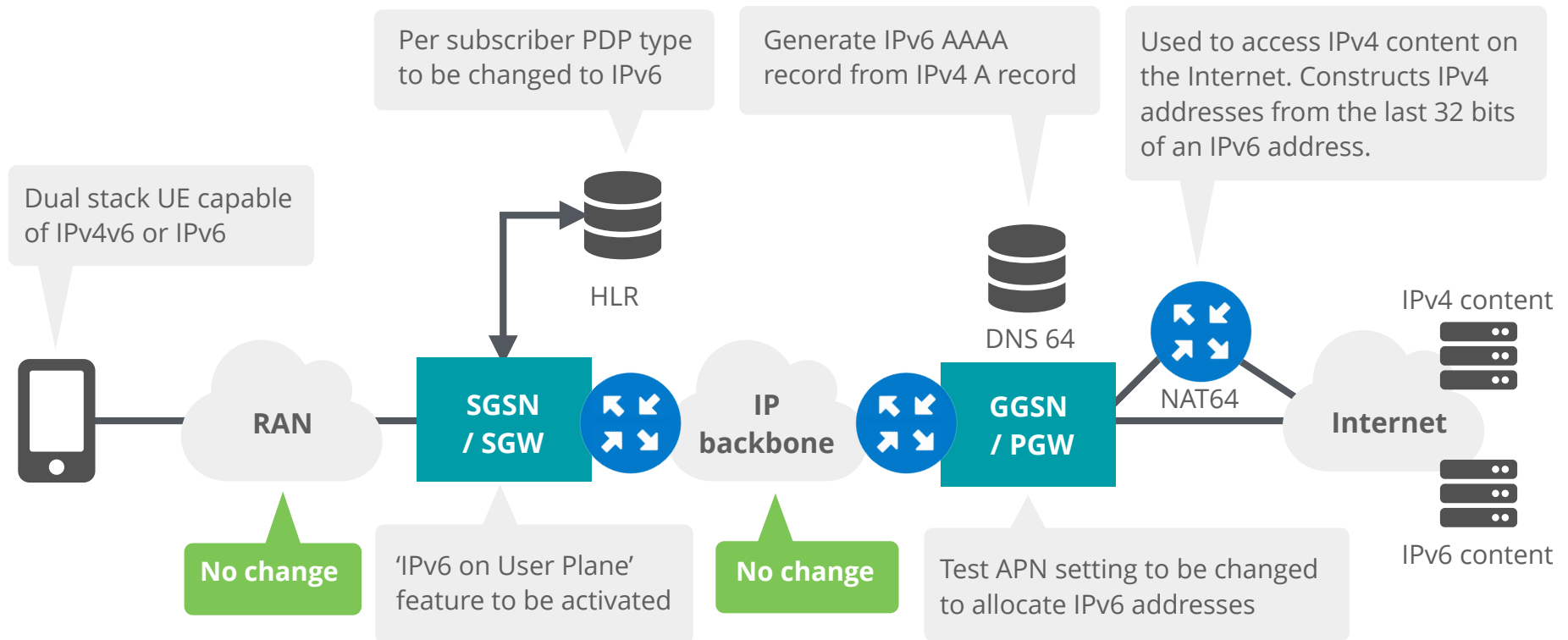
- Works with Evolved Packet System (EPS) Bearer
  - Initiated by the phone to establish a connection
  - IPv4, IPv6 and IPv4v6 supported
- Always-on Packet Data Network (PDN) Connection
- Due to the need for supporting VoIP

# 5G



- 5G is still in evolving phase
- IPv6 is a must for 5G networks as they bring massive amount of connections with mMTC
- Dual-Stack should probably still be supported - Due to the mixed nature of current networks

# What Needs To Be Done



Source: Cameron Byrne T-Mobile USA





# What Needs To Be Done

- Handset:
  - IPv6 capable
- Home Location Register (HLR)
  - Subscriber management needs to understand new PDP types (IPv6, IPv4v6)
- SGSN/SGW
  - 'IPv6 or IPv4v6 on user plane' needs to be enabled
  - Fallback strategy



# What Needs To Be Done

- GGSN/PGW
  - New PDP types (IPv6, IPv6v4)
  - IPv6 routing
  - DHCPv6
  - Neighbor Discovery Protocol
  - DNS Configuration
  - Fallback strategy
  - Billing



# What Needs To Be Done

- And the usual....
  - Firewalls, servers, etc.
  
- Policy Decision Functions (PCRF/PCF)



# Tethering & IPv6

- A /64 prefix is received through an RA to the phone
- An /128 from that /64 is used for the own WAN
- The same /64 is used for the LAN (and for tethering)
  - Tethering is done through RA
  - DAD is used to avoid duplicate addresses



# Challenges

- Only 1 IPv4 address and 1 IPv6 subnet on a handset
- Fallback from IPv4v6 to IPv4-only or IPv6-only is difficult in some cases



# Challenges - Roaming

- Major problems happen at three stages:

## Attach

IPv4v6 PDP/PDN type may not be supported in visited networks which have old versions\* of SGSNs.

## PDP Creation

IPv4v6 and IPv6 PDP support on visited SGSN is the main concern

Single-stack IPv6 in the home routed mode is recommended

## Service Request

464XLAT may cause problems

Disabling local-breakout scenario or supporting IPv4-only on roaming should be considered.



# Overview Day 2

- Transition Mechanisms
- Exercise: NAT64/DNS64
- Host Configuration
- Exercise: SLAAC
- DHCPv6
- Exercise: DHCPv6
- Security
- IP Address Management



# Transition Mechanisms

Section 10



# Transitioning: Solving Two Problems

- Cope with the scarcity of IPv4 addresses
  - Our workarounds today; NAT/CGN/LSN
  - We need a better solution!
- Provide IPv6 connectivity to the host and to the network
  - We need to find a way to connect to the emerging IPv6-only networks.

# Three Different Types of Transition



**Tunnelling**

**v6inv4**

**v4inv6**

6 in4

DS-Lite

6RD

MAP-E



**Translation**

**v6tov4**

**v4tov6**

NAT64

NOT USED

464XLAT

MAP-T



**Dual-Stack**

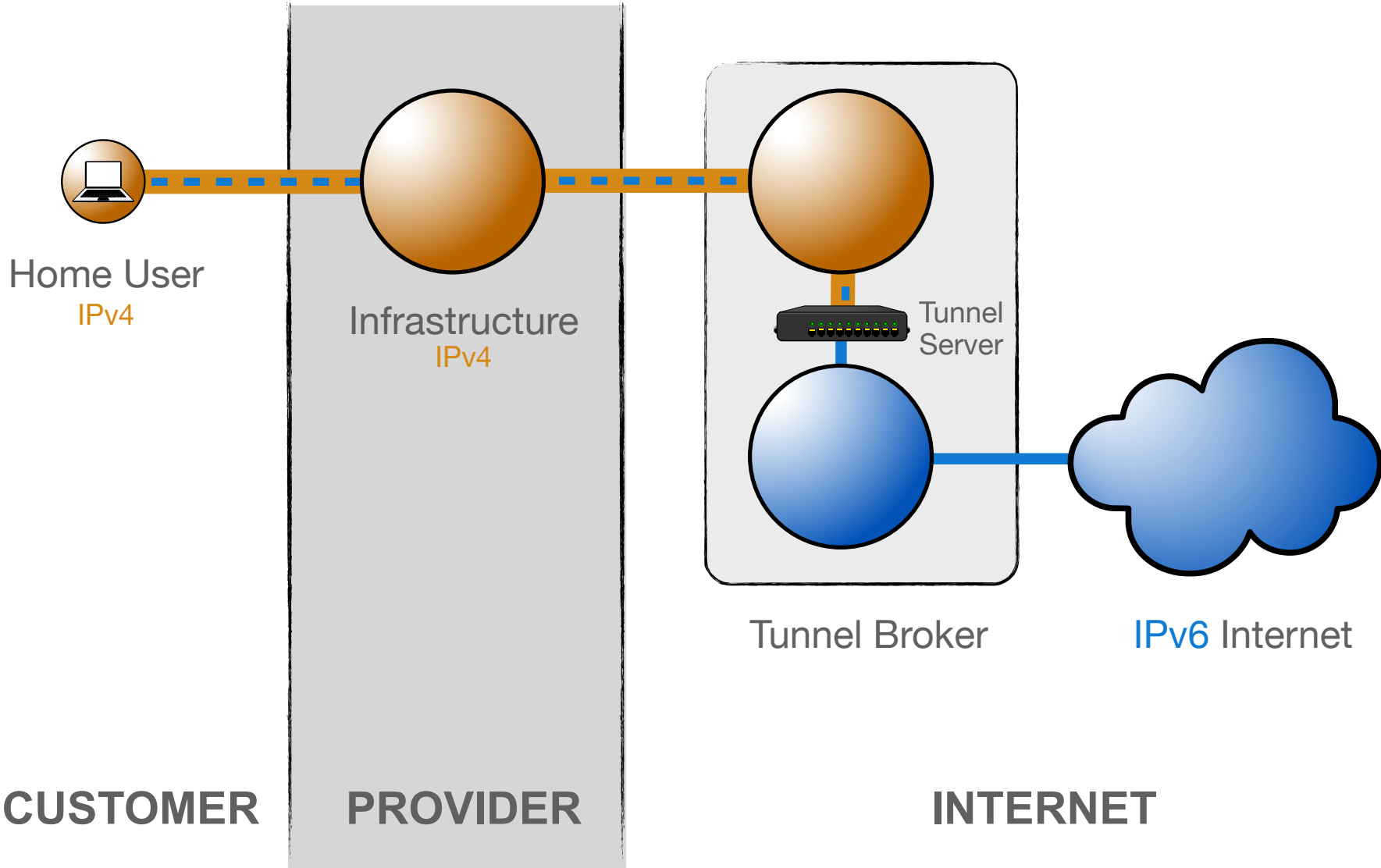


# Tunnelling - 6in4

- Manually configured tunnels towards a fixed tunnel broker like Hurricane Electric or your own system
- Stable and predictable but not easily deployed to the huge residential markets
- MTU might cause issues



# Tunnelling - 6in4



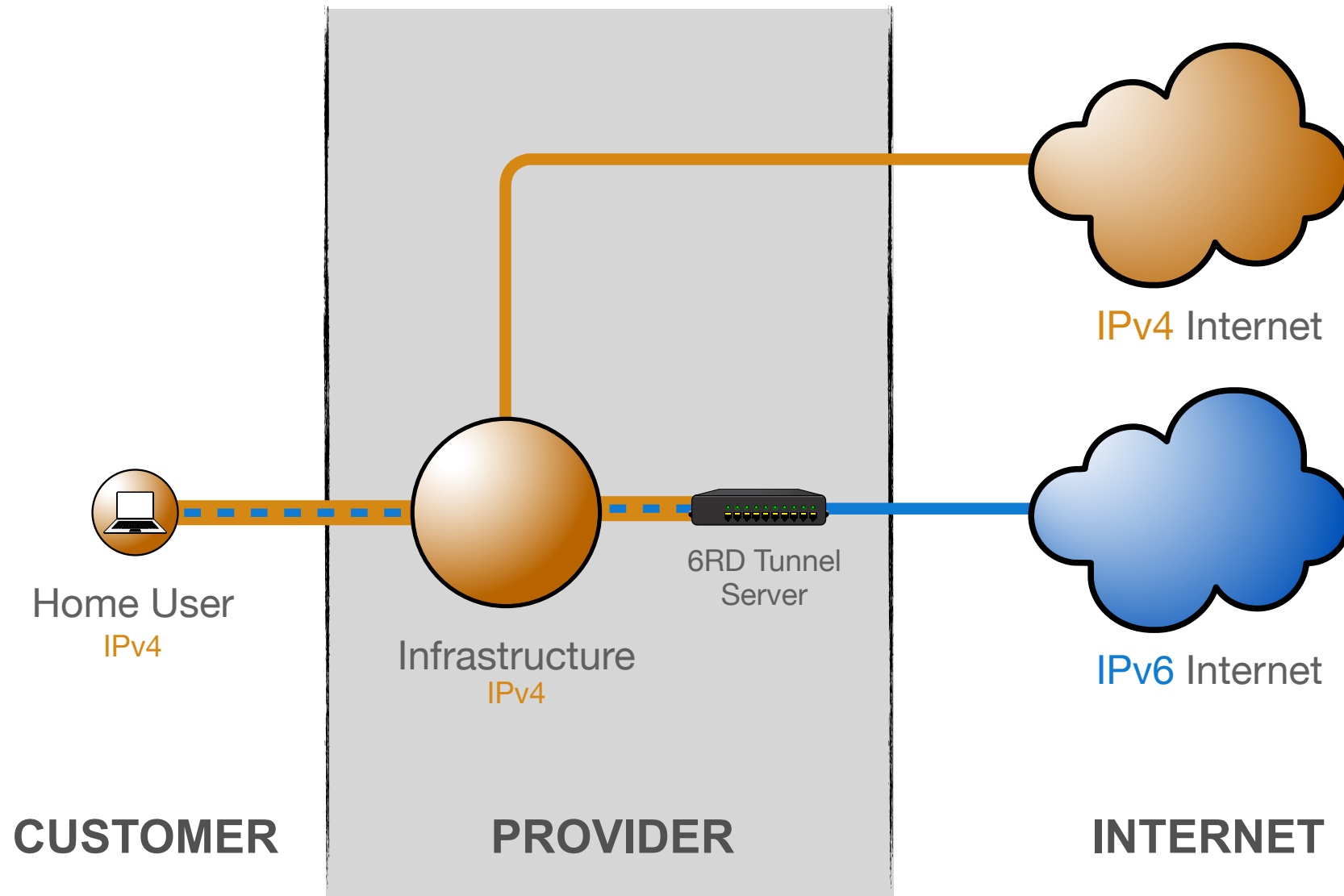


# Tunnelling - 6RD

- Encodes the IPv4 address in the IPv6 prefix
- Uses address space assigned to the operator
- The operator has full control over the relay
- Can work with both public and private IPv4 space
- Needs additional software for signalling
- CPE should also support 6RD



