RIPE NCC
RIPE NETWORK COORDINATION CENTRE

Deploying IPv6-mostly access networks

IPv6-only and dual stack in one network

Ondřej Caletka | 30 September 2022 | NLNOG Day

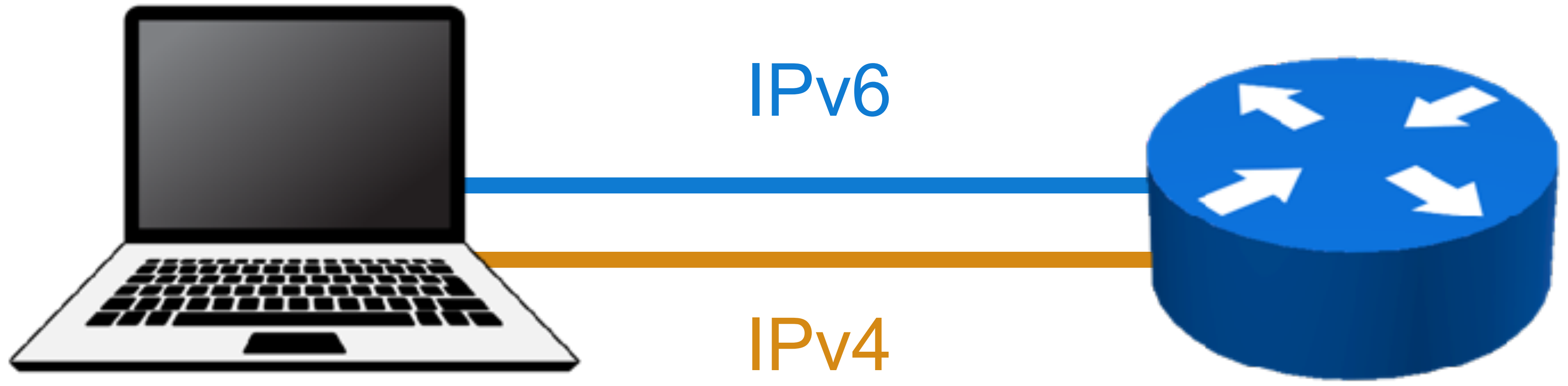
The endless transition to IPv6

- IPv6 is *slowly* being deployed
- IPv4 is still the protocol of the Internet
- There are simply **not enough IPv4 addresses**
  - repurposing 240/4, 127/8 or 0/8 will not help
- There are many transition mechanisms, two of which are special:
  - **Dual stack:** Running both protocols at the same time
  - **NAT64:** Allowing limited access from unmodified IPv6-only hosts to IPv4 resources

