Deploying IPv6-mostly access networks
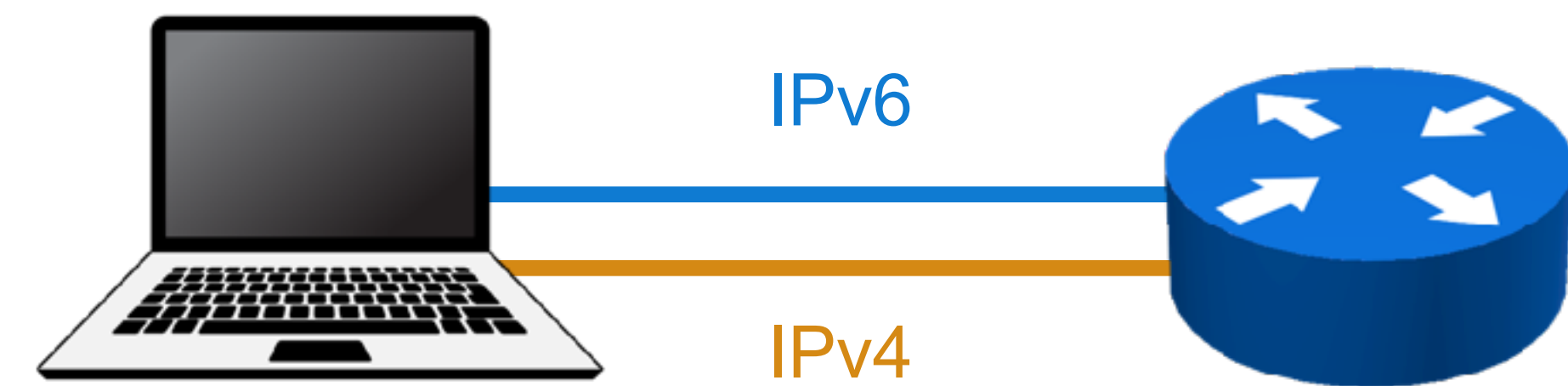
IPv6-only and dual stack in one network