# Tunnelling - 6RD



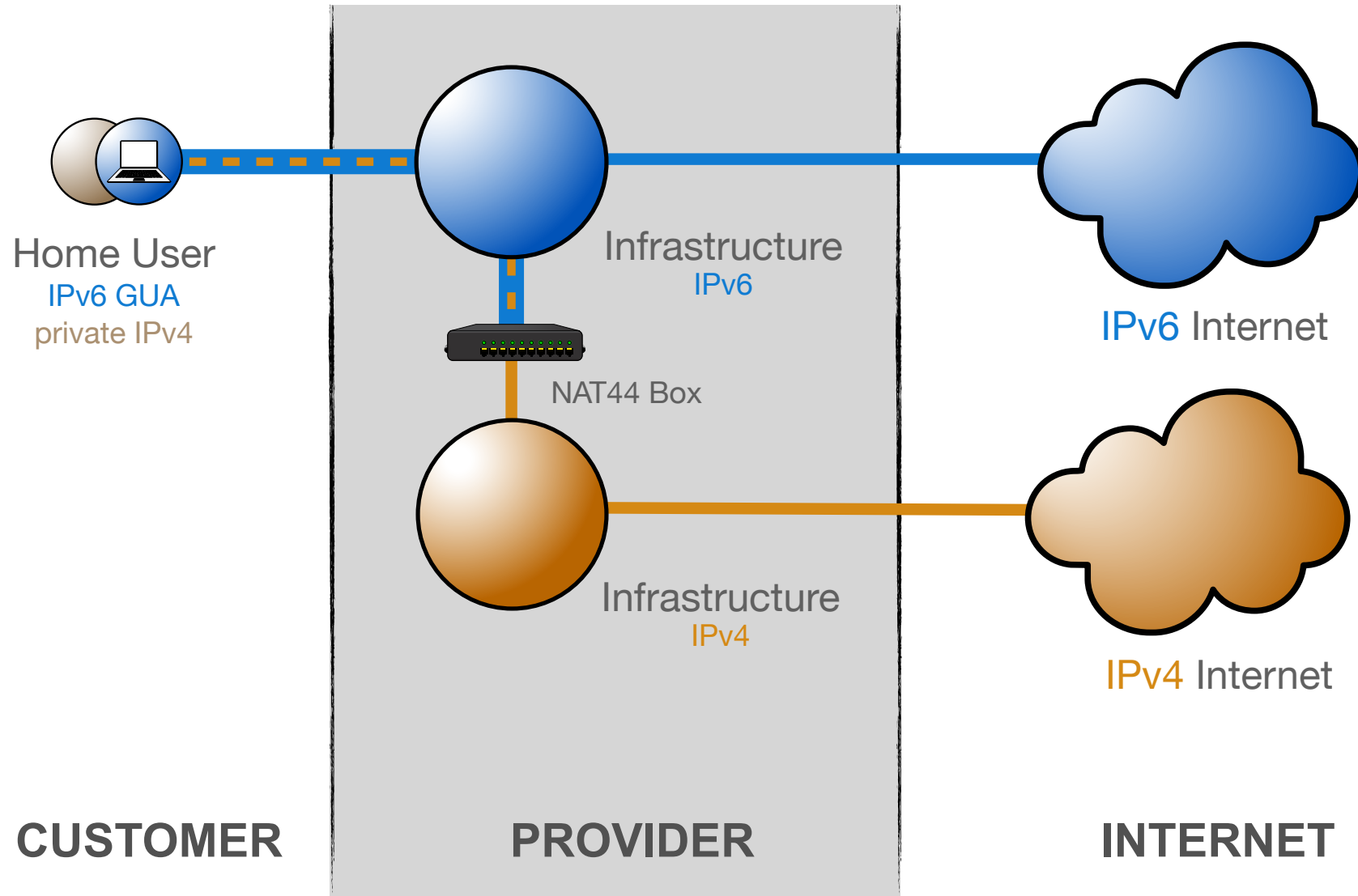


# Tunnelling - DS-lite

- Tunnelling IPv4 over IPv6
- Allows clients to use RFC1918 addresses without doing NAT themselves
- NAT is centrally located at the provider
- Client's IPv6 address is used to maintain state and to keep clients apart
  - Allows for duplicate IPv4 ranges



# Tunnelling - DS-lite





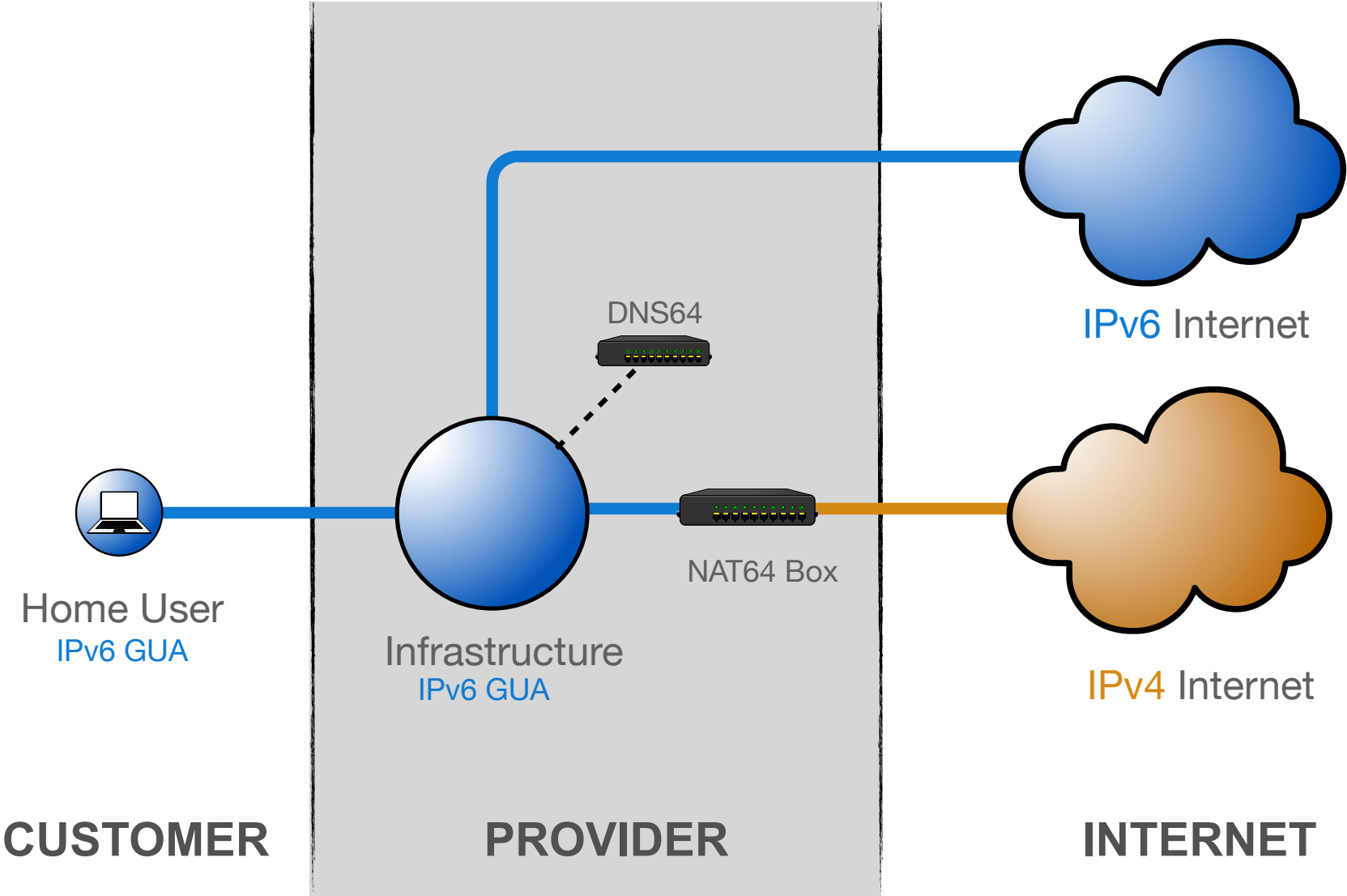


# Translation - NAT64 / DNS64

- Single-stack clients will only have IPv6
- Translator box will strip all headers and replace them with IPv4
- Requires some DNS “magic”
  - Capture responses and replace A with AAAA
  - Response is crafted based on target IPv4 address
- Usually implies address sharing on IPv4



# Translation - NAT64 / DNS64



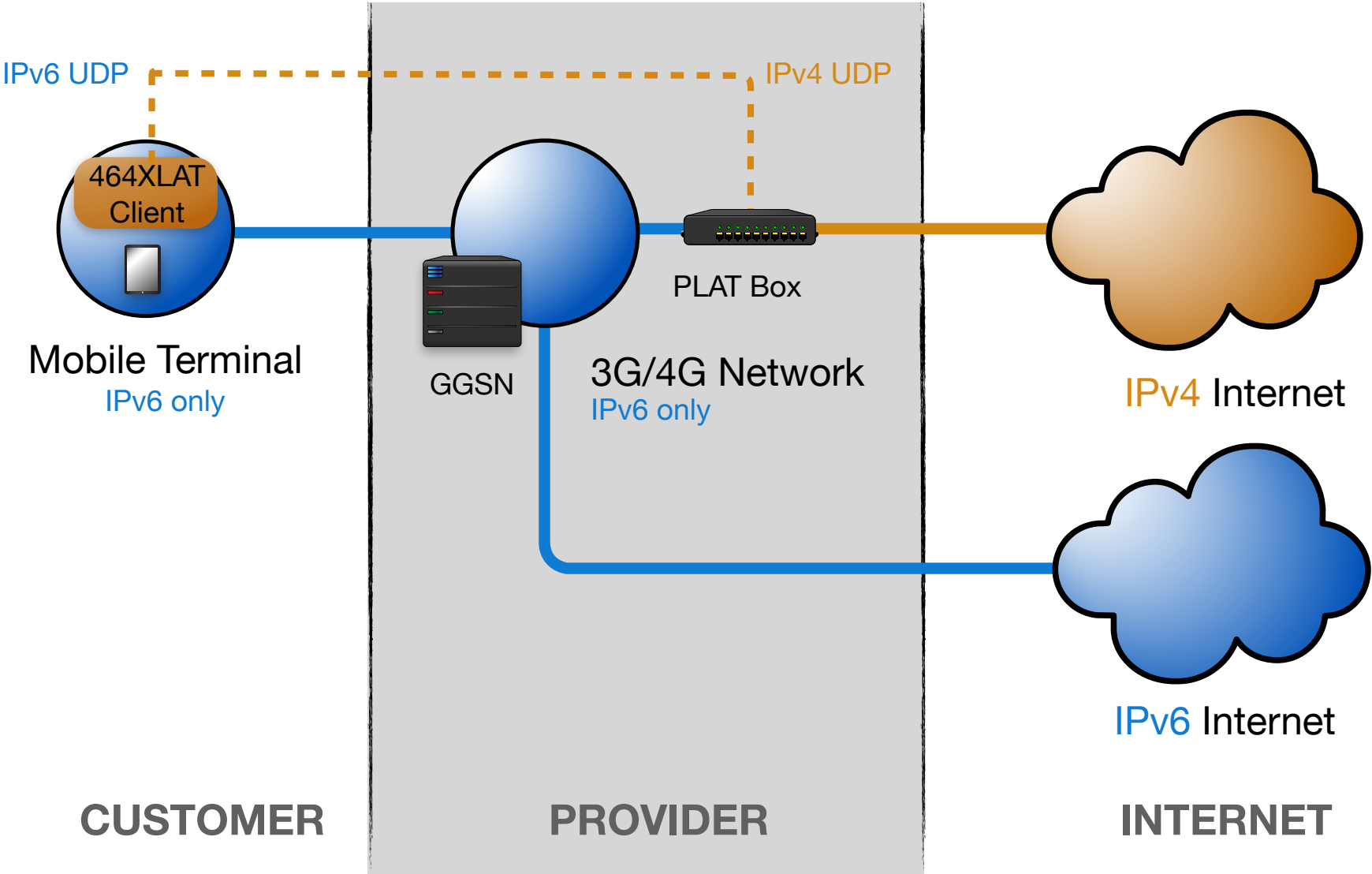


# Translation - 464XLAT

- Extension to NAT64 to access IPv4-only applications (legacy applications)
- Handset pretends there is an IPv4 address and the CLAT translates IPv4 packets to IPv6
- Then in the PLAT translates from IPv6 to IPv4

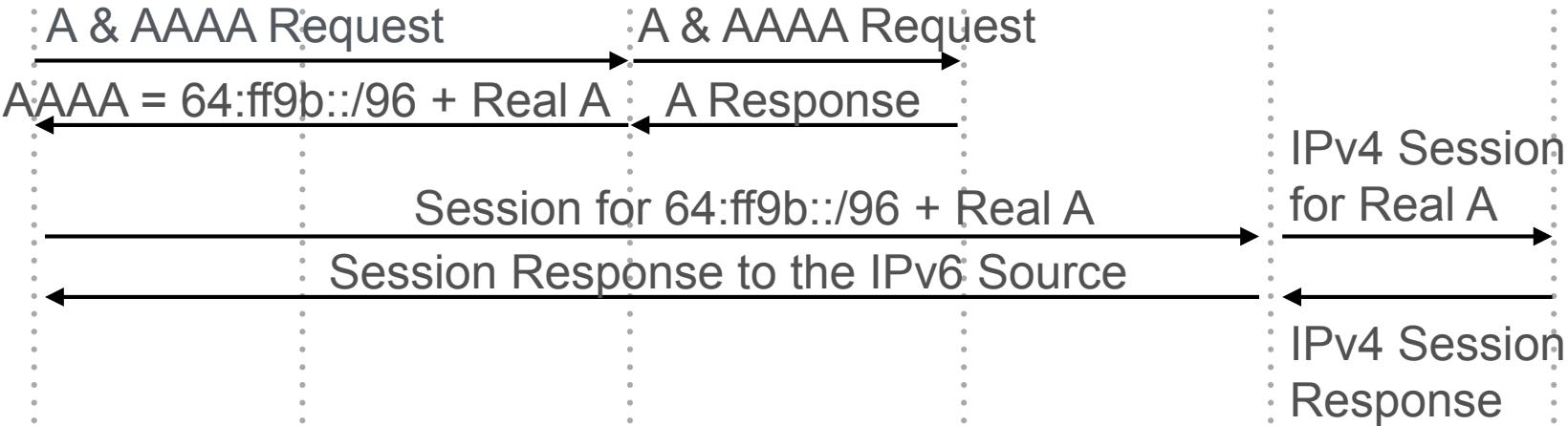
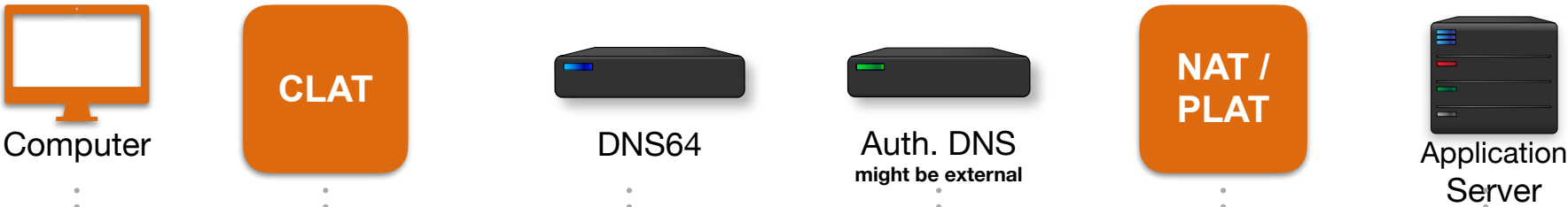