The best transition mechanism

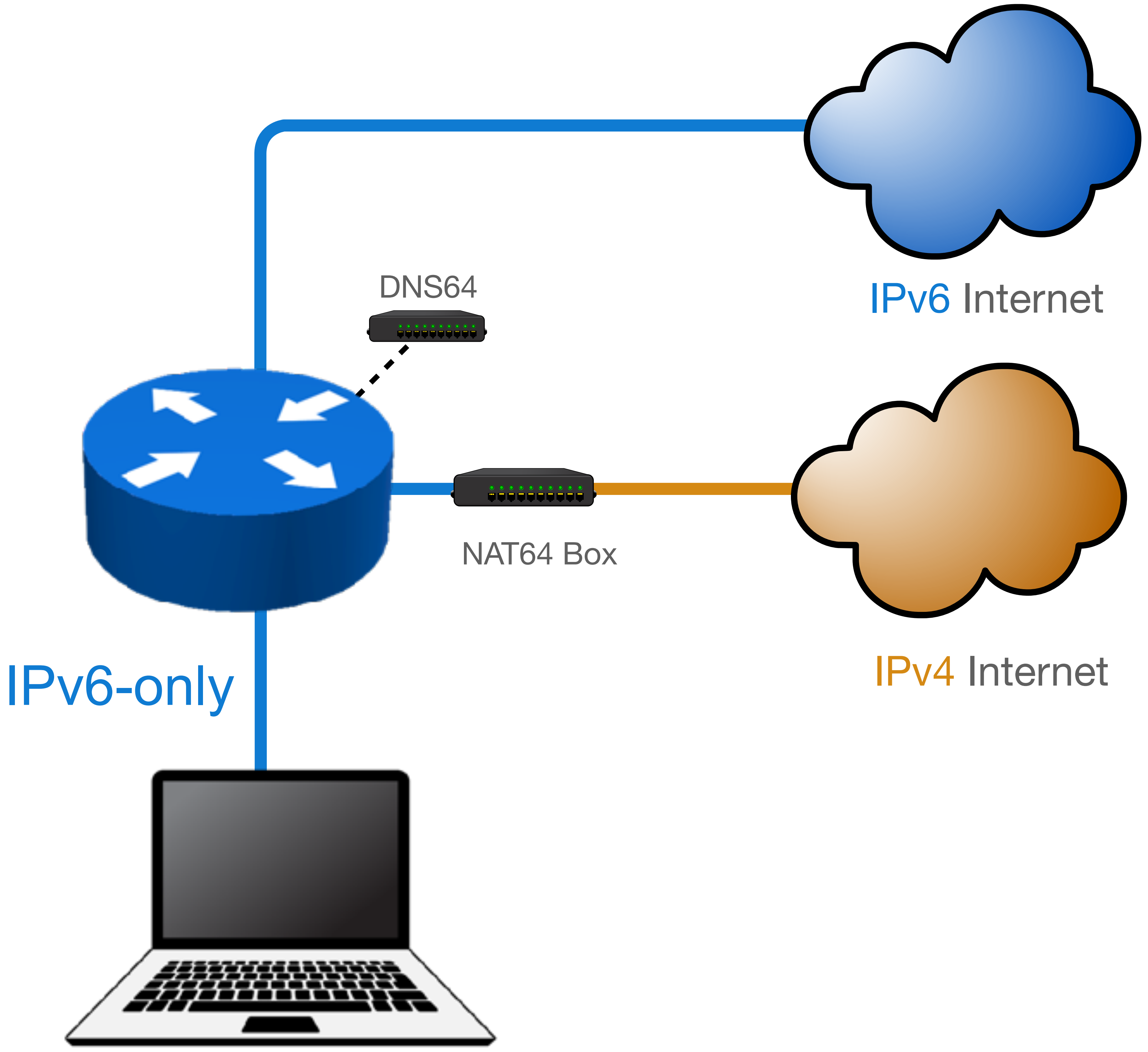
- IPv4-only and IPv6-only resources directly accessible
- IPv6 preferred for dual-stack resources
- Problems with IPv6 masked by Happy Eyeballs algorithm
- But it **does not address IPv4 scarcity**