Ondřej Caletka | 28 February 2023 | APNIC 55

The best transition mechanism

- IPv4-only and IPv6-only resources **directly accessible**

- IPv6 preferred for dual-stack resources

- Problems with IPv6 **masked** by Happy Eyeballs algorithm

- But it **does not address IPv4 scarcity**

Dual Stack

IPv6

IPv4

NAT64 allows IPv6-only networks

- IPv6 accessible natively

- IPv4 is translated into part of IPv6 address space

- Together with **DNS64**, everything seems to be **accessible over IPv6**

- **But sometimes you run into…**

  - IPv4 literals

  - Legacy software opening IPv4-only sockets

  - Dual-stack servers with broken IPv6

DNS64

IPv6 Internet

NAT64 Box

IPv4 Internet
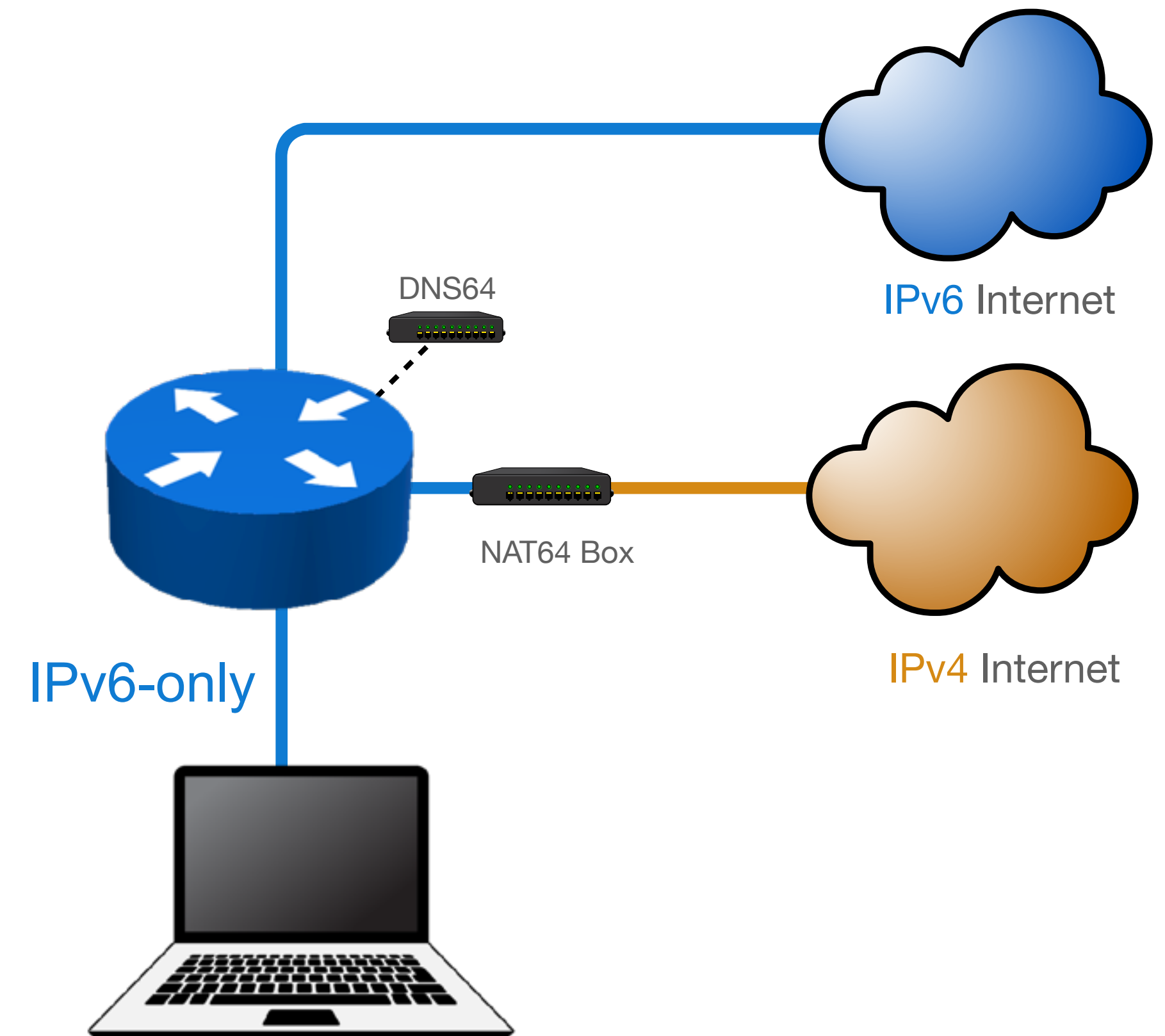
IPv6-only

Mobiles are ready

- Apple forces all iOS apps to work well on IPv6-only networks with NAT64

- There is Happy Eyeballs 2.0 for IPv4 literals or broken IPv6 on dual stack servers

- Finally CLAT is used for tethering to a computer

- Android uses just CLAT (464XLAT)

    - so IPv4 is accessible via two translations

DNS64

IPv6 Internet

NAT64 Box

IPv4 Internet
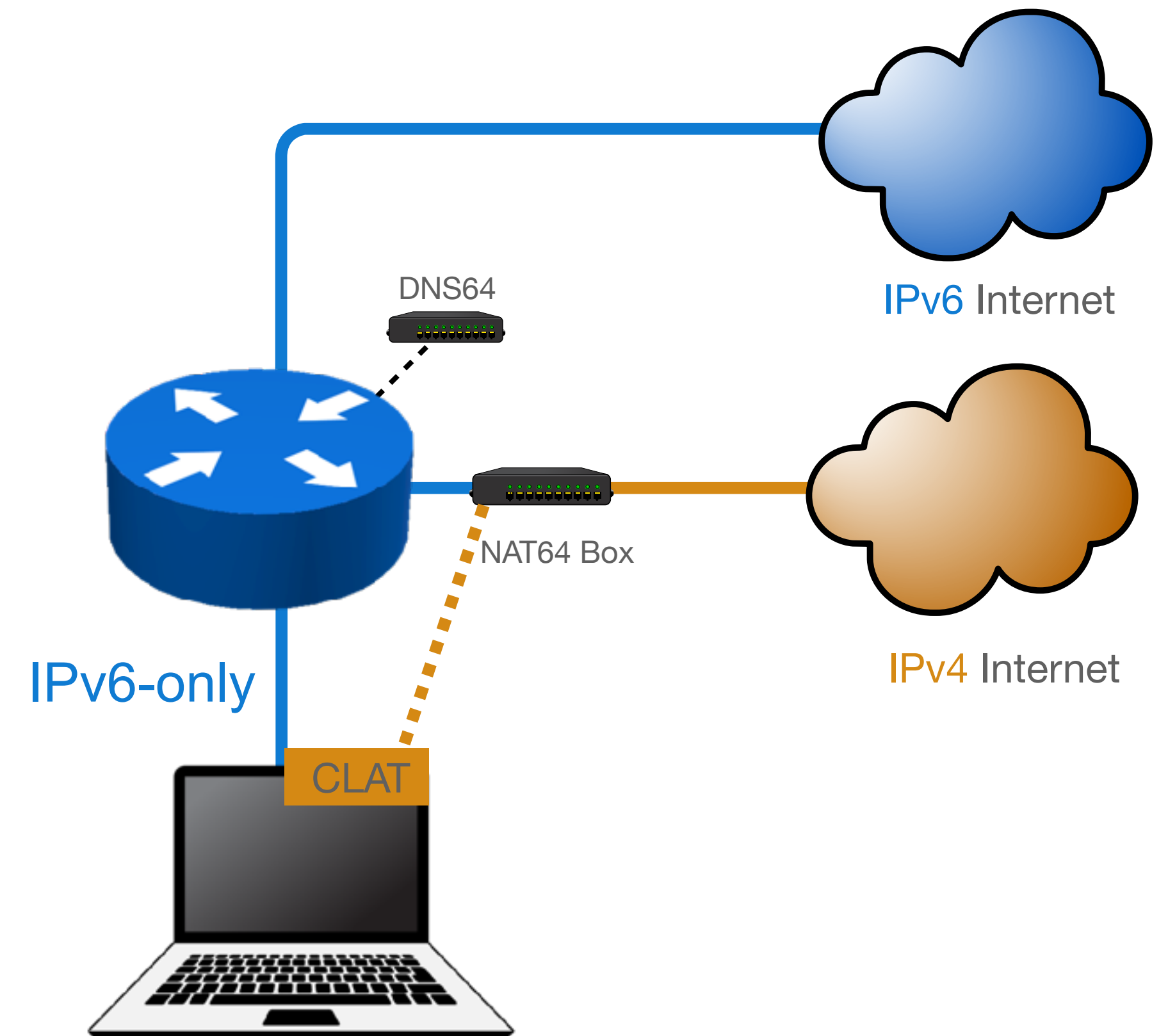
IPv6-only

CLAT

Desktops suffer on IPv6-only

- No Happy Eyeballs 2.0 implementation outside Apple

  - and even on Apple, only high-level APIs support it *(eg. Safari, not Chrome)*

- **No CLAT** in Windows, Linux or ChromeOS

- Well known small problems:

  - Legacy applications using IPv4-only sockets

  - IPv4 literals do not work

  - Dual-stack servers where IPv6 is broken do not work

  - Legacy Happy Eyeballs **doesn't help** since there's no IPv4 to fall back to

  - Most corporate VPNs do not work (often *just* a configuration issue)

Can we do IPv6-only?