# Translation - 464XLAT



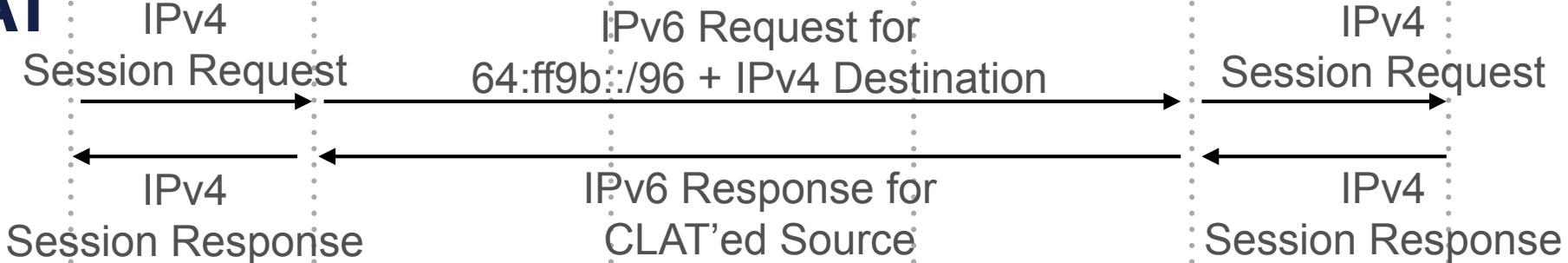


# Translation - NAT64 & 464XLAT



## NAT64

## 464XLAT



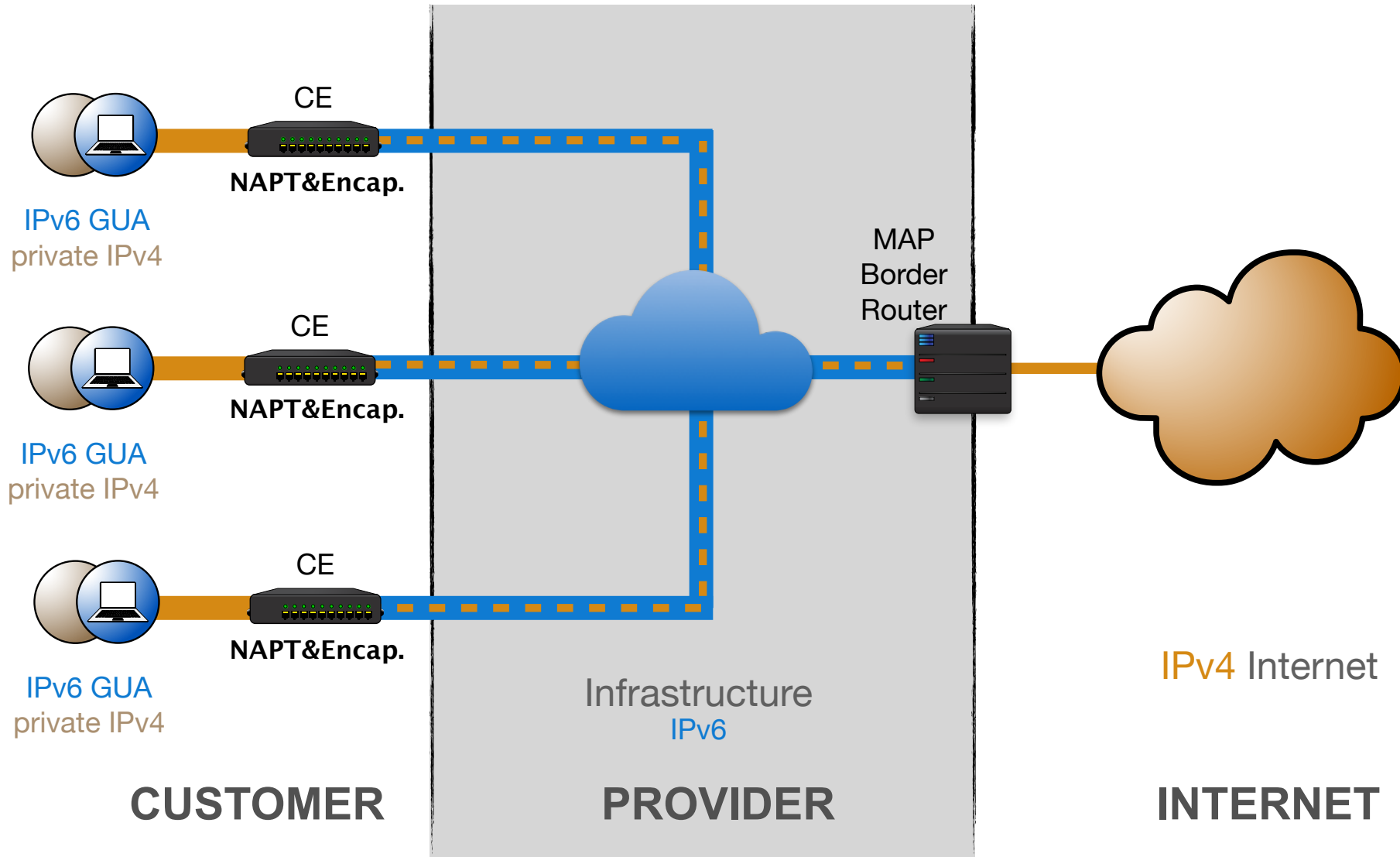


# MAP-E / MAP-T

- IPv4 over IPv6 - Encapsulated or Translated
- Clients get private IPv4 and IPv6 GUA
- IPv4 address/port mapped into IPv6 address
- Stateless NAT64 allows traffic to flow asymmetrically in and out of MAP domain

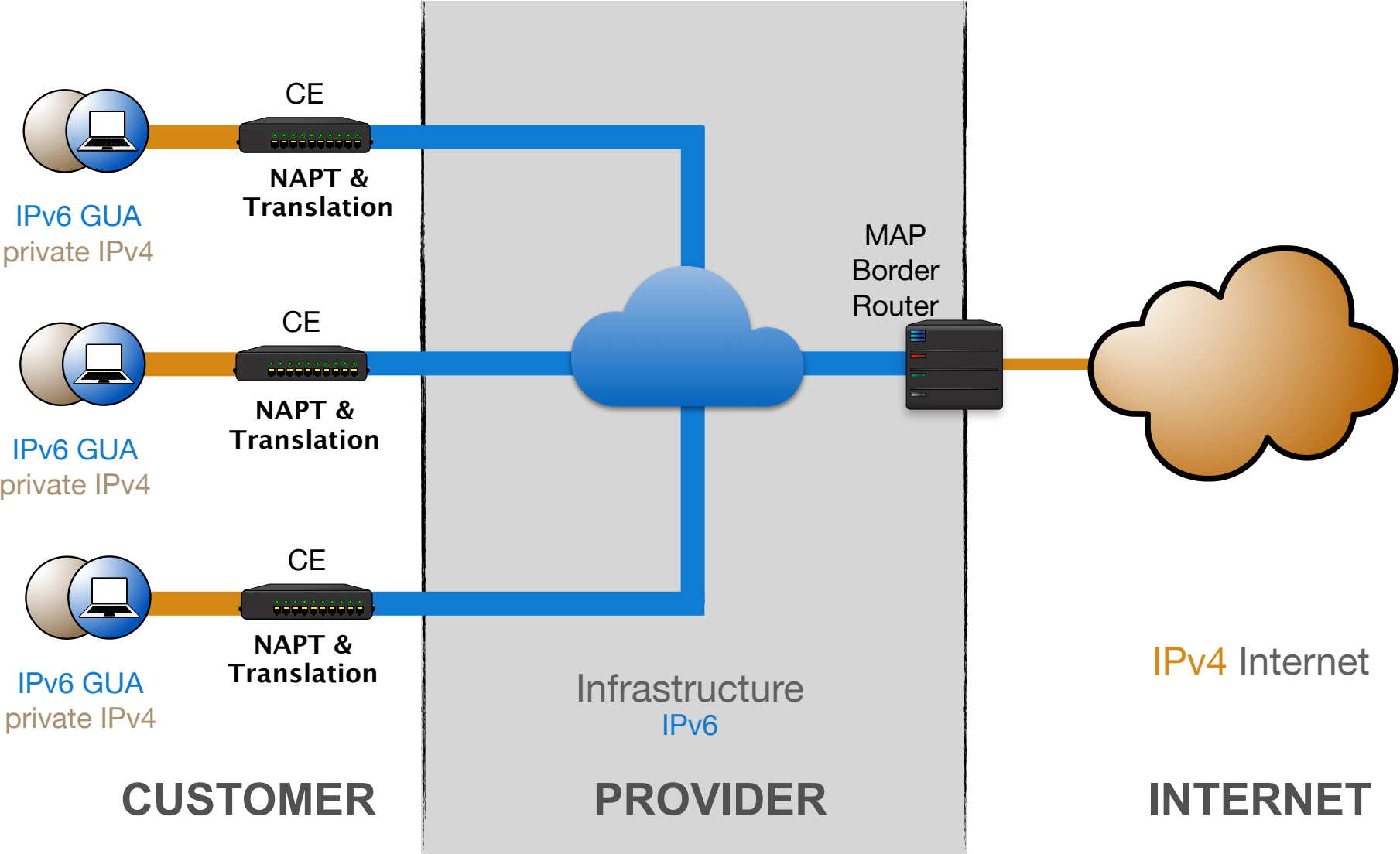


# Tunnelling - MAP-E





# Translation - MAP-T





# Transition Question

Which of the matchings below is not correct?

- a. 6RD; Tunnelling - v6 in v4
- b. NAT64; Translation - v6 to v4
- c. DS-LITE; Tunnelling - v4 in v6
- d. Dual Stack; Translation - v6 to v4



# Transition Answer

Which of the matchings below is not correct?

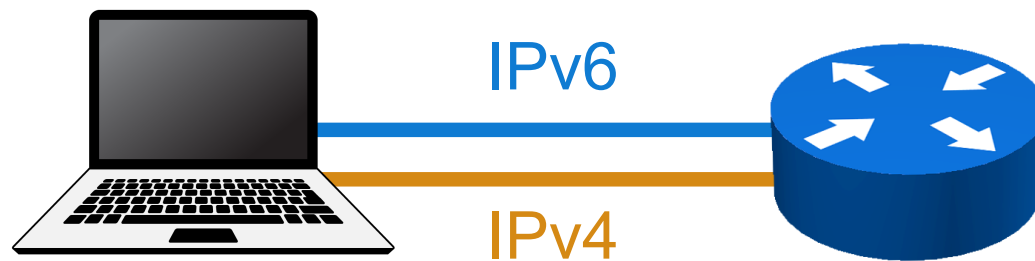
- a. 6RD; Tunnelling - v6 in v4
- b. NAT64; Translation - v6 to v4
- c. DS-LITE; Tunnelling - v4 in v6
- Dual Stack; Translation - v6 to v4**





# What about Dual-Stack?

# Dual Stack





# Configuring NAT64

Section 11

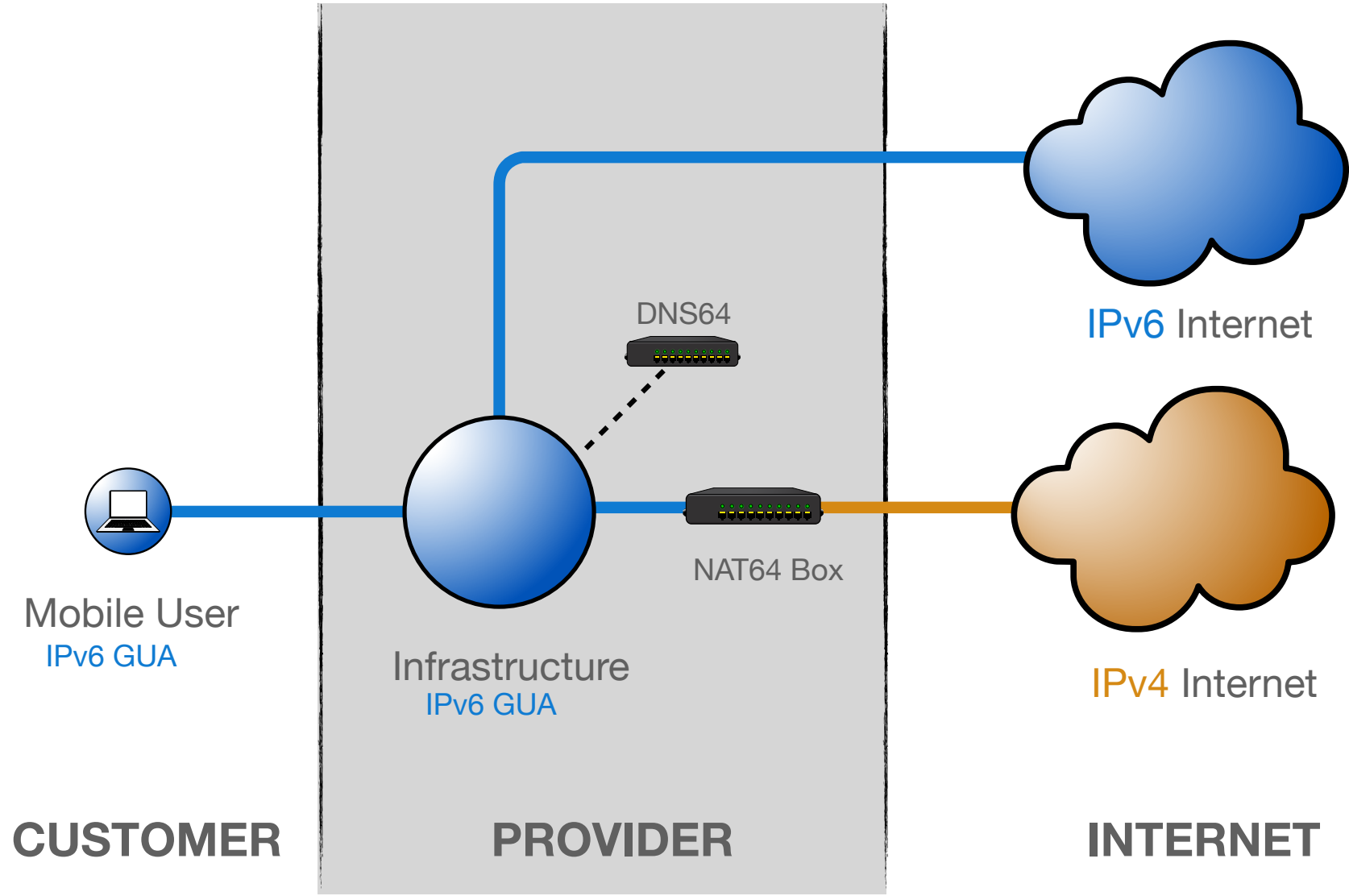


# NAT64 / DNS64

- Single-stack clients will only have IPv6
- Translator box (NAT64) will strip all headers and replace them with IPv4
- Requires some DNS “magic”
  - Capture responses and replace A with AAAA
  - Response is crafted based on target IPv4 address
- Usually implies address sharing on IPv4



# NAT64/DNS64





# Well Known Prefix

- 64:ff9b::/96
  - Algorithmic translation from an IPv4 address to an IPv6 address and vice versa

Well-Known Prefix	IPv4 address	IPv4-Embedded IPv6 address
64:ff9b::/96	192.0.2.33	64:ff9b::192.0.2.33

- Can not be used for private IPv4 addresses
- Only used within the operator, not announced in global routing table
- Or use your own prefix, but be careful

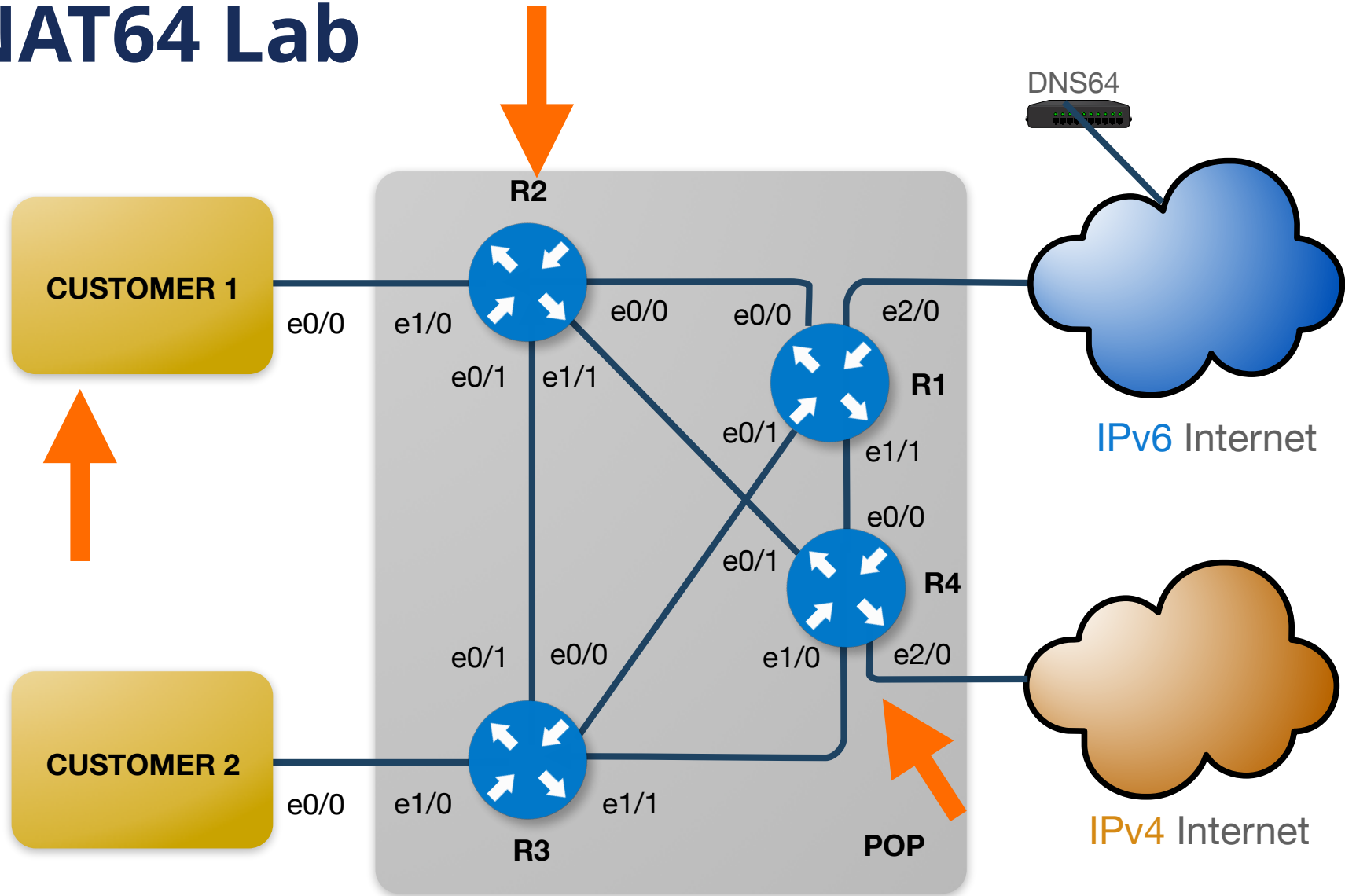


# NAT64 Lab

- The problem is: We will have IPv6-only customers that cannot reach IPv4-only content
- Step 1: Configure C1 as an IPv6-only customer
- Step 2: Ping from C1 to [www.example.com](http://www.example.com) (dual-stack)
- Step 3: Ping from C1 to [ipv4.example.com](http://ipv4.example.com) (v4 only)
- Step 4: Setup R4 as NAT64 box
- Step 5: Set static route on R2
- Step 6: Ping from customer to [ipv4.example.com](http://ipv4.example.com) (v4 only)