Dual Stack



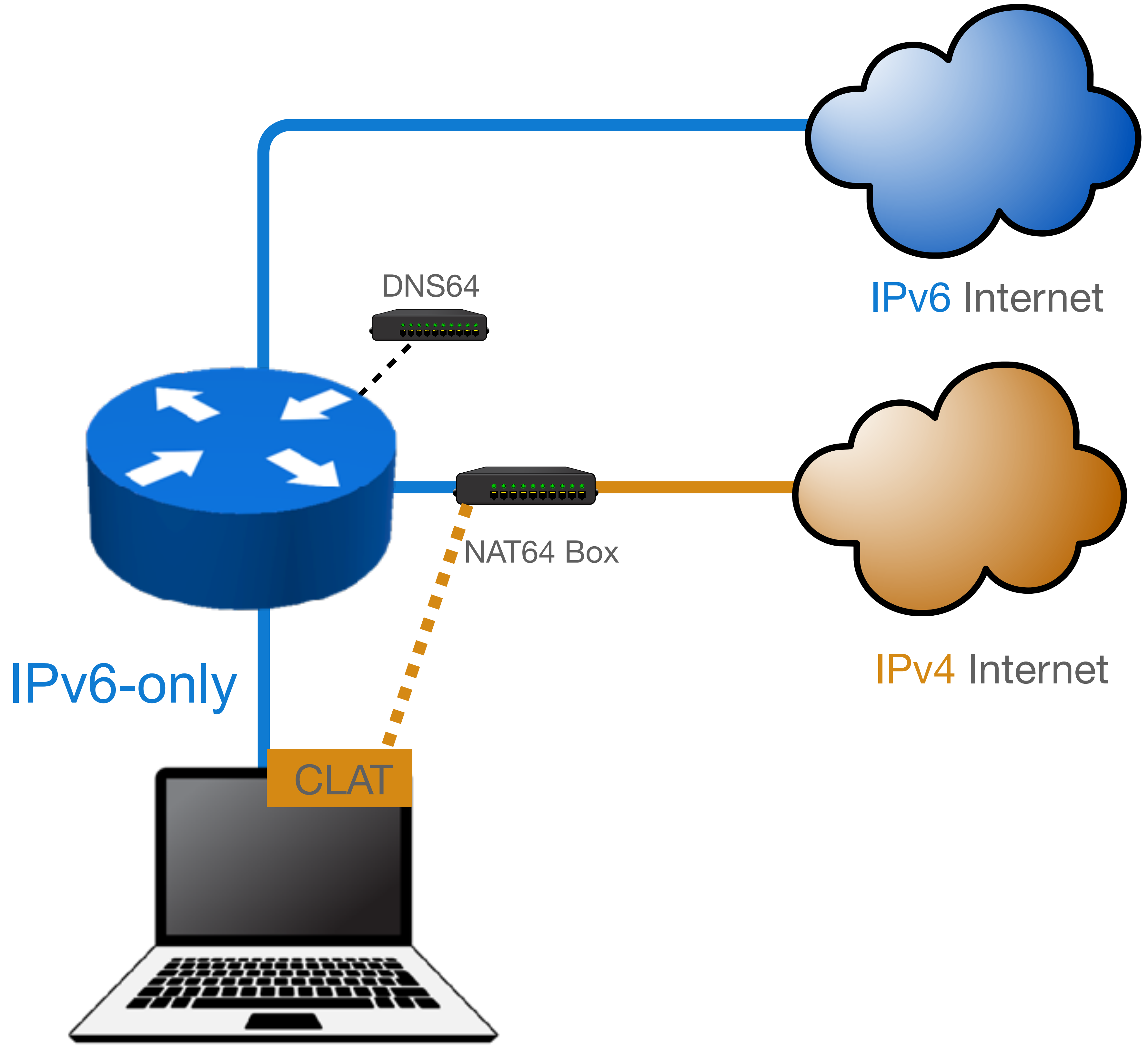
NAT64 allows IPv6-only networks

- 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



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



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
- Famous problems known for years:
  - Spotify desktop app does not work
  - IPv4 literals do not work
  - Dual-stack servers with broken IPv6 do not work
  - Legacy Happy Eyeballs doesn’t help since there's no IPv4 to fall back to
  - Most corporate VPNs will not work (often just a configuration issue)

Can we do IPv6-only?

At least for some devices…

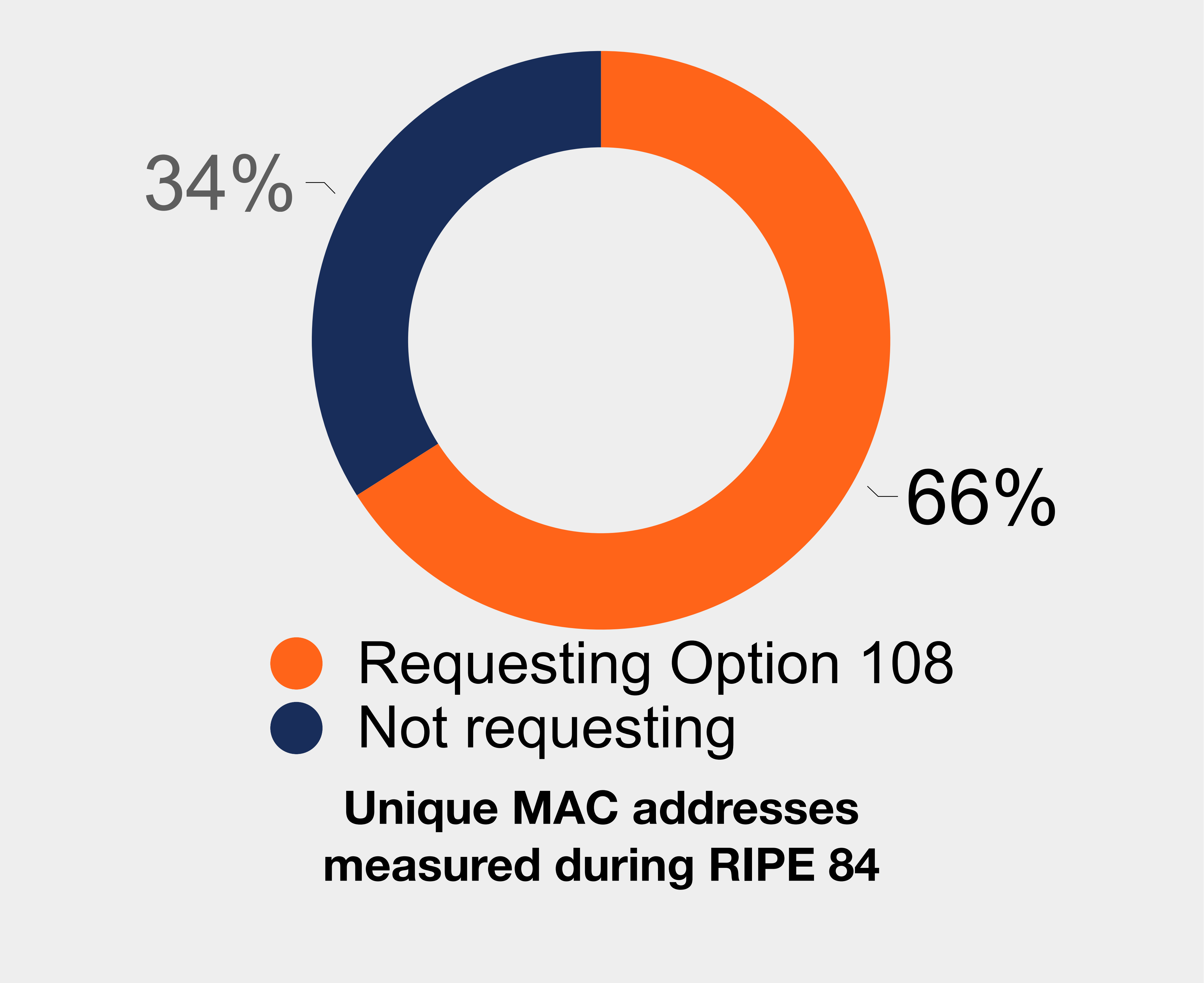
Signalling IPv6-only capability

- New DHCP(v4) option number 108: **IPv6-only Preferred** (RFC 8925)
  - Requested by DHCP clients of devices **capable of running IPv6-only**
  - Offered by DHCP servers for networks that **support IPv6-only** operation
  - When offered by DHCP server, the client will **deactivate IPv4 stack**
  - If not **requested by client** or not **offered by server** DHCP handshake continues normally
- IPv6-only capable devices can **opt-out from IPv4**
- Legacy devices **keep using dual stack**
- Users are not required to select proper network based on device capability (they will always prefer dual stack as it has no downsides for them)

Is DHCP option 108 already deployed?

You bet! Option 108 is requested by recent:

- Android
- iOS
- **macOS**

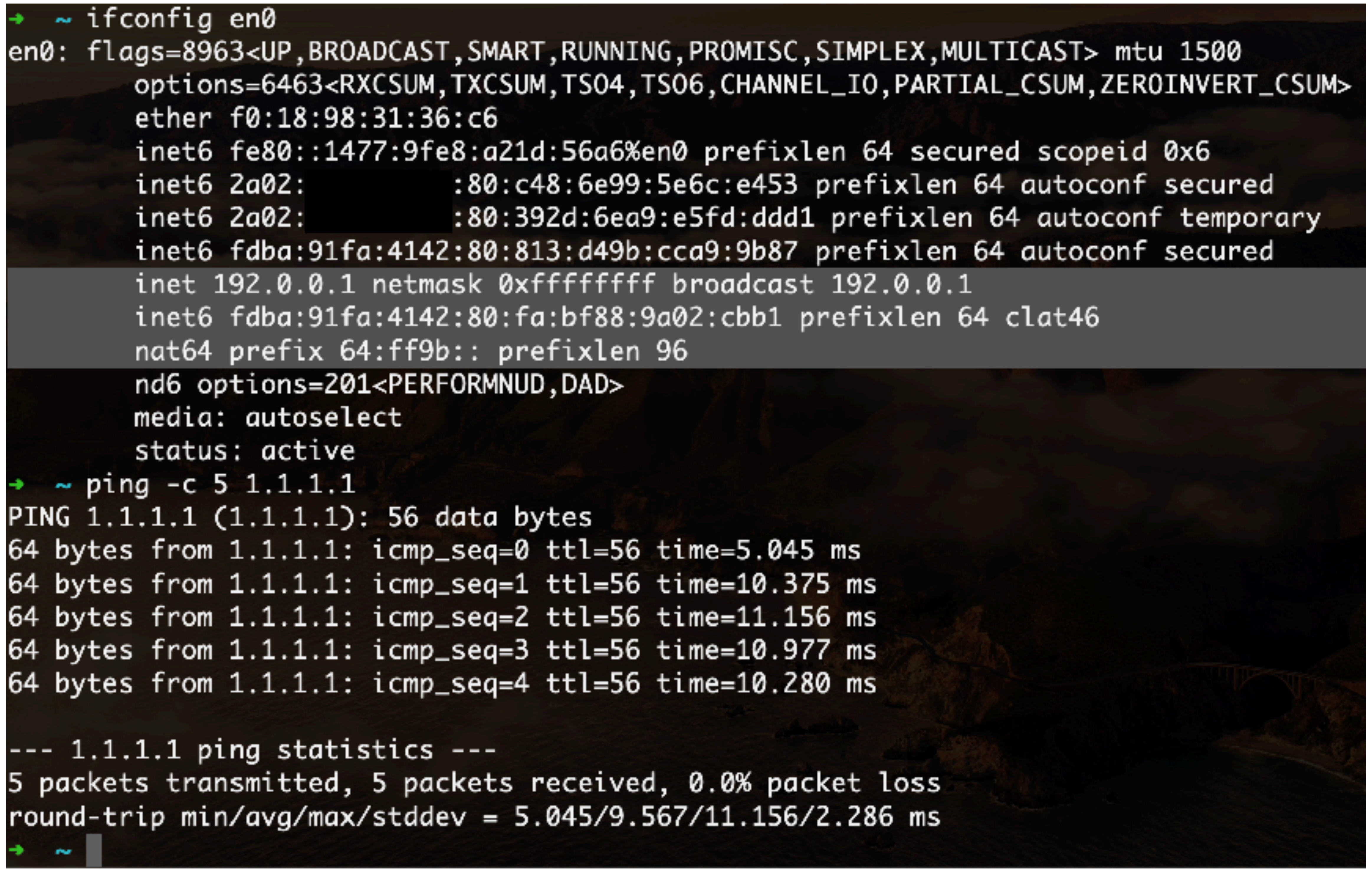


Unique MAC addresses measured during RIPE 84

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 (Spotify for Desktop, for instance)
- Pure IPv6-only would break such applications
- It turned out **there is CLAT in macOS** too!
  - It gets activated by DHCP Option 108 **together with RA Option PREF64**



```
➜  ~ 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:[redacted]: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 synthesis or Happy Eyeballs 2.0 (dealing with IPv4 literals)
- Replaces NAT64 prefix discovery using DNS64 query for `ipv4only.arpa` (RFC 7050)
- Shares fate with other configuration parameters
  - can be trusted **a bit more** than DNS64
- Supported by recent Android, iOS or 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                    |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NAT64 / PREF64 / DNS64 / IPv4

- **PREF64 is sufficient** to setup CLAT on all platforms
- In theory, DNS64 should be optional
  - This would force all IPv4 to go through the CLAT
- In practice, you **have to use DNS64** for Safari and iOS
  - When DHCP option 108 is received, Safari and most iOS apps refuse to use any IPv4
  - Without DNS64, **IPv4 internet is invisible** to them
  - On iOS, CLAT is used mostly for VoWiFi and perhaps for tethering
- You still need IPv4 and DHCP(v4)
  - For legacy devices and to trigger CLAT on Apple devices
  - The DHCP pool can be smaller, though

Running IPv6-mostly

DHCP option 108 is easy

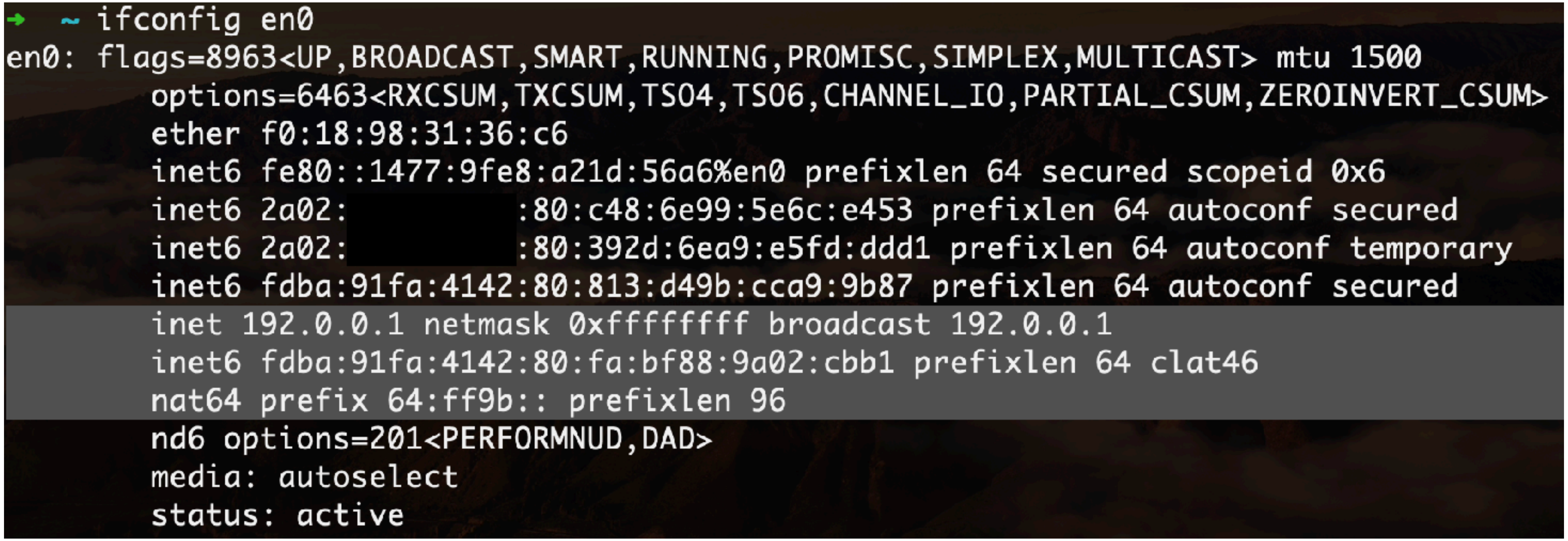
- **Native support** in the latest Kea
- Most DHCP servers support defining **custom options**
  - for instance: `dnsmasq -O 108,0:0:1:2c`
  - the option value represents duration for which the IPv4 stack should be disabled
- No special processing on the DHCP server side is *required*
- But there **have 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
- There are patches for some software routers:
  - `radvd` (merged but unreleased)
  - `FRR` (pull request pending)
  - `odhcpd` (pull request pending)

Surprises on macOS

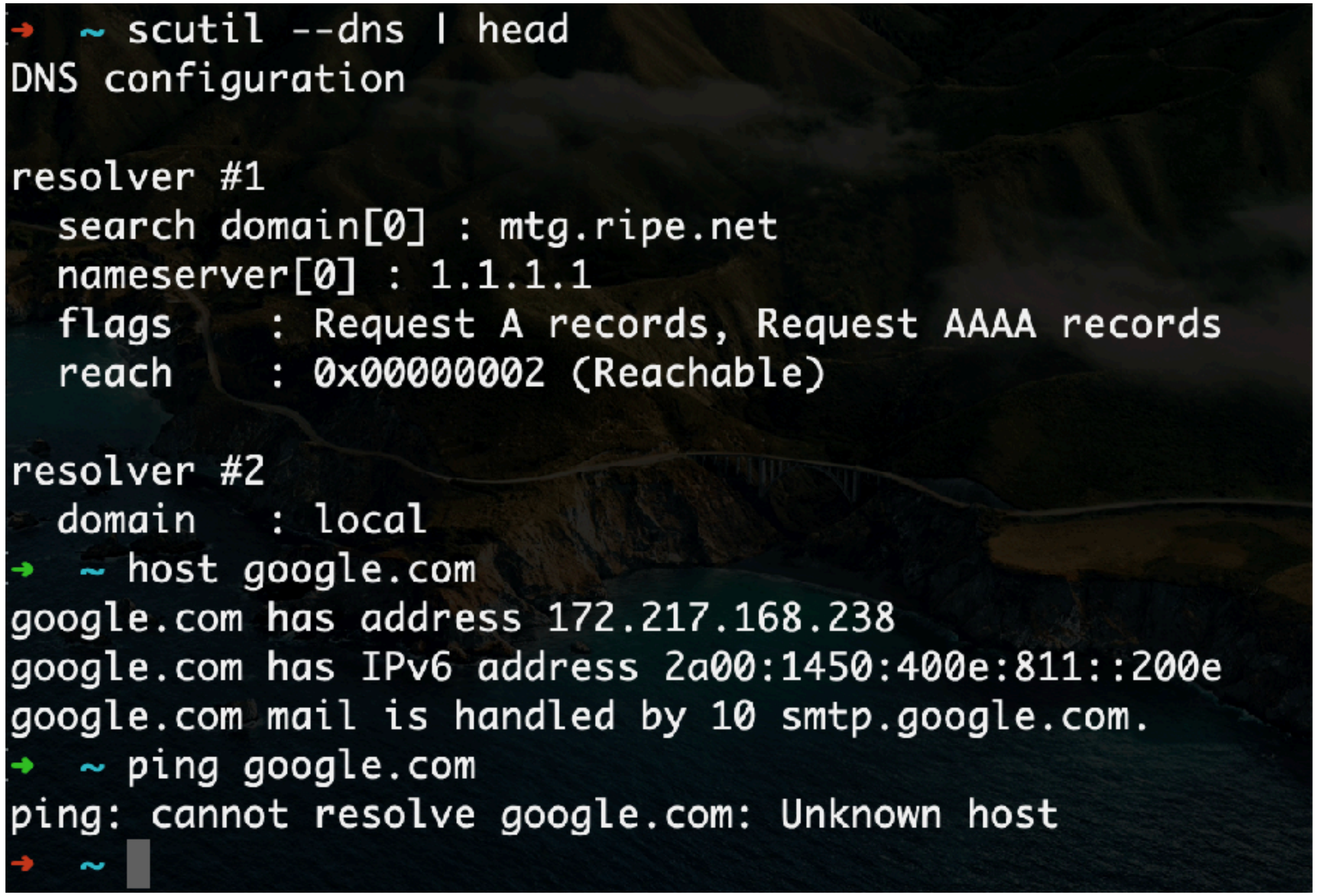
If there are multiple network prefixes, CLAT picks up a single address from a **random one**, without considering ULA or deprecated prefixes



```
➜  ~ 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:[redacted]:80:c48:6e99:5e6c:e453 prefixlen 64 autoconf secured
        inet6 2a02:[redacted]: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



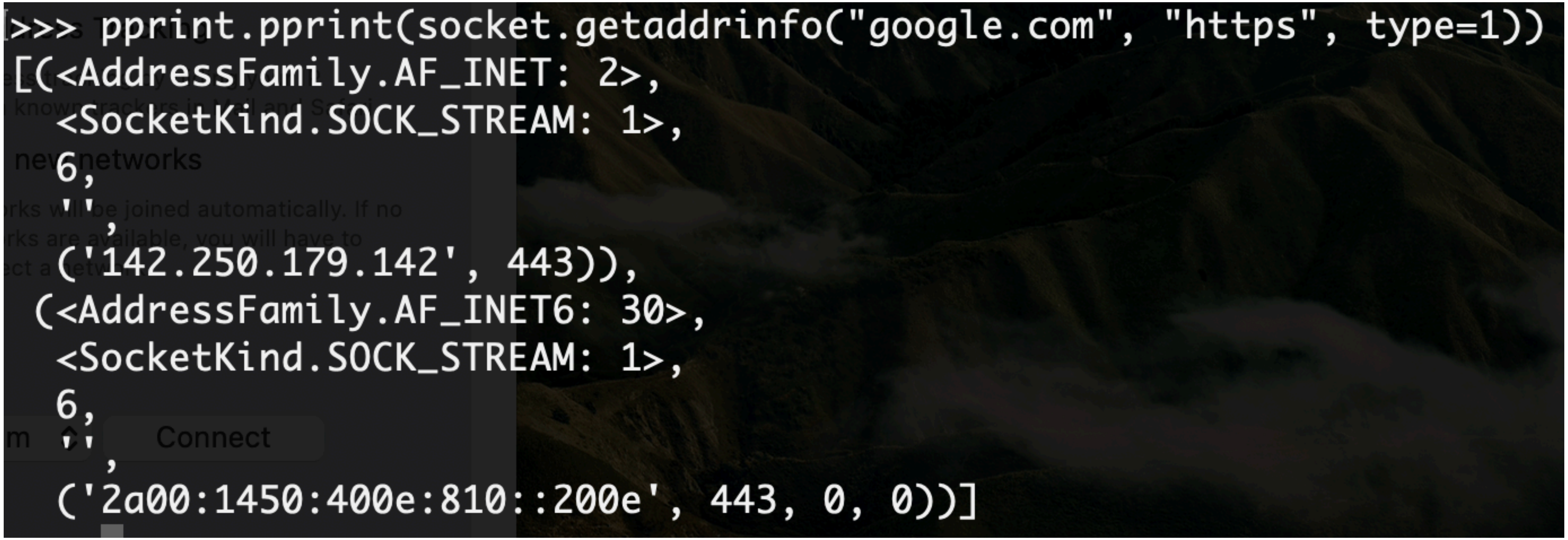
```
➜  ~ scutil --dns | head
DNS configuration

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
➜  ~
```

Surprises on macOS

When CLAT is active, the order of `getaddrinfo(3)` output is altered so IPv4 (via CLAT) is **preferred over native IPv6**



```
>>> pprint.pprint(socket.getaddrinfo("google.com", "https", type=1))
[(<AddressFamily.AF_INET: 2>,
  <SocketKind.SOCK_STREAM: 1>,
  6,
  '',
  ('142.250.179.142', 443)),
 (<AddressFamily.AF_INET6: 30>,
  <SocketKind.SOCK_STREAM: 1>,
  6,
  '',
  ('2a00:1450:400e:810::200e', 443, 0, 0))]
```

Summary

Pros

- Users have **only one network** to join
- IPv4 addresses are **not wasted** for devices that don't need them
  - 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 over native IPv4

Cons

- **Most complex** network setup
- IPv4 still **has to be deployed**
- NAT64 is **still needed**
- IPv4 communication between dual-stack and IPv6-only hosts is *problematic*
  - Setting up a Chromecast from an Android phone is impossible

Questions ?

Ondrej.Caletka@ripe.net
@ripencc

NOGs Participants Survey

- Takes 5-7 mins!
- Your views on:
  - Channels to stay connected
  - Important topics for NOGs to discuss
  - Challenges that prevent you from attending NOGs

https://ripe-ncc.typeform.com/to/SjgKEKSx