At least for some devices…

IPv6-only Preferred option of DHCP

(RFC 8925)
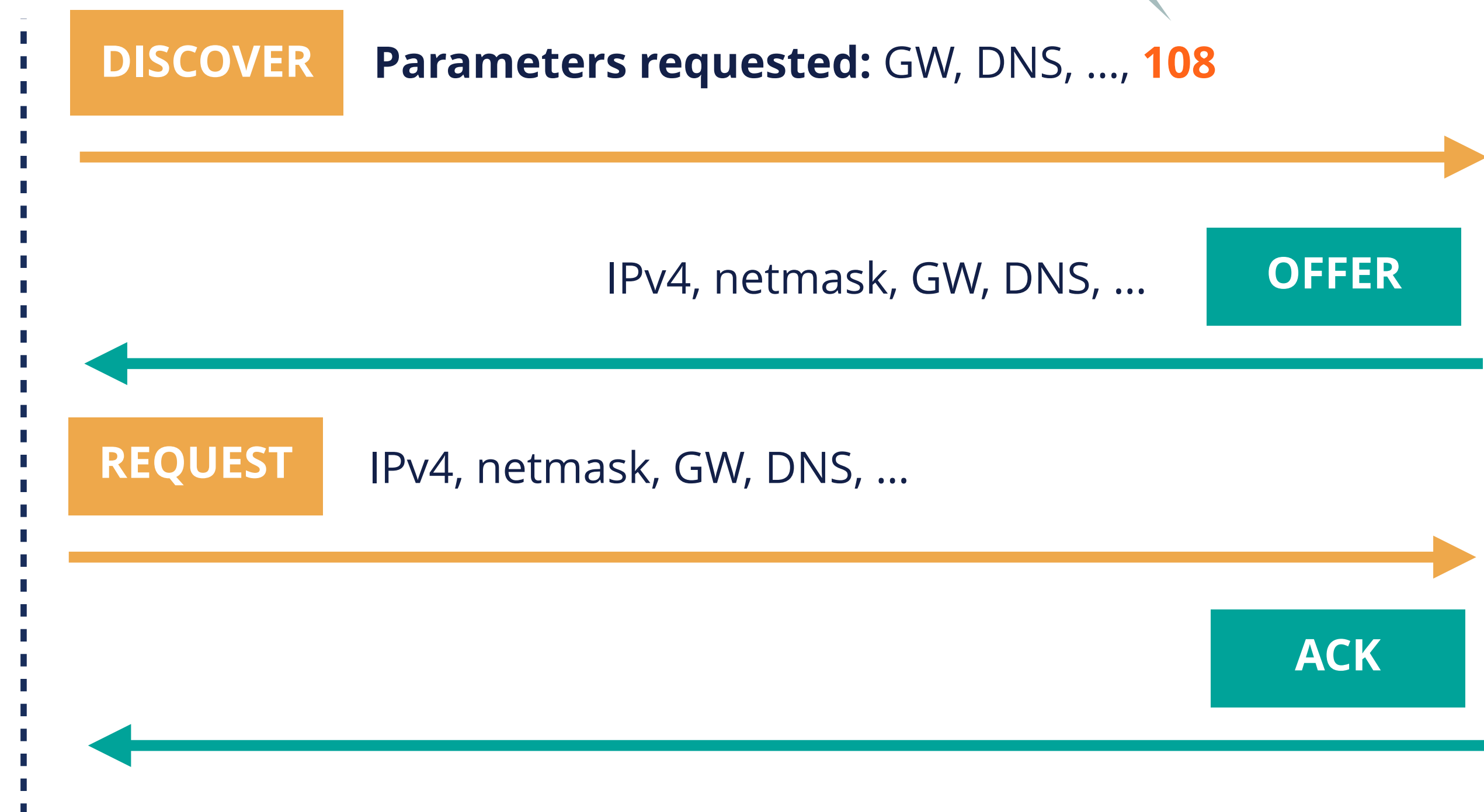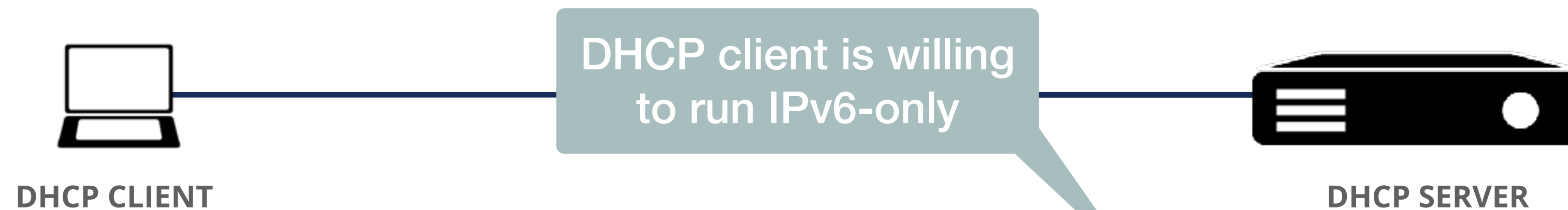
Using DHCP to turn IPv4 off

(RFC 8925)

DHCP CLIENT

DHCP SERVER

DISCOVER **Parameters requested:** GW, DNS, …, 108

IPv4, netmask, GW, DNS, …, **108: 30 minutes** OFFER

DHCP client aborts the transaction and waits 30 minutes

DHCP server is configured to prefer IPv6-only operation

Is DHCP option 108 already deployed?

You bet! Option 108 is requested by recent:

Android

iOS

macOS

34%   66%

● Requesting Option 108
● Not requesting

Unique MAC addresses measured during RIPE 84

Devices are ready, networks are lagging behind.

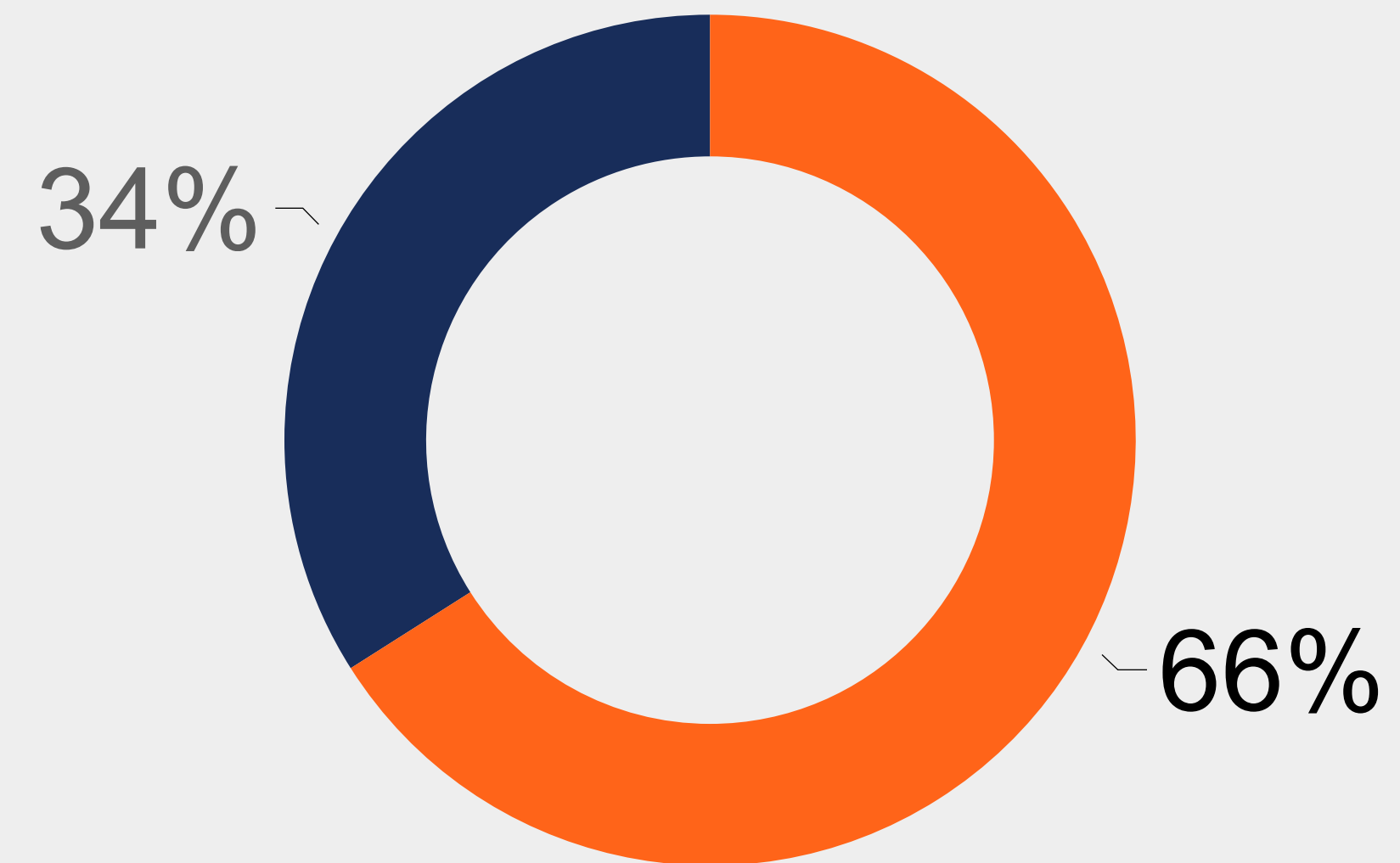# But what about macOS?

- It allows you to run *any* software including those using legacy IPv4-only APIs

- It turned out **there is CLAT in macOS** too!

  - On macOS 12, it gets activated by DHCP Option 108 **together with RA Option PREF64**

  - Since macOS 13, it gets **activated without any special requirements**

  - At the same time, **pure IPv6-only networks** (without NAT64) are not supported anymore

```
→  ~ ifconfig en0
en0: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=6463<RXCSUM,TXCSUM,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
    ether f0:18:98:31:36:c6
    inet6 fe80::1477:9fe8:a21d:56a6%en0 prefixlen 64 secured scopeid 0x6
    inet6 2a02:      :80:c48:6e99:5e6c:e453 prefixlen 64 autoconf secured
    inet6 2a02:      :80:392d:6ea9:e5fd:ddd1 prefixlen 64 autoconf temporary
    inet6 fdba:91fa:4142:80:813:d49b:cca9:9b87 prefixlen 64 autoconf secured
    inet 192.0.0.1 netmask 0xffffffff broadcast 192.0.0.1
    inet6 fdba:91fa:4142:80:fa:bf88:9a02:cbb1 prefixlen 64 clat46
    nat64 prefix 64:ff9b:: prefixlen 96
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
→  ~ ping -c 5 1.1.1.1
PING 1.1.1.1 (1.1.1.1): 56 data bytes
64 bytes from 1.1.1.1: icmp_seq=0 ttl=56 time=5.045 ms
64 bytes from 1.1.1.1: icmp_seq=1 ttl=56 time=10.375 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=56 time=11.156 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=56 time=10.977 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=56 time=10.280 ms

--- 1.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 5.045/9.567/11.156/2.286 ms
→  ~
```

PREF64 RA Option

- A Router Advertisement option **carrying NAT64 prefix**

- Needed for **CLAT configuration**, local DNS64 or Happy Eyeballs 2.0 *(dealing with IPv4 literals)*

- **Shares fate** with other configuration parameters

  - can be trusted **a bit more** than DNS64

- Supported by recent Android, iOS and macOS

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |      Scaled Lifetime    | PLC |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|               Highest 96 bits of the Prefix                   |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Running IPv6-mostly

DHCP option 108 is easy

- **Native support** in the latest Kea

- Most DHCP servers support defining **custom options**

  - for instance: `dnsmasq -O 108,0:0:1:2c`

  - the option value represents duration for which the IPv4 stack should be disabled

- **No special processing** on the DHCP server side is *required*

- But there **have to be free addresses** in the IPv4 address pool

  - Otherwise the DHCP server will not respond

PREF64 RA option is harder

- No **custom RA option** support in routers

    - We already **had this issue** with Recursive DNS Server option, now we **have it again**

    - Router vendors should really implement **custom options** similar to DHCP

- Adoption is *slowly* increasing:

    - `radvd` (merged but unreleased)

    - FRR (pull request pending)

    - `odhcpd` (pull request pending)

    - rad (part of OpenBSD)

    - MikroTik RouterOS v7.8 beta2

Surprises on macOS

If there are multiple network prefixes, CLAT picks up a single address from a **random one**, without considering ULA or deprecated prefixes


```
→   ~ ifconfig en0
en0: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=6463<RXCSUM,TXCSUM,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether f0:18:98:31:36:c6
        inet6 fe80::1477:9fe8:a21d:56a6%en0 prefixlen 64 secured scopeid 0x6
        inet6 2a02:          :80:c48:6e99:5e6c:e453 prefixlen 64 autoconf secured
        inet6 2a02:          :80:392d:6ea9:e5fd:ddd1 prefixlen 64 autoconf temporary
        inet6 fdba:91fa:4142:80:813:d49b:cca9:9b87 prefixlen 64 autoconf secured
        inet 192.0.0.1 netmask 0xffffffff broadcast 192.0.0.1
        inet6 fdba:91fa:4142:80:fa:bf88:9a02:cbb1 prefixlen 64 clat46
        nat64 prefix 64:ff9b:: prefixlen 96
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

Surprises on macOS

If user sets up a **custom IPv4 DNS server address**, DNS will not work, despite commands like `host` working normally

```
➜  ~ scutil --dns | head
DNS configuration

resolver #1
  search domain[0] : mtg.ripe.net
  nameserver[0] : 1.1.1.1
  flags    : Request A records, Request AAAA records
  reach    : 0x00000002 (Reachable)

resolver #2
  domain   : local
➜  ~ host google.com
google.com has address 172.217.168.238
google.com has IPv6 address 2a00:1450:400e:811::200e
google.com mail is handled by 10 smtp.google.com.
➜  ~ ping google.com
ping: cannot resolve google.com: Unknown host
➜  ~ ▮
```

Summary

Pros

- **Only one network** to join

- **No waste of IPv4** addresses for every single device
  - Cool if you **don't use NAT**

- Even for dual-stack clients, the usage of IPv4 is **minimal**
  - DNS64 will force all IPv6-capable applications to use NAT64 instead of native IPv4

Cons

- **Most complex** network setup

- IPv4 still **has to be deployed**

- NAT64 is **needed**

- **Problematic** interoperability between dual-stack and IPv6-only hosts within the network
  - Setting up a Chromecast from an Android phone is *impossible*

When to consider IPv6-mostly

- You **don't use NAT** and your DHCP pool is filling up

- You do use NAT but are **running out** of private addresses

- There are mostly **mobile** or **Apple devices** in your network

- You **already have NAT64** in place and want to gradually undeploy IPv4

RIPE 85 Meeting network experience

- Three networks deployed on the venue:
  - Main: **IPv6-mostly**
  - NAT64: **IPv6-only**
  - Legacy: **dual-stack**

- **74 %** of devices in the main network were **running IPv6-only**

- Biggest issue: **custom DNS servers** or **disabled IPv6** on a Mac

- Some Apple users rather connected to the legacy network

- Only observed issues with Cisco AnyConnect / OpenConnect VPN

- Networked printer (Lexmark CS510de) printed without issues

Questions

Ondrej.Caletka@ripe.net
@ripencc