# NAT64 Lab





# Step 1: Configure C1

- The customer will only have an IPv6 address

```
ipv6 unicast-routing  
ipv6 cef
```

```
interface e0/0  
  ipv6 address 2001:ffXX:0:ff01::a/64  
  no ipv6 redirects  
  ipv6 nd ra suppress all  
  no shut
```

- Set up a static route to R2

```
ipv6 route 2000::/3 2001:ffXX:0:ff01::b  
ipv6 route 64:ff9b::/96 2001:ffXX:0:ff01::b
```

- Set up a name server

```
ip domain lookup  
ip name-server 2001:ff53::53
```



# Step 2: Ping from C1 to dual-stack

- Ping from the customer to a dual-stacked website

```
ping www.example.com
```



# Step 3: Ping from C1 to IPv4-only

- Ping from the customer to an IPv4-only website

```
ping ipv4.example.com
```

- What's the IPv6 address that ping tries to use?



# Step 4: Setup R4 as NAT64 box

- Enable IPv6

```
ipv6 unicast-routing  
ipv6 cef
```

- Configure the loopback interface for IPv4 & IPv6

```
interface lo0  
  ipv6 address 2001:ffXX::4/128  
  no ipv6 redirects
```



# Step 4: Setup R4 as NAT64 box

- Configure the 3 internal interfaces

```
interface e0/0
  ipv6 address 2001:ffXX:0:04::a/127
  no ipv6 redirects
  ipv6 nd ra suppress all
```

```
interface e0/1
  ipv6 address 2001:ffXX:0:05::a/127
  no ipv6 redirects
  ipv6 nd ra suppress all
```

```
interface e1/0
  ipv6 address 2001:ffXX:0:06::a/127
  no ipv6 redirects
  ipv6 nd ra suppress all
```



# Step 4: Setup R4 as NAT64 box

- Setup OSPFv3 on the router

```
ipv6 router ospf 1
 log-adjacency-changes
 router-id 172.X.255.Y    (Y is router number)
 passive-interface Ethernet2/0
 redistribute connected
```

- Setup OSPFv3 on the three interfaces : (e0/0, e0/1, e1/0)

```
interface xyz
 ipv6 ospf network point-to-point
 ipv6 ospf 1 area 0
 no shut
```



# Step 4: Setup R4 as NAT64 box

- Configure the transit interface

```
(config)# interface Ethernet2/0  
(config-if)# ip address 10.132.x.2 255.255.255.252  
(config-if)# no shut
```

- Test if transit provider router is reachable

```
# ping 10.132.x.1
```





# Step 4: Setup R4 as NAT64 box

- Create a filter

```
(config)# ip prefix-list transit-out-v4 seq 5 permit 10.X.0.0/22
```

- Configure the BGP session

```
(config)# router bgp 1XX  
(config-router)# bgp log-neighbor-changes  
(config-router)# neighbor 10.132.X.1 remote-as 44  
(config-router)# neighbor 10.132.X.1 prefix-list transit-out-v4 out
```

- Set the network statement

```
(config-router)# network 10.X.0.0 mask 255.255.252.0
```



# Step 4: Setup R4 as NAT64 box

- Insert static Null route
  - Before BGP advertised its network, it checks for an exact match of network number and mask on router's routing table

```
(config)# ip route 10.x.0.0 255.255.252.0 null0
```



## Step 4: Setup R4 as NAT64 box

- From R4 ping the IPv4-only host behind the transit

```
ping 193.0.21.80
```

# Step 4: enable NAT64 on interfaces



- Interface lo0, e0/0, e0/1, e1/0 and e2/0

```
interface xyz
  nat64 enable
```

- On the router define the prefix used for the translations

```
(config)# nat64 prefix stateful 64:ff9b::/96
```



# Step 4: Setup R4 as NAT64 box

- Set up an access list

```
(config)# ipv6 access-list allow-nat64  
(config-acl)# permit ipv6 2001:ffXX::/32 any  
(config-acl)# exit
```



## Step 4: Setup R4 as NAT64 box

- Define the pool of IPv4 addresses used for the translation

```
(config)# nat64 v4 pool nat64-v4-pool 10.x.3.0 10.x.3.255
```

- Map pool and access list

```
(config)#nat64 v6v4 list allow-nat64 pool nat64-v4-pool overload
```



# Step 5: Setup a static route on R2

- We need a static route from R2 to R4

```
(config)# ipv6 route 64:ff9b::/96 2001:ffXX::4
```

- We need to route our special prefix “64:ff9b::/96” to the NAT64 box

# Step 6: ping ipv4.example.com from C1



```
C1# ping ipv4.example.com
```

- On R4 verify how NAT64 is working

```
R4# show nat64 translation
```





# Questions





# Host Configuration

Section 12



# Operating Systems

- We will look at Windows, MacOS
- All of them support IPv6 natively
- We will check:
  - How to configure them for DHCPv6.
  - What is supported and what's not.
  - How to see the IPv6 configuration.
  - How to change the IPv6 configuration.



# Managing clients

- Users might not notice that their computer is using IPv6
- For management purposes, you want control over addresses. You may want to disable;
  - SLAAC,
  - Privacy extensions
- Disabling SLAAC does not mean disabling Router Advertisements



# Windows 10 & 11

- DHCPv6 works off the shelf
- GUI by default in auto configuration mode:

General

You can get IPv6 settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IPv6 settings.

Obtain an IPv6 address automatically

Use the following IPv6 address:

IPv6 address:

Subnet prefix length:

Default gateway:

Obtain DNS server address automatically

Use the following DNS server addresses:

Preferred DNS server:

Alternate DNS server:

Validate settings upon exit

Advanced...

OK Cancel



# DHCPv6 Client on Windows

- Some versions did not do DHCP out-of-the-box
  - Some may have it disabled by default
  - Or it was somehow disabled
  
- Let's see how to check it and enable it



# DHCPv6 client on Windows

- First, get the interface ID:

```
netsh interface ipv6 show interfaces
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>netsh int ipv6 show int

Idx      Met      MTU      State      Name
-----
1        50      4294967295  connected  Loopback Pseudo-Interface 1
14       50      1280     disconnected isatap.guestnet.ripe.net
13       10      1500     connected  Local Area Connection 2

C:\Users\IEUser>
```




# Check DHCPv6 configuration

- To check the configuration (run as Admin):

```
netsh interface ipv6 show interface XX
```

- If it is not enabled:

```
Neighbor Discovery : enabled
Neighbor Unreachability Detection : enabled
Router Discovery : enabled
Managed Address Configuration : disabled
Other Stateful Configuration : disabled
Weak Host Sends : disabled
Weak Host Receives : disabled
```







# DHCPv6 client on Windows

- With the interface ID instead of the red **XX**, run the command (as Admin):

```
netsh interface ipv6 set interface XX  
advertise=enabled managed=enabled
```




# Check configuration

- To check the configuration (run as Admin):

```
netsh interface ipv6 show interface XX
```

- If it is enabled:

```
Neighbor Discovery           : enabled
Neighbor Unreachability Detection : enabled
Router Discovery             : enabled
Managed Address Configuration : enabled
Other Stateful Configuration  : disabled
Weak Host Sends              : disabled
Weak Host Receives          : disabled
```





# Windows and DHCPv6

- You can either:
- Configure a router to supply the “M” flag
  - But with no prefix announced
- Or disable router discovery
  - And let other clients pick up addresses from SLAAC



# Windows 7

- By default, many services/protocols are enabled:
  - Privacy extensions
  - Teredo
  - 6to4
  - ISATAP
- You might want to disable some/all of them



# Windows 7: Before

```
Command Prompt

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix . : guestnet.ripe.net
    IPv6 Address . . . . . : 2001:67c:2e8:13:15df:c4ba:ac73:a179
    Temporary IPv6 Address . . . . . : 2001:67c:2e8:13:298a:3de:3711:2dd
    Link-local IPv6 Address . . . . . : fe80::15df:c4ba:ac73:a179%13
    IPv4 Address . . . . . : 193.0.10.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::13:0:0:1%13
    . . . . . : 193.0.10.1

Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix . :
    IPv6 Address . . . . . : 2001:0:9d38:90d7:2840:ada:3eff:f5d8
    Link-local IPv6 Address . . . . . : fe80::2840:ada:3eff:f5d8%14
    Default Gateway . . . . . :

Tunnel adapter isatap.guestnet.ripe.net:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . : guestnet.ripe.net

C:\Users\IEUser>
```



# De-configuring Windows 7

- First, disable all the transition methods
- On the command prompt, as administrator:

```
netsh interface ipv6 6to4 set state state=disabled  
netsh interface ipv6 isatap set state state=disabled  
netsh interface ipv6 set teredo disable
```



# Windows 7 Privacy Extensions

- Disable privacy extensions

```
netsh interface ipv6 set privacy state=disable
```

```
netsh interface ipv6 set global randomizeidentifier=disabled
```



# Windows 7:After

```
Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix . : guestnet.ripe.net
    IPv6 Address . . . . . : 2001:67c:2e8:13:15df:c4ba:ac73:a179
    Link-local IPv6 Address . . . . . : fe80::15df:c4ba:ac73:a179%13
    IPv4 Address . . . . . : 193.0.10.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::13:0:0:1%13
                                193.0.10.1

Tunnel adapter isatap.guestnet.ripe.net:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . : guestnet.ripe.net

C:\Users\IEUser>
```





# MacOS

- It will automatically configure IPv6
- It will look in the RA messages to check M flag
  - If present, it will check with DHCPv6



# MacOS Configuration



Wi-Fi

Wi-Fi | TCP/IP | DNS | WINS | 802.1X | Proxies | Hardware

Configure IPv4: Using DHCP

IPv4 Address: 192.168.178.20

Renew DHCP Lease

Subnet Mask: 255.255.255.0

DHCP Client ID:

(If required)

Router: 192.168.178.1

Configure IPv6: Automatically

Router: fe80::420d:10ff:fe8a:ce78

IPv6 Address	Prefix Length
2001:1c04:2b1c:9900:18a3:1134:8f6:3058	64
2001:1c04:2b1c:9900:7d61:ec2e:5184:8e05	64

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=6463<RXCSUM, TXCSUM, TS04, TS06, CHANNEL_IO, PARTIAL_CSUM, ZEROINVERT_CSUM>
ether f8:4d:89:81:56:c6
inet6 fe80::14f0:2563:da5d:cddc%en0 prefixlen 64 secured scopeid 0xe
inet6 2001:1c04:2b1c:9900:18a3:1134:8f6:3058 prefixlen 64 autoconf secured
inet6 2001:1c04:2b1c:9900:7d61:ec2e:5184:8e05 prefixlen 64 autoconf temporary
inet 192.168.178.20 netmask 0xfffff00 broadcast 192.168.178.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active Network
```





# Linux

- As client, same behaviour as MacOS
  - Everything works out of the box
  - IPv6 is enabled automatically
- As server, static configuration is required



# Linux Static configuration

- For CentOS/Red Hat:
- /etc/sysconfig/network
- Add:

```
NETWORKING_IPV6=yes
```



# Linux Static configuration

- In `/etc/sysconfig/network-scripts/ifcfg-ethX`
- Add:

```
IPV6INIT=yes
IPV6ADDR=2001:0db8:aaaa:bbbb:0000:0000:0000:0002/64
IPV6_DEFAULTGW=2001:db8:aaaa:bbbb:0000:0000:0000:0001
IPV6_AUTOCONF=no
```

- Where **X** is the number of the interface, then:

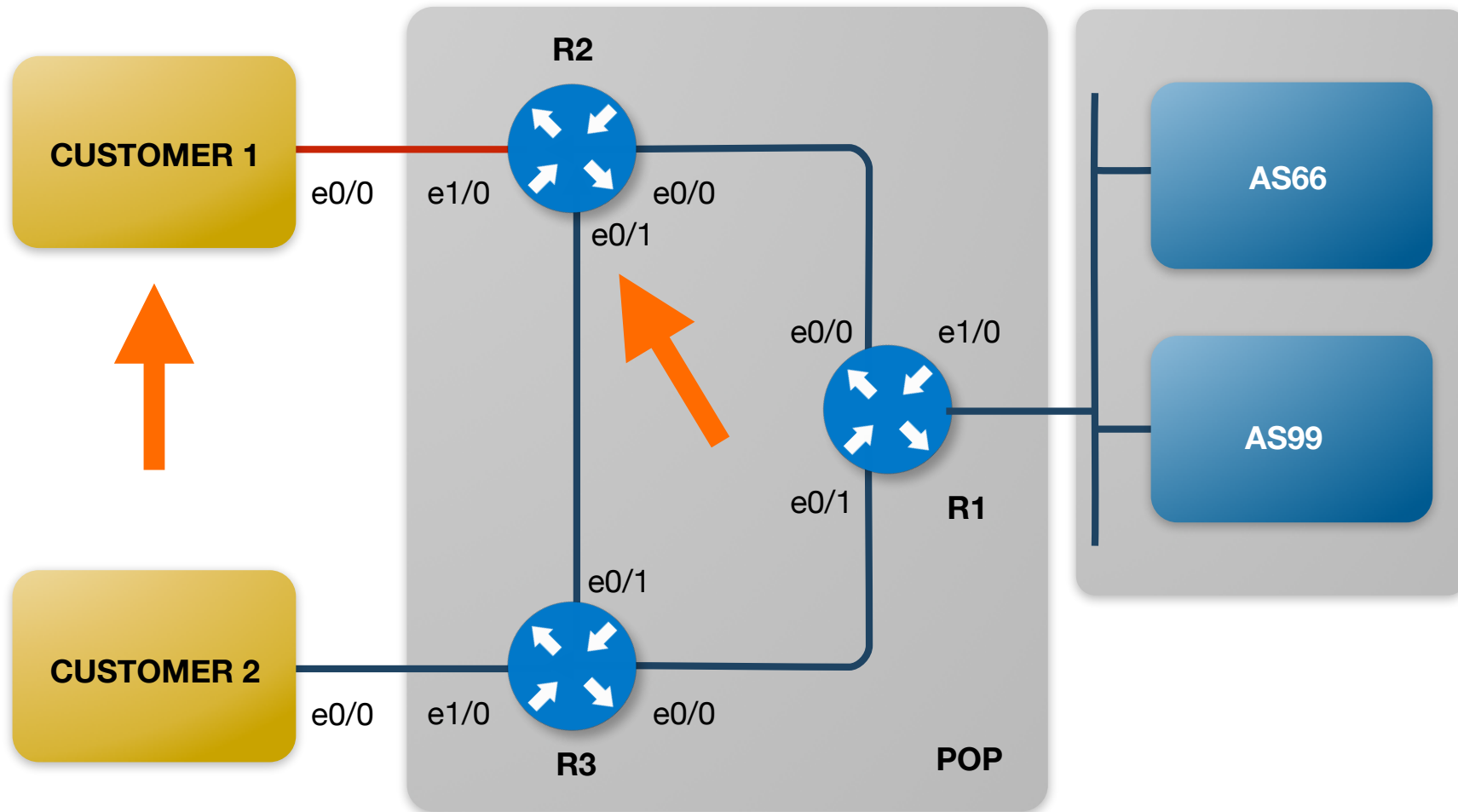
```
service network restart
```



# SLAAC

Section 13

# SLAAC





# On C1

- Now we will enable SLAAC

```
interface e0/0
  ipv6 address autoconfig default
  no shutdown
```

- Leave configuration mode

```
debug ipv6 nd
```

- Enable debug ND:

```
undebug all
```

- To stop debug messages:





# On R2

- Now we will remove the suppression

```
interface e1/0
  ipv6 address 2001:ffXX:0:ff01::b/64
  no ipv6 nd ra suppress all
  no shutdown
```

- Leave configuration mode

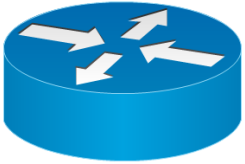


# Debugging SLAAC

- Can you find the new IPv6 address?
- Look at the routing...
- Do you see any interesting debug messages?

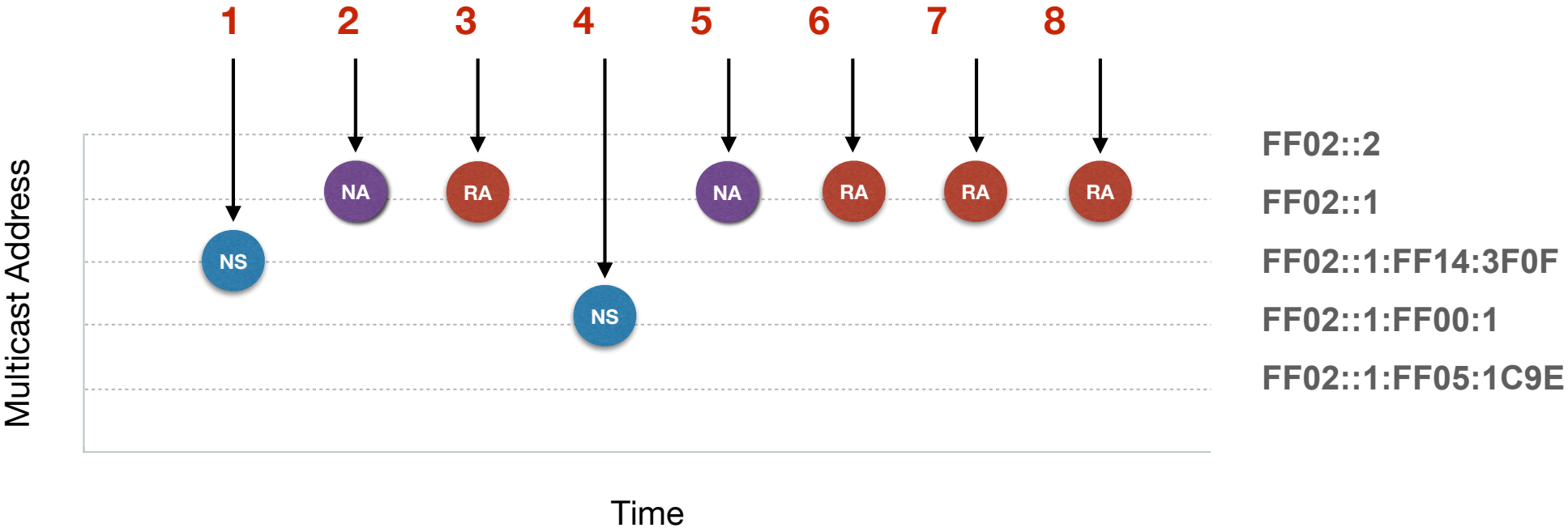


# SLAAC: Router Messages



Link-local: fe80::a390:45ff:fe14:3f0f

Global unicast: 2001:db8:a:b::1

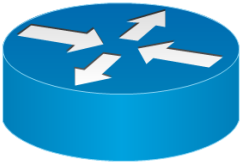


Link-local: fe80::ba8d:12ff:fe05:1c9e

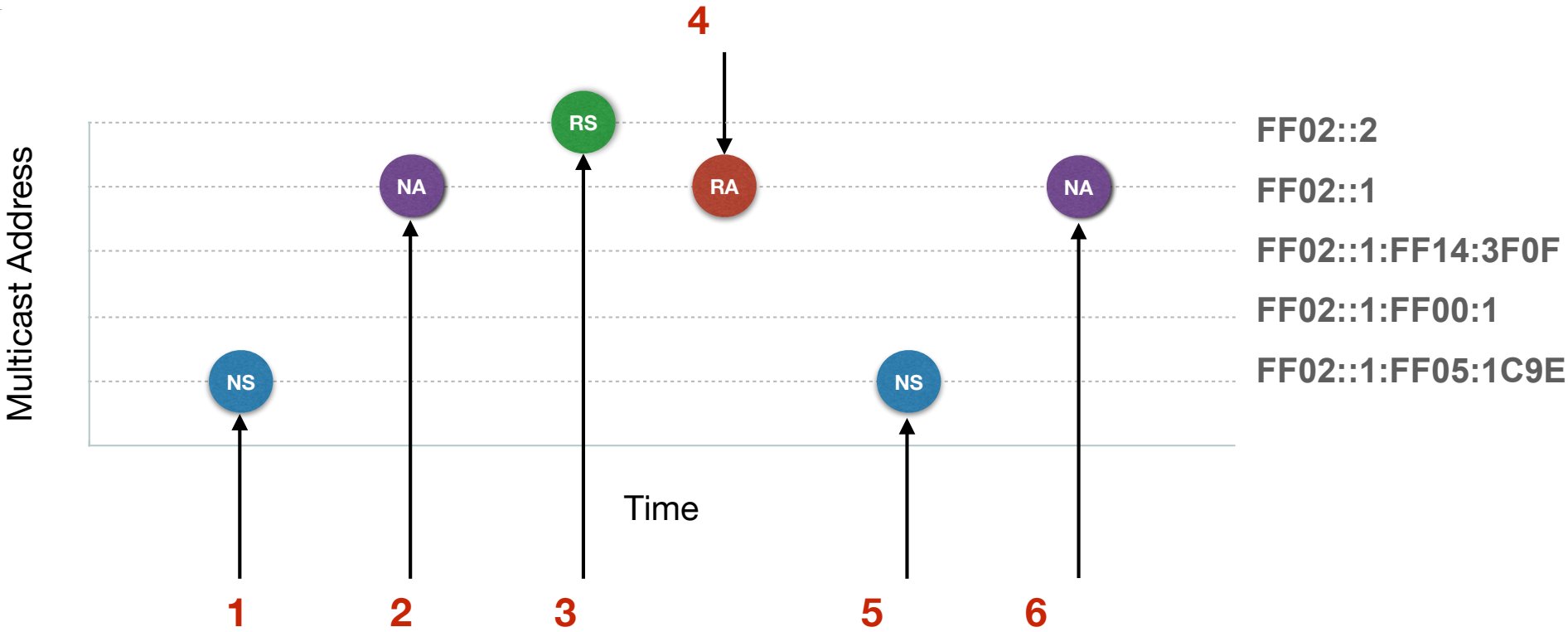
Global unicast: 2001:db8:a:b:ba8d:12ff:fe05:1c9e



# SLAAC: Client Messages



Link-local: fe80::a390:45ff:fe14:3f0f  
Global unicast: 2001:db8:a:b::1



Link-local: fe80::ba8d:12ff:fe05:1c9e  
Global unicast: 2001:db8:a:b:ba8d:12ff:fe05:1c9e



# Questions





# DHCPv6

Section 14



# About DHCPv6

- Requires IPv6 transport
- Offer similar functionality to DHCPv4 but for IPv6
- Allows more control than SLAAC
  - Routers and servers can have static or dynamic assignments
- Is supported by Cisco IOS, Microsoft, etc.

# Information provided by DHCPv6



- No routing information is distributed
  - no default route (routers in IPv6 have different roles in the network)
- Only host configuration protocol
- Other configuration parameters
  - includes DNS, NTP, domain name, TFTP server etc





# DHCPv6 Fundamentals

- Client driven via DHCPv6 request message
- DHCPv6 uses;
  - UDP port 546 for clients
  - UDP port 547 for server
- DHCPv6 options are similar to those in DHCPv4
- DHCP-PD can deliver prefix information to the routers to be used by the host behind them.



# DHCPv6 Operation

- Client first detects the presence of routers on link
- Client examines router advertisements to check if DHCP can be used (managed flag)
- If no router is found or if DHCP can be used, the client:
  - sends DHCP solicit message to “all-DHCP-agents” multicast address (ff02::1:2)
  - uses link-local address as source address

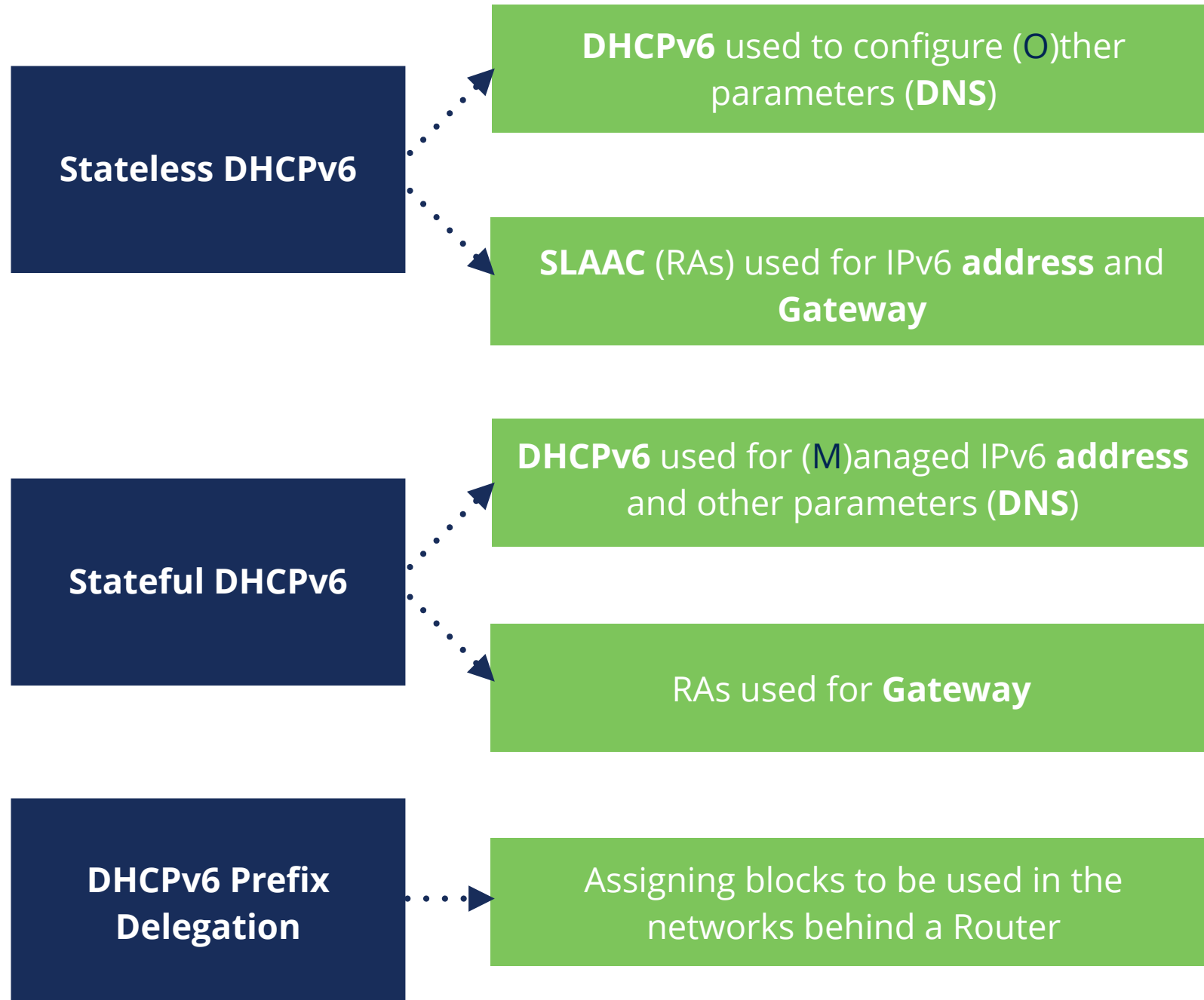


# DUID

- DHCP Unique Identifier
- A globally unique identifier used to identify the single machine/device
  - One DUID per DHCPv6 client
- DHCPv6 does not use only MAC address as identifier
- Variable length between 96 - 160 bits
  - Example Client DUID:  
00030001001A2F875602



# DHCPv6 Modes



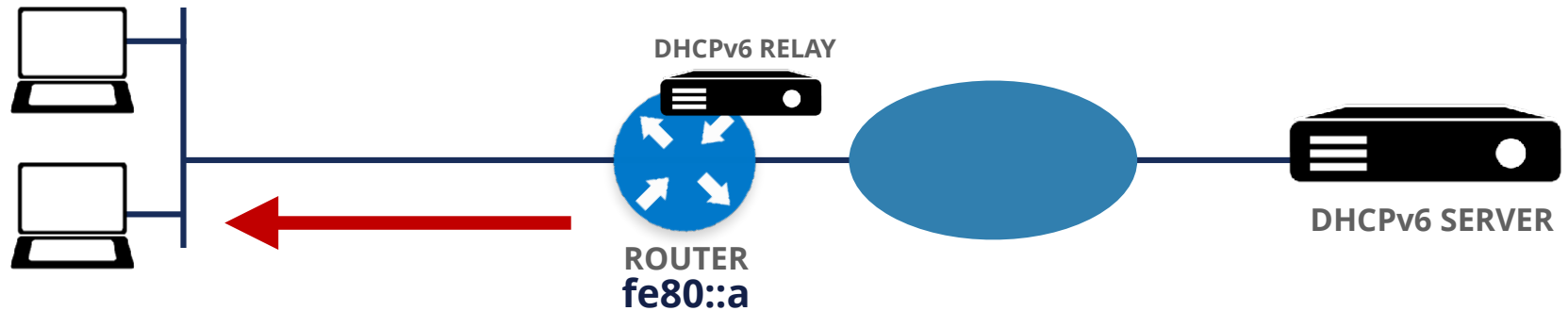


# Stateless DHCPv6

- Complements SLAAC configuration:
  - i.e. host obtain the address using SLAAC and the DNS server address from DHCPv6
  - In dual-stack networks we can obtain IPv4 DNS server addresses from DHCPv4
- Configure a DHCP pool with additional parameters:
  - DNS Server
  - Domain name
  - NTP
- Activated by “other configuration” flag in ND



# Stateless DHCPv6 Scenario

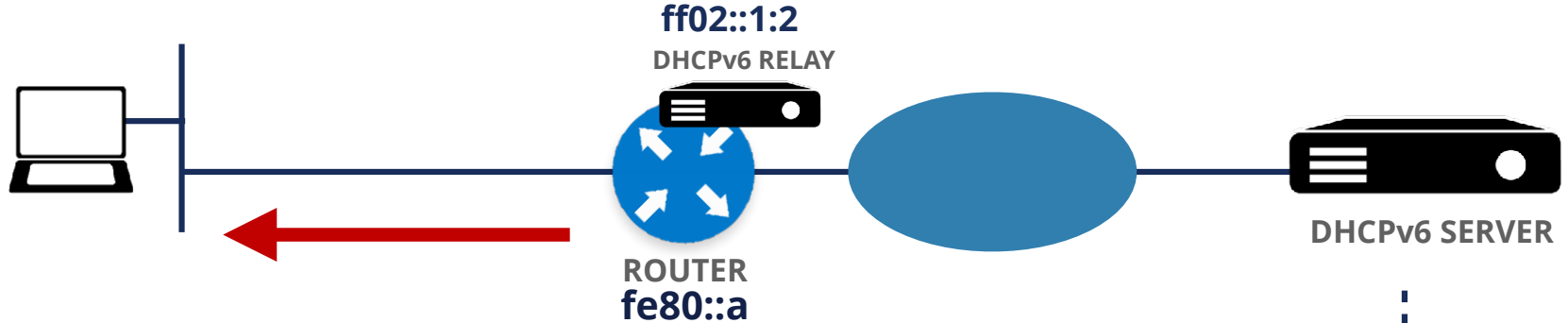


Router's link-local	
<b>Default Gateway</b>	<b>fe80::a</b>
<b>IPv6 Address</b>	
<b>2001:db8:1:1::/64</b>	<b>IID</b>
Prefix in RA	Generated by Host

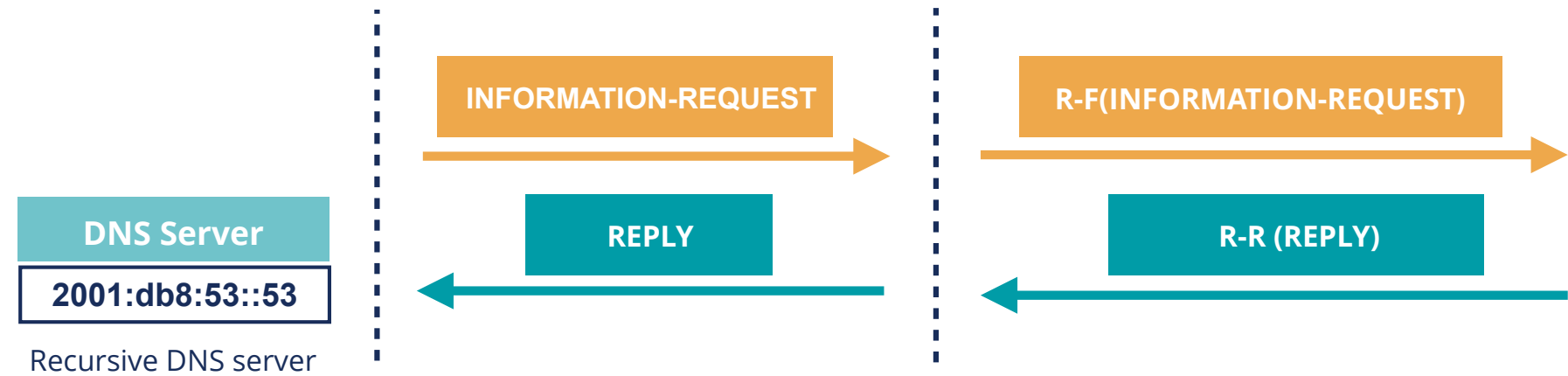
IPv6	ICMPv6 RA
IPv6.Source IPv6	fe80::a
IPv6.Destination IPv6	ff02::1
RA.M Flag	0
RA.O Flag	1
RA.Prefix Info. Option.Prefix	Prefix/64
RA.Prefix Info. Option.A Flag	1



# Stateless DHCPv6 Scenario



Router's link-local		IPv6	ICMPv6 RA
Default Gateway	fe80::a	IPv6.Source IPv6	fe80::a
IPv6 Address		IPv6.Destination IPv6	ff02::1
2001:db8:1:1::/64	IID	RA.M Flag	0
Prefix in RA	Generated by Host	RA.O Flag	1
		RA.Prefix Info. Option.Prefix	Prefix/64
		RA.Prefix Info. Option.A Flag	1





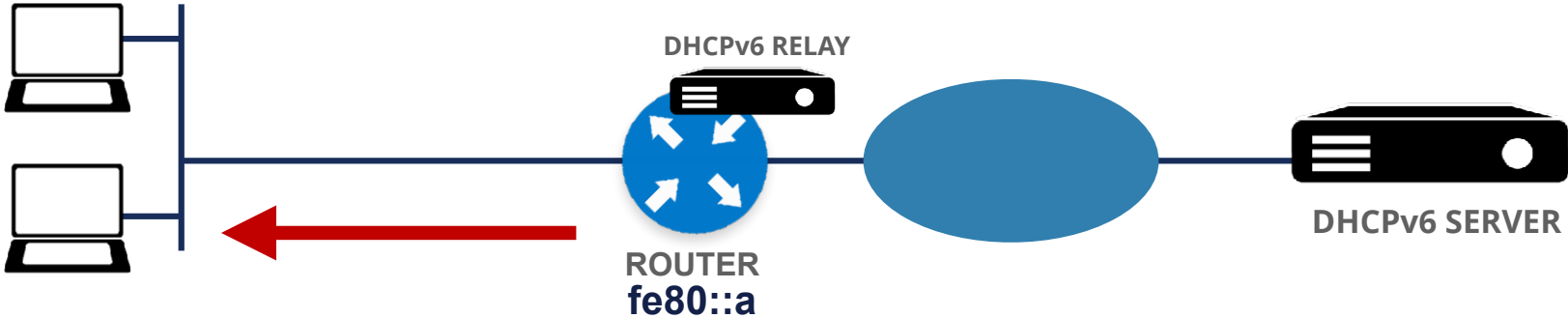
# Stateful DHCPv6

- Similar to DHCPv4
- A router can act as a DHCP server
- Configuration parameters include:
  - DHCP pool name
  - Prefix information
  - List of DNS servers
  - Addresses for clients
- Server holds the states of the clients





# Stateful DHCPv6 Scenario



Default Gateway

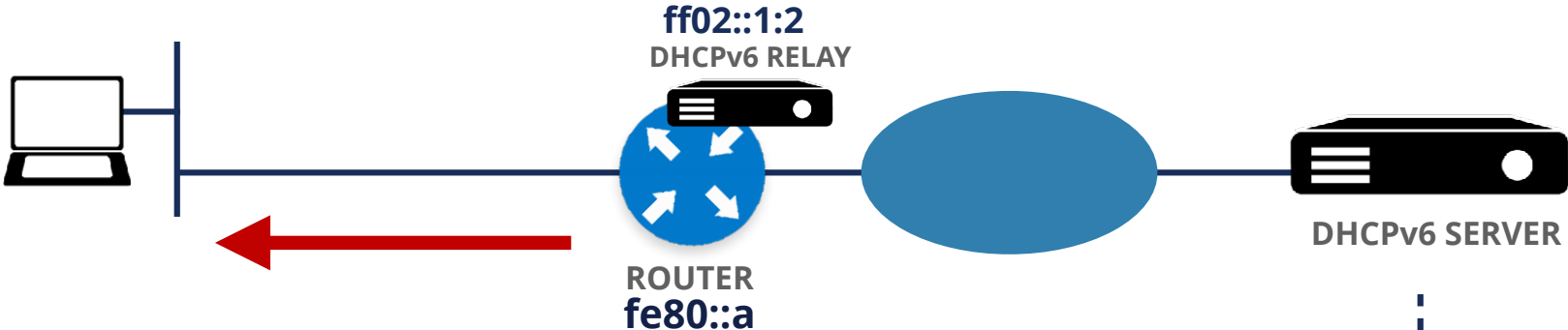
Router's link-local

**fe80::a**

IPv6	ICMPv6 RA
IPv6.Source IPv6	fe80::a
IPv6.Destination IPv6	ff02::1
RA.M Flag	<b>1</b>
RA.O Flag	<b>1 or 0</b>



# Stateful DHCPv6 Scenario



Default Gateway

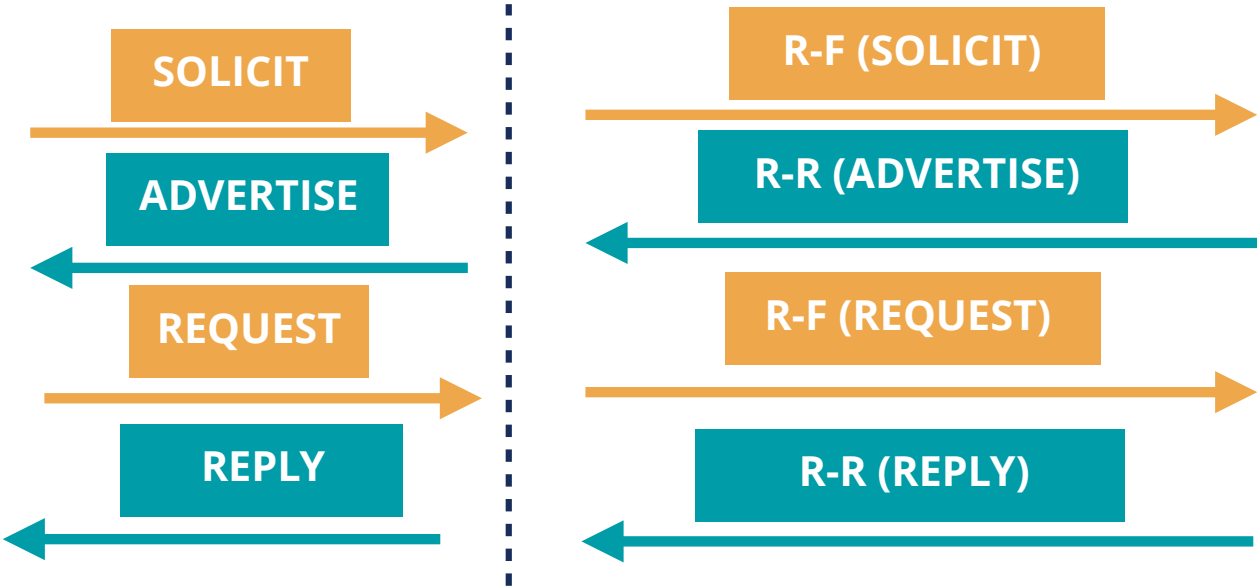
Router's link-local

fe80::a

IPv6	ICMPv6 RA
IPv6.Source IPv6	fe80::a
IPv6.Destination IPv6	ff02::1
RA.M Flag	1
RA.O Flag	1 or 0

IPv6 Address  
2001:db8:1:1::af12:b31/128

DNS Server  
2001:db8:53::53





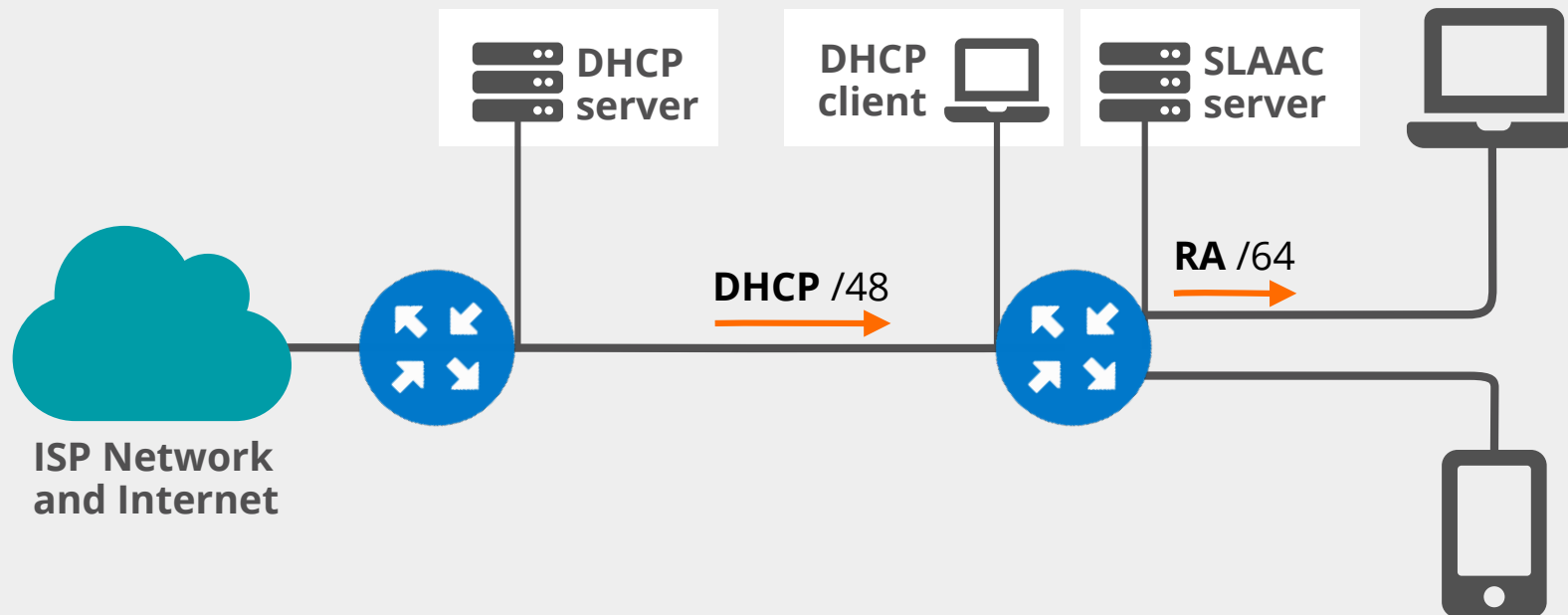
# DHCPv6 Prefix Delegation

- It is new and only needed for IPv6
- It is stateful
- IPv4 deployments:
  - ISP only has to deliver a public IPv4 address
  - NAT is used for translation using RFC1918
- IPv6 deployments:
  - IPv6 end-to-end reachability:
  - Home network gets its own IPv6 prefix (global addresses)
  - No NAT



# DHCPv6 Prefix Delegation

- ISP assigns a block of addresses for delegation to customers (e.g. /48)
- Customer assigns /64 prefixes to LAN interfaces



# DHCP Question

Why do we need DHCP-PD in IPv6?

- a. Since we need a mechanism for the automation of assigning GUAs to the devices on the subscriber LAN
- b. Since we need a mechanism for translation of the address used on the subscriber LAN
- c. Since we need to use NDP over Internet for the devices behind the CPE
- d. Since we need a low latency mechanism for IPv6 subscribers



# DHCP Answer

Why do we need DHCP-PD in IPv6?

- ✓ **Since we need a mechanism for the automation of assigning GUAs to the devices on the subscriber LAN**
- b. Since we need a mechanism for translation of the address used on the subscriber LAN
- c. Since we need to use NDP over Internet for the devices behind the CPE
- d. Since we need a low latency mechanism for IPv6 subscribers





# Questions



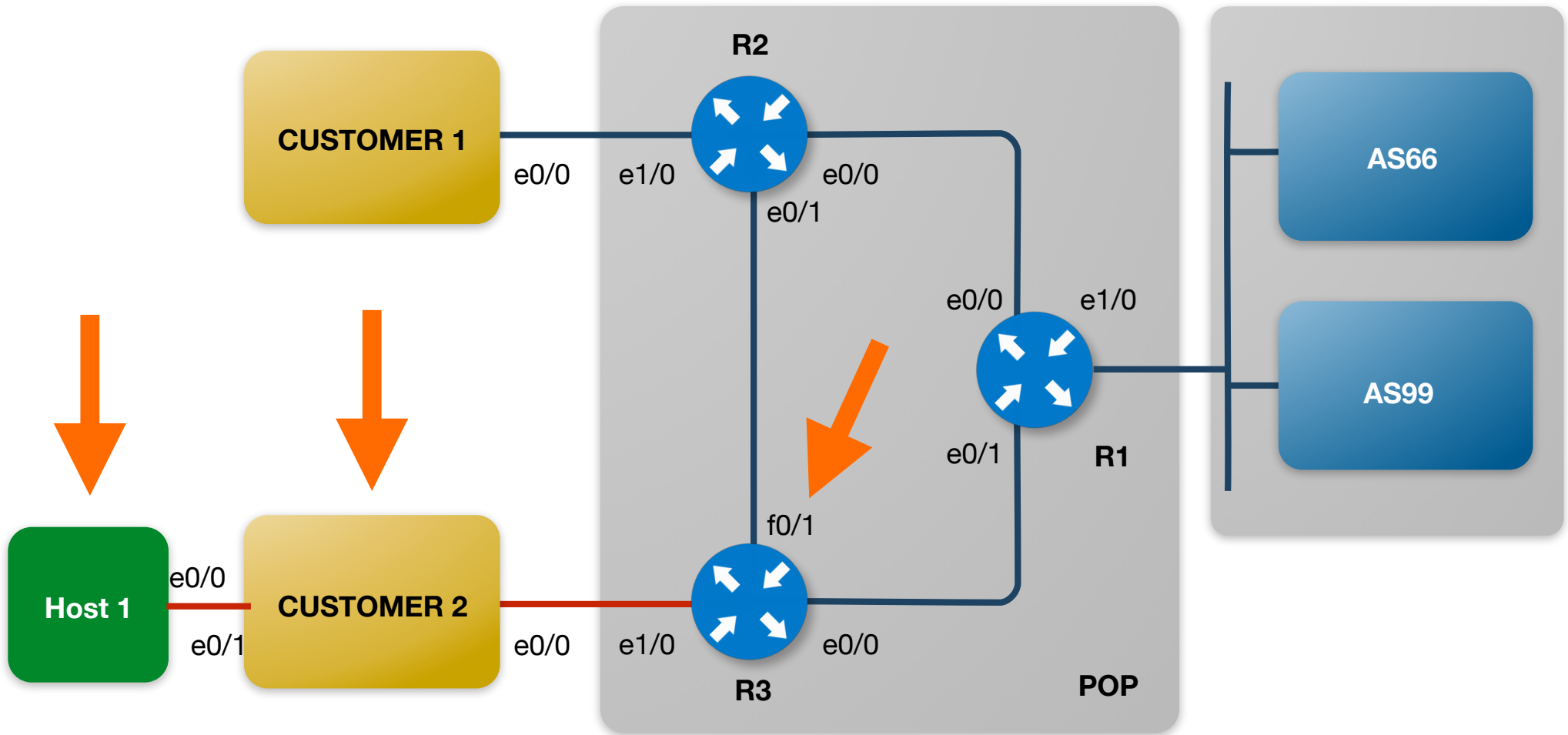


# DHCPv6-PD

## Exercise 15



# DHCPv6





# DHCPv6-PD Router Configuration

- DHCP pool named “DHCP\_CUSTOMERS” references local pool “DHCP\_POOL”
- DHCP\_POOL details about the address pool
- **On R3**

```
ipv6 dhcp pool DHCP_CUSTOMERS
  prefix-delegation pool DHCP_POOL
!
interface e1/0
  ipv6 address 2001:ffXX:0:ff02::b/64
  ipv6 dhcp server DHCP_CUSTOMERS
  no shutdown
!
ipv6 local pool DHCP_POOL 2001:ffXX:ff02::/48 56
```



# DHCPv6-PD C2 Configuration

- ISP facing interface is the DHCP client
- LAN facing interface is the IPv6 router sending RA message

```
interface e0/0
  ipv6 address autoconfig default
  ipv6 dhcp client pd PREFIX
  no shutdown
!
interface e0/1
  ipv6 address PREFIX ::1:b00c:caf3:bab3:1/64
  no ipv6 redirects
  no shutdown
```



# On H1

- Enable IPv6 on the router
  - You know how
- And enable SLAAC

```
interface e0/0
  ipv6 address autoconfig default
  no shutdown
```



# Summary

- We have now distributed an IPv6 prefix to Customer 2
- Customer has distributed prefixes to LAN interfaces automatically and distributed SLAAC to the host
  
- Can you find the DUID address of C2 ?



**DHCPv6-PD**  
**with static assignment**  
15-b - Exercise

# DHCPv6-PD with Static Assignment



- Assign to Customer 2 static prefix:
  - 2001:ffXX:ff02:AB00::/56



# On R3

- Find the DUID of the customer

```
#show ipv6 dhcp binding
```

- We have to create a pool, set-up prefix delegation and specify the DUID

```
(config)#ipv6 dhcp pool DHCP_CUSTOMERS  
(config-dhcpv6)#prefix-delegation 2001:ffXX:ff02:AB00::/56 DUID
```





# On C2

- To make sure the changes we made are propagated, we shut and no shut the interface towards R3

```
(config)# interface e0/0  
(config-if) shut  
(config-if) no shut
```

- It can take some time to propagate, up to 4 minutes

```
# show ipv6 interface brief
```



# Security

16 - Section

# IPv6 Security - Why Does It Matter?

- Most operating systems have IPv6 enabled by default nowadays
- IPv6 is present in your IPv4-only networks
  - tunnels
  - autoconfiguration on hosts
- The default IPv6 policies will not be what you need
- Often everything open



# Subnet Scanning

- In IPv6, scanning the whole address space is not possible anymore, but people use:
  - words (dead, beef, babe, cafe)
  - lower numbers (::1, ::2, ::3)
  - IPv4 based addresses (2001:db8::192:168:1:1/128)



# Subnet Scanning

- Scanning multicast addresses
  - ff02::1 - all hosts
  - ff05::5 - all DHCP servers
  - ff05::2 - all routers
- You can use easy to remember addresses, but remember that scanning will work the same as in IPv4



# ICMPv6

- ICMPv6 is used to report errors, ping and discover others (Neighbor Discovery)
- ICMPv6 is an integral part of IPv6
- Disabling ICMPv6 will break your network



# Firewall Filtering and ICMPv6

- IPv6 border filter example

Action	Src	Dst	ICMPv6 type	ICMPv6 Code	Name
Permit	Any	A	128	0	echo reply
Permit	Any	A	129	0	echo request
Permit	Any	A	1	0	no route to dest
Permit	Any	A	2	0	packet too big
Permit	Any	A	3	0	TTL exceeded
Permit	Any	A	4	0	parameter problem



# IPv6 Headers

- In IPv6, the header of a packet can be extended
- Extension headers are used for routing, fragmentation, IPSEC, etc.
- Some Intrusion Detection Systems find it hard to figure out where layer 4 starts and the extension header ends





# IPSec

- IPSec in IPv6 is the same as in IPv4
- There is nothing automatically secure in IPv6
- IPSec should be supported in IPv6
  - PKI infrastructure costs time and money



# RA Guard

- RFC7113
- Implement on a L2 switch, so they can filter out rogue or misconfigured routers sending router advertisements
- Filtering based on:
  - MAC address
  - Port where the RA was received
  - IP source address



# Hosts

- Hosts can get an IPv6 address unnoticed
- Hosts can set up tunnels
- Keep software up-to-date
- Host security controls should inspect IPv4 and IPv6
  - Firewalls
  - VPN clients



# Routers

- Protect vty lines

```
ipv6 access-list line-vty-in  
  remark company management prefix  
  permit ipv6 2001:db8:0:1::/64 any
```

```
line vty 0 15  
  ipv6 access-class line-vty-in in
```

- Use a /127 for point-to-point links if possible



# IPv6 Bogons

- Documentation prefix
  - 2001:db8::/32
- 6bone
  - 3ffe::/16
  - Returned to the IANA pool
- Cymru bogon list (very long!)
  - Also available as BGP feed
  - <https://www.team-cymru.org/Services/Bogons/fullbogons-ipv6.txt>



# Questions





# IP Address Management

Section 17



# Why IP Address Management?

- How do you currently keep track?
  - There are many subnets in IPv6
  - Your spreadsheet might not scale
  - And you want to take care of DNS/reverse DNS
- There are 524288 /48s in a /29
- That is 34359738368 /64s!





# Benefits of Using IPAM

- Repository of assignable IP addresses stays up-to-date
- Prevents duplicates in the networks
- Assists troubleshooting
- For regulatory or legal compliance



# What Are the Functionalities?

- IP search functionality
- IP address/Device discovery
- Allocations & making reservations
- Connection to RIR DBs (pull and/or push information)
- DNS and DHCP integration - for example, DNS can update itself when it learns the IP address a client has obtained via DHCP
- Task creation and assignment (optional)
- Multiple access profile support ( Admin, Operator, User/Read-only)
- Import/Export of files



# What Should You Compare?

- Is it for “data center/web service management” or “small/medium enterprise”?
- Is it easy to learn?
- Cleaner graphical user interface
- Paid vs Free - Customer support?
- Licensed vs Open-source
- Native VRF support
- File Import/Export support



# Address Management

- There are many open source IPv6 IPAM tools
  - NetDot
  - GestióIP
  - phpIPAM
  - Netbox
  - NIPAP
  - NOCProject



# NetDot

{net.} NETWORK DOCUMENTATION TOOL

netdot.foobar.org

Management | Contacts | Cable Plant | Advanced | Reports | Export | Help

Devices | Assets | VLANs | Address Space | DNS Records | DNS Zones | DHCP

### Address Space Tasks

**Search**

IP[/prefix]:  Search

IP regex:  Search

Keywords:  Search

---

[\*]: 2001:db8::/32

Children | Sites | Zones | Access Rights | Attributes | Comments | All

Address: 2001:db8::/32	Owner
Status: <b>Container</b>	Used by
Description: Santa's Global Network	Netmask
First Seen 2014-06-03 12:31:31	Broadcast
Last Seen 2014-06-03 12:31:31	Usable Addresses
RIR: RIPE	Address Utilization
ASN: 64512	Space Allocated

### Tree View

```
graph LR; A["2001:db8::/32  
Santa's Global Network"] --- B["2001:db8::/64  
Santa's BGP Loopbacks"]; A --- C["2001:db8:1::/48  
North Pole POP"]; A --- D["2001:db8:2::/48  
South Pole POP"]; C --- E["2001:db8:1:1::/64  
Polar Bear assignment"]; C --- F["2001:db8:1:2::/64  
Arctic Fox Internet"]; D --- G["2001:db8:2:1::/64  
Penguins International"]; D --- H["2001:db8:2:2::/64  
McMurdo Station"];
```

- Device discovery via SNMP
- DNS and DHCP configuration management
- IP&MAC address tracking
- BGP peer and AS Number tracking



# NetDot

{net.} NETWORK DOCUMENTATION TOOL

netdot.foobar.org

Management | Contacts | Cable Plant | Advanced | Reports | Export | Help

Devices | Assets | VLANs | Address Space | DNS Records | DNS Zones | DHCP

### Address Space Tasks

**Search**

IP[/prefix]:

IP regex:

Keywords:

---

[\*]: 2001:db8::/32

Children | Sites | Zones | Access Rights | Attributes | Comments | All

Address: 2001:db8::/32	Owner
Status: <b>Container</b>	Used by
Description: Santa's Global Network	Netmask
First Seen 2014-06-03 12:31:31	Broadcast
Last Seen 2014-06-03 12:31:31	Usable Addresses
RIR: RIPE	Address Utilization
ASN: 64512	Space Allocated

### Tree View

```
graph LR; A["2001:db8::/32  
Santa's Global Network"] --- B["2001:db8::/64  
Santa's BGP Loopbacks"]; A --- C["2001:db8:1::/64  
Polar Bear assignment"]; C --- D["2001:db8:1:1::/64  
Arctic Fox Internet"]; C --- E["2001:db8:1:2::/64  
Penguins International"]; F["2001:db8:2::/64  
South Pole POP"] --- G["2001:db8:2:1::/64  
Penguins International"]; F --- H["2001:db8:2:2::/64  
McMurdo Station"];
```

- Multi-level user access: Admin, Operator, User
- Export scripts for
  - Nagios
  - Cacti
  - RANCID

# GestióIP



network search

networks VLANs sites lines CM import/export

networks

ALL  
10.0.0.0/8  
192.168.0.0/16

v4 site category show rootnets show endpoints

network	BM	description	site	category
<input type="checkbox"/> 10.0.1.0	29	sync FWs	Lon1	prod
<input type="checkbox"/> 10.0.1.8	29	sync LBs	Lon1	prod
<input type="checkbox"/> 172.17.0.0	16			
<input type="checkbox"/> 192.168.0.0	24	frontends	Lon1	prod
<input type="checkbox"/> 192.168.1.0	25	backends	Lon1	prod
<input type="checkbox"/> 192.168.1.128	25	frontends II	Lon1	prod
<input type="checkbox"/> 192.168.2.0	24	application server	Lon1	prod
<input type="checkbox"/> 192.168.3.0	24	Vips-int	Lon1	prod
<input type="checkbox"/> 192.168.4.0	24	management	Lon1	prod
<input type="checkbox"/> 192.168.5.0	26		Lon1	prod
<input type="checkbox"/> 192.168.8.0	24	backup	Lon1	prod
<input type="checkbox"/> 192.168.30.0	24	frontends pre	Lon1	pre
<input type="checkbox"/> 192.168.31.0	25	backends pre	Lon1	pre
<input type="checkbox"/> 192.168.32.0	25	255.255.255.128 - 126 hosts	Lon1	pre
<input type="checkbox"/> 192.168.40.0	25	fontends dev	Lon1	dev
<input type="checkbox"/> 192.168.40.128	25	backends dev	Lon1	dev
<input type="checkbox"/> 192.168.50.0	24	administrators	BCN	corp
<input type="checkbox"/> 192.168.51.0	24	developers	BCN	corp
<input type="checkbox"/> 192.168.52.0	24	marketing	BCN	corp
<input type="checkbox"/> 192.168.53.0	25	printer	BCN	corp
<input type="checkbox"/> 192.168.54.0	23	wifi	BCN	corp

select all unselect all

- Web based IPAM, easy usage
- Based on Surfnet document
- Shows free ranges
- VLAN management system
- Host discovery via SNMP and DNS



# GestiÓIP

The screenshot shows the GestiÓIP web interface. At the top, there is a navigation bar with a search box labeled 'network search' and several menu items: 'networks', 'VLANs', 'sites', 'lines', 'CM', and 'import/export'. Below the navigation bar, the main content area is titled 'networks'. On the left side, there is a sidebar with a tree view showing a hierarchy of IP ranges: 'ALL', '10.0.0.0/8', and '192.168.0.0/16'. The main table displays a list of networks with columns for 'network', 'BM', 'description', 'site', and 'category'. Each row includes a checkbox and a small progress indicator. A tooltip is visible over the '192.168.32.0' row, showing '255.255.255.128 - 126 hosts'. At the bottom of the table, there are links for 'select all' and 'unselect all'.

network	BM	description	site	category
<input type="checkbox"/> 10.0.1.0	29	sync FWs	Lon1	prod
<input type="checkbox"/> 10.0.1.8	29	sync LBs	Lon1	prod
<input type="checkbox"/> 172.17.0.0	16			
<input type="checkbox"/> 192.168.0.0	24	frontends	Lon1	prod
<input type="checkbox"/> 192.168.1.0	25	backends	Lon1	prod
<input type="checkbox"/> 192.168.1.128	25	frontends II	Lon1	prod
<input type="checkbox"/> 192.168.2.0	24	application server	Lon1	prod
<input type="checkbox"/> 192.168.3.0	24	Vips-int	Lon1	prod
<input type="checkbox"/> 192.168.4.0	24	management	Lon1	prod
<input type="checkbox"/> 192.168.5.0	26		Lon1	prod
<input type="checkbox"/> 192.168.8.0	24	backup	Lon1	prod
<input type="checkbox"/> 192.168.30.0	24	frontends pre	Lon1	pre
<input type="checkbox"/> 192.168.31.0	25	backends pre	Lon1	pre
<input type="checkbox"/> 192.168.32.0	255.255.255.128 - 126 hosts		Lon1	pre
<input type="checkbox"/> 192.168.40.0	25	fontends dev	Lon1	dev
<input type="checkbox"/> 192.168.40.128	25	backends dev	Lon1	dev
<input type="checkbox"/> 192.168.50.0	24	administrators	BCN	corp
<input type="checkbox"/> 192.168.51.0	24	developers	BCN	corp
<input type="checkbox"/> 192.168.52.0	24	marketing	BCN	corp
<input type="checkbox"/> 192.168.53.0	25	printer	BCN	corp
<input type="checkbox"/> 192.168.54.0	23	wifi	BCN	corp

- Multi lingual
- DNS zone file generator for forward and reverse zones
  - Supports BIND and tinydns zone files
- Statistics about the ranges
- Offers subscription based cloud model





# phpIPAM

devel.phpipam.net/1.3/tools/devices/

{phpIPAM} / Tools / Devices

> Tools Subnets VLAN VRF **Devices** NAT PowerDNS DHCP Racks

Show all configured devices

List of network devices

Manage

Name	IP address	Description	Number of objects
CoreSwitch	10.10.10.254	Core switch 1	3 Objects
j4300-test	91.185.192.5		1 Objects
Test-switch	77.111.32.222		1 Objects
Wifi-1	10.10.20.245		0 Objects
Device not specified			1315 Objects

phpIPAM IP address management [v1.27] rev001 | In case of pro...

- AJAX based using jQuery libraries
  - PHP script, javascript and some HTML5/CSS3
  - Modern browser is preferred
- E-mail notifications
- Displays free ranges and numbers of clients

# phpIPAM



devel.phpipam.net/1.3/tools/devices/

{phpIPAM} / Tools / Devices

> Tools Subnets VLAN VRF **Devices** NAT PowerDNS DHCP Racks

Show all configured devices

List of network devices

Manage

Name	IP address	Description	Number of objects
CoreSwitch	10.10.10.254	Core switch 1	3 Objects
j4300-test	91.185.192.5		1 Objects
Test-switch	77.111.32.222		1 Objects
Wifi-1	10.10.20.245		0 Objects
Device not specified			1315 Objects

Open "devel.phpipam.net/1.3/tools/devices/" in a new tab | phpIPAM IP address management [v1.27] rev001 | In case of pro...

- Import and export to XLS files
- Can pull info from the RIPE DB
- VRF management/support



# NetBox

The screenshot shows the NetBox web interface. On the left is a navigation menu with items: Organization, Devices, Connections, Wireless, IPAM, Overlay, Virtualization, Circuits, Power, and Other. The main content area has a search bar at the top. Below it are three summary cards: 'Organization' (30 Sites, 12 Tenants), 'Inventory' (43 Racks, 22 Device Types, 88 Devices), and 'Wireless' (1 Wireless LANs, 0 Wireless Links). To the right of these cards is a vertical sidebar with links for VRFs, Aggregat, Prefixes, IP Range, IP Address, VLANs, Power Pa, and Power Fe. At the bottom is a table of recent activity:

ID	Time	Username	Full Name	Action	Type
755	2022-08-26 14:22	admin	—	Created	DCIM > sit
754	2022-08-26 14:17	admin	—	Created	DCIM > de
753	2022-08-26 14:15	admin	—	Created	DCIM > mi

- Comprehensive tool for network management including IPAM functionality
- IPAM and DCIM (Datacenter Infrastructure Management)
- Written in Python



# NetBox

The screenshot shows the NetBox web application interface. On the left is a navigation menu with items: Organization, Devices, Connections, Wireless, IPAM, Overlay, Virtualization, Circuits, Power, and Other. The main content area features a search bar at the top right. Below it are three summary cards: 'Organization' (30 Sites, 12 Tenants), 'Inventory' (43 Racks, 22 Device Types, 88 Devices), and 'Wireless' (1 Wireless LANs, 0 Wireless Links). To the right of these cards is a vertical sidebar with links for VRFs, Aggregat, Prefixes, IP Range, IP Address, VLANs, Power Pa, and Power Fe. At the bottom, a log table displays recent actions:

ID	Time	Username	Full Name	Action	Type
755	2022-08-26 14:22	admin	—	Created	DCIM > sit
754	2022-08-26 14:17	admin	—	Created	DCIM > de
753	2022-08-26 14:15	admin	—	Created	DCIM > mi

- Web application with PostgreSQL database
- No network monitoring, DNS, RADIUS, Config management



# NIPAP

IP NIPAP

127.0.0.1:5000/prefix/list#query\_string=test&search\_opt\_parent=undefined&search\_opt\_child

VRF +

test

Query took 0.64 seconds.

VRF	Prefix	Order
No VRF	+ 1.0.0.0/8	[gear] R
	+ 1.0.0.0/16	[gear] R
	1.0.1.0/24	[gear] A
	- 1.0.5.0/24	[gear] A
	1.0.5.1/24	[gear] H
	1.0.5.2/24	[gear] H
	1.0.5.3/24	[gear] H
	1.0.5.4/24	[gear] H
	1.0.5.5/24	[gear] H
	1.0.5.6/24	[gear] H
	1.0.5.7/24	[gear] H
	- 1.3.0.0/16	[gear] R
	1.3.0.0/24	[gear] A
	1.3.3.0/24	[gear] A
	2.0.1.0/24	[gear] A
	2.0.5.0/24	[gear] A
	2.0.6.0/24	[gear] A
	2.0.7.0/24	[gear] A
	2.0.8.0/24	[gear] A

- Web and CLI interface
- Native support for IPv6 (full feature parity with IPv4)
- Native VRF support - overlapping prefixes in different VRFs
- Very **powerful search function** (featuring regexp)



# NIPAP

IP NIPAP

127.0.0.1:5000/prefix/list#query\_string=test&search\_opt\_parent=undefined&search\_opt\_child

VRF +

test

Query took 0.64 seconds.

VRF	Prefix	Order
No VRF	+ 1.0.0.0/8	[gear] [R]
	+ 1.0.0.0/16	[gear] [R]
	1.0.1.0/24	[gear] [A]
	- 1.0.5.0/24	[gear] [A]
	1.0.5.1/24	[gear] [H]
	1.0.5.2/24	[gear] [H]
	1.0.5.3/24	[gear] [H]
	1.0.5.4/24	[gear] [H]
	1.0.5.5/24	[gear] [H]
	1.0.5.6/24	[gear] [H]
	1.0.5.7/24	[gear] [H]
	- 1.3.0.0/16	[gear] [R]
	1.3.0.0/24	[gear] [A]
	1.3.3.0/24	[gear] [A]
	2.0.1.0/24	[gear] [A]
	2.0.5.0/24	[gear] [A]
	2.0.6.0/24	[gear] [A]
	2.0.7.0/24	[gear] [A]
	2.0.8.0/24	[gear] [A]

- Statistics over used and free addresses
- Integrated audit log
- IP address request system for automatically assigning suitable prefixes



# NOC Project

**NOC: Unconfigured Installation**

Navigation

- Main
- Project Management
- Workflow
- GIS
- Inventory
- Service Activation
- Fault Management
- Performance Management
- Configuration Management
- Address Space Management
  - Assigned Addresses
  - Reports
  - Setup
  - VRFs
- Virtual Circuit Management
- DNS
- Peering Management
- Knowledge Base

Assigned Addresses

Начало > Address Space Management > VRF: default > IPv4 >

Navigation

VRF: default > IPv4 >

**0.0.0.0/0**

Root

Allocated Prefixes

Prefix	State
10.0.0.0/8	ALLOCATED
27.20.100.0/30	ALLOCATED
27.20.100.0/24	ALLOCATED
45.15.21.0/28	ALLOCATED
45.145.0.0/29	ALLOCATED
15.145.0.0/28	ALLOCATED
15.145.0.0/30	ALLOCATED
15.145.0.0/30	ALLOCATED
45.145.0.0/30	ALLOCATED

★ Favorites

- BSD licensed
- Complete OSS system
- Clean web interface
- DNS integration



# NOC Project

**NOC: Unconfigured Installation**

Navigation

- Main
- Project Management
- Workflow
- GIS
- Inventory
- Service Activation
- Fault Management
- Performance Management
- Configuration Management
- Address Space Management
  - Assigned Addresses
  - Reports
  - Setup
  - VRFs
- Virtual Circuit Management
- DNS
- Peering Management
- Knowledge Base

Assigned Addresses

Начало > Address Space Management > VRF: default > IPv4 >

Navigation

VRF: default > IPv4 >

**0.0.0.0/0**

Root

Allocated Prefixes

Prefix	State
10.0.0.0/8	ALLOCATED
27.20.100.0/30	ALLOCATED
27.20.100.0/24	ALLOCATED
45.15.21.0/28	ALLOCATED
45.149.0.0/29	ALLOCATED
15.149.21.0/28	ALLOCATED
15.149.21.0/30	ALLOCATED
15.149.21.0/30	ALLOCATED
45.149.21.0/30	ALLOCATED

★ Favorites

- Reporting tools
- Quick view options (free space)
- Hierarchical user groups
- Large developer team





# Questions





# What's Next in IPv6



## Webinars

**Attend another webinar live wherever you are.**

- ❖ Introduction to IPv6 (2 hrs)
- ❖ IPv6 Host Configuration (2 hrs)
- ❖ IPv6 Addressing Plan (1 hr)
- ❖ Basic IPv6 Protocol Security (2 hrs)
- ❖ IPv6 Associated Protocols (2 hrs)
- ❖ IPv6 Security Myths, Filtering and Tips (2 hrs)

↓ For more info  
click the link  
below



[learning.ripe.net](https://learning.ripe.net)



## Face-to-face

**Meet us at a location near you for a training session delivered in person.**

- ❖ Basic IPv6 (8.5 hrs)
- ❖ Advanced IPv6 (17 hrs)
- ❖ IPv6 Security (8.5 hrs)



## E-learning

**Learn at your own pace at our online Academy.**

- ❖ IPv6 Fundamentals (15 hrs)
- ❖ IPv6 Security (24 hrs)

↓ For more info  
click the link  
below



[academy.ripe.net](https://academy.ripe.net)



## Examinations

**Learnt everything you needed? Get certified!**

- ❖ IPv6 Fundamentals - Analyst
- ❖ IPv6 Security - Expert

↓ For more info  
click the link  
below



[getcertified.ripe.net](https://getcertified.ripe.net)



Learn something new today!  
**[academy.ripe.net](https://academy.ripe.net)**

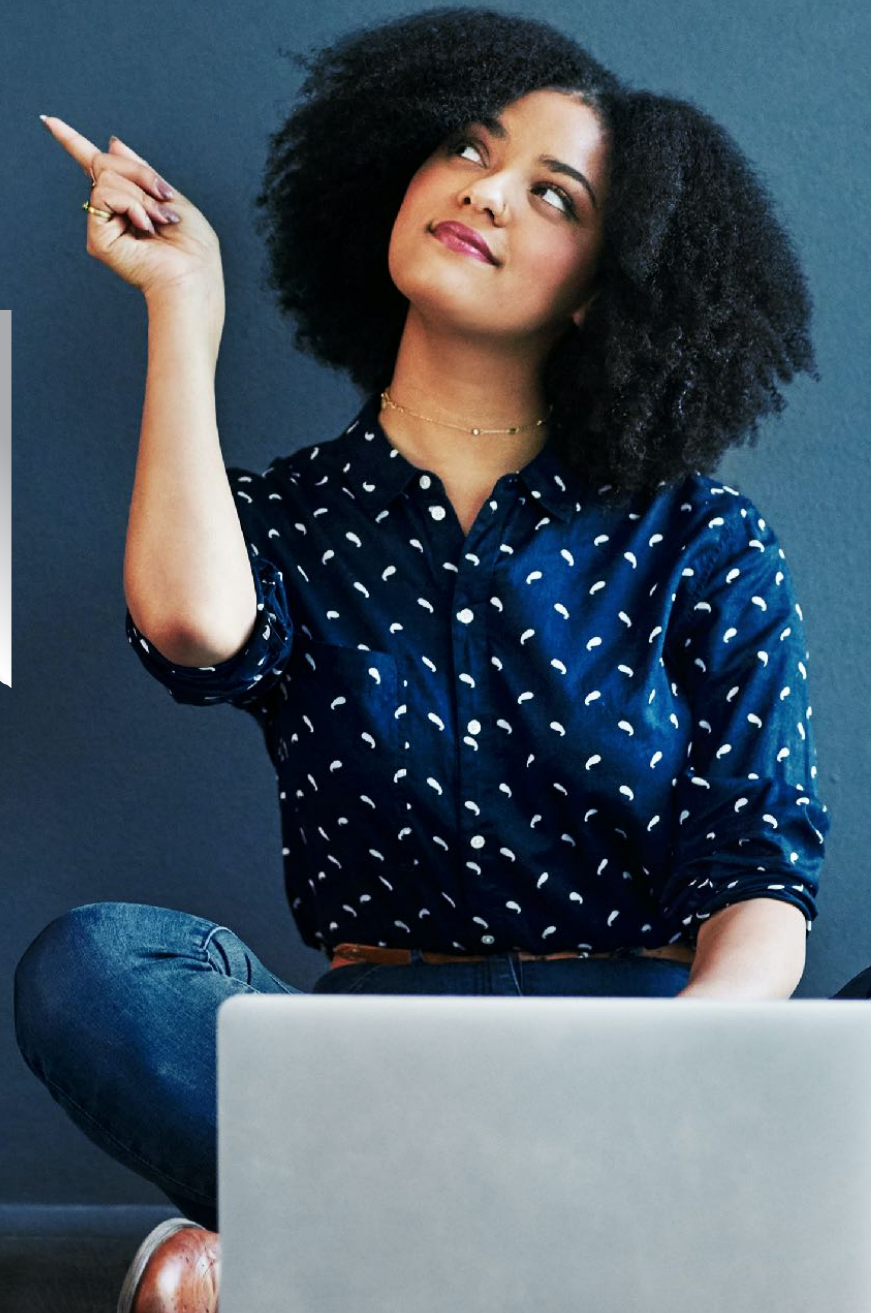




# RIPE NCC Certified Professionals



<https://getcertified.ripe.net/>





# We want your feedback!

What did you think about this session?

Take our survey at:

<https://www.ripe.net/feedback/av6/>



# Copyright Statement

[...]

The RIPE NCC Materials may be used for **private purposes, for public non-commercial purpose, for research, for educational or demonstration purposes**, or if the materials in question specifically state that use of the material is permissible, and provided the RIPE NCC Materials are not modified and are properly identified as RIPE NCC documents. Unless authorised by the RIPE NCC in writing, any use of the RIPE NCC Materials for advertising or marketing purposes is strictly forbidden and may be prosecuted. The RIPE NCC should be notified of any such activities or suspicions thereof.

[...]

Find the full copyright statement here:

<https://www.ripe.net/about-us/legal/copyright-statement>





# Questions



Änn Соңы An Críoch پايان Y Diwedd  
Vége Endir Finvezh Ende Koniec  
Son დასასრული ұтырғ Kінецъ Finis  
Lõpp Amaia תסה Tmiem  
Sfârșit Loppu Slutt Liðugt Kraj  
Kraj النهاية Конец Fund  
Fine Fin النهاية Fí Край Konec Τέλος  
Slut Einde Fim Pabaiga  
Beigas

E<sub>1</sub> N<sub>1</sub> D<sub>2</sub>