

IPv6 – Mostly: Mission **Im**Possible

Turning IPv4 Off in Google Enterprise
Network

Jen Linkova, furry@google.com



Image source: <https://freesvg.org/magical-unicorn>

Why?

- Running out of private IPv4 addresses
- Dogfood and testing
- Dual stack is hard

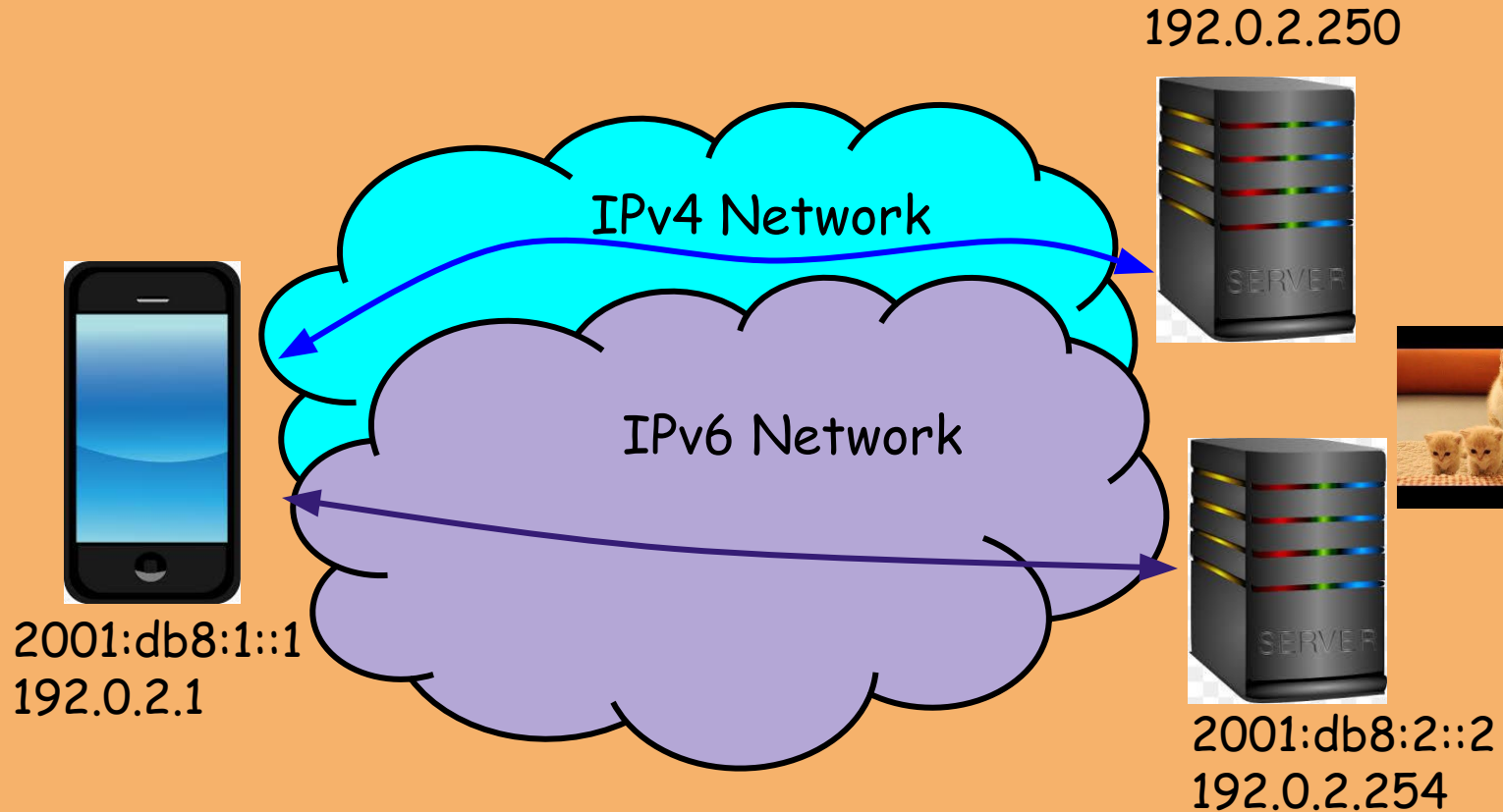


Source: www.wikipedia.org

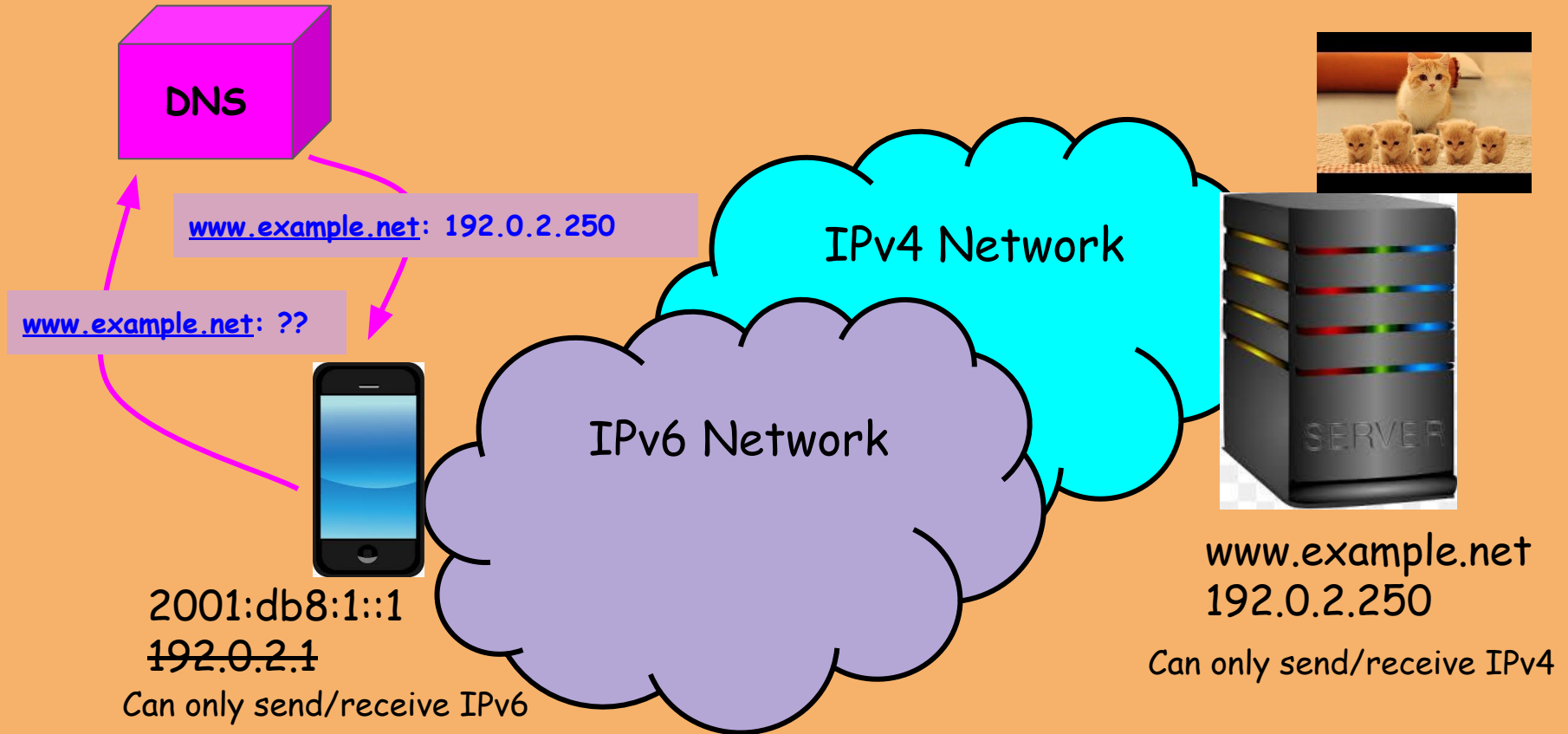
"Entities should not be multiplied without necessity."

— William of Ockham

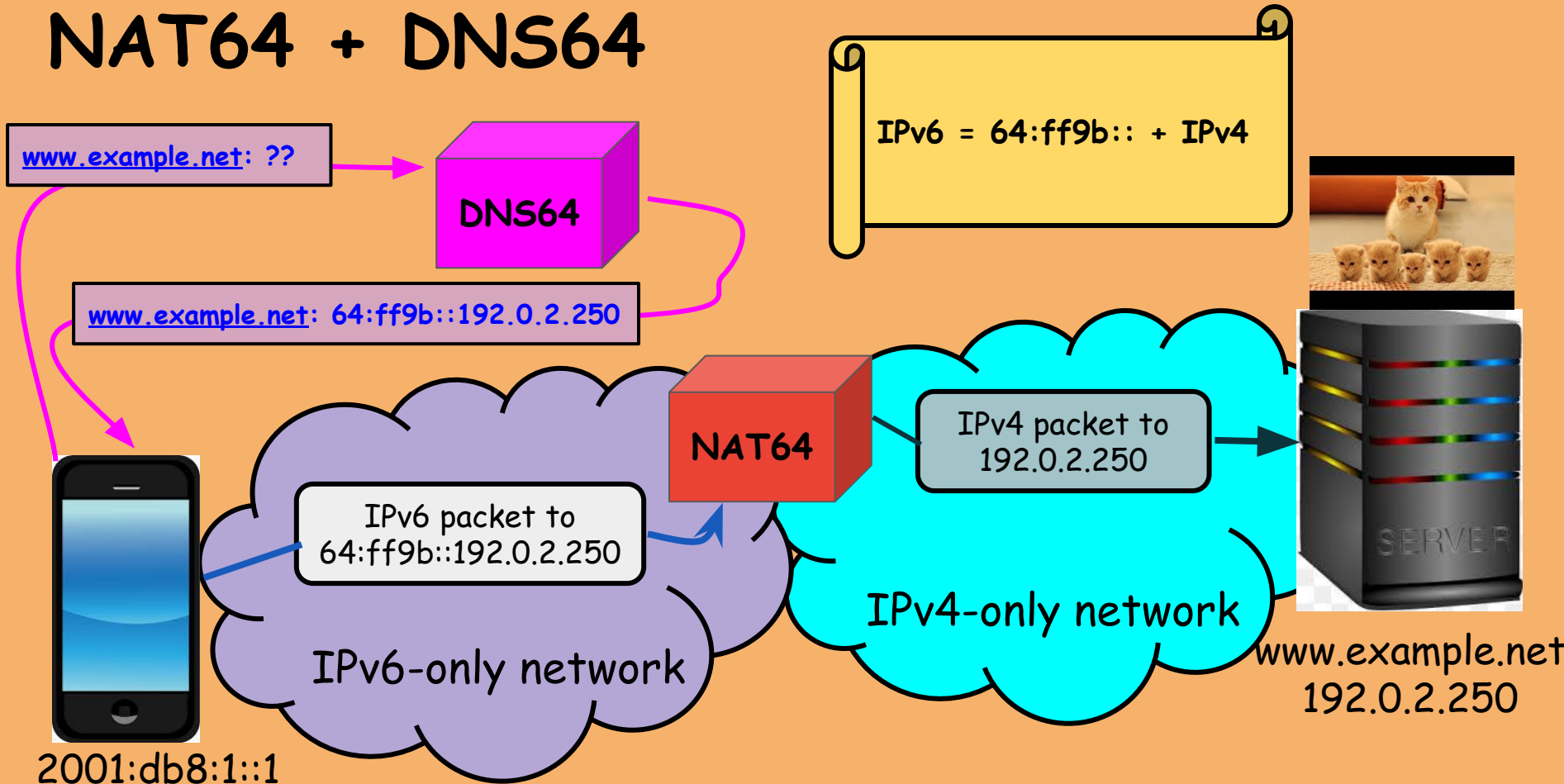
Dual-Stack Network



How Can We Remove IPv4?



NAT64 + DNS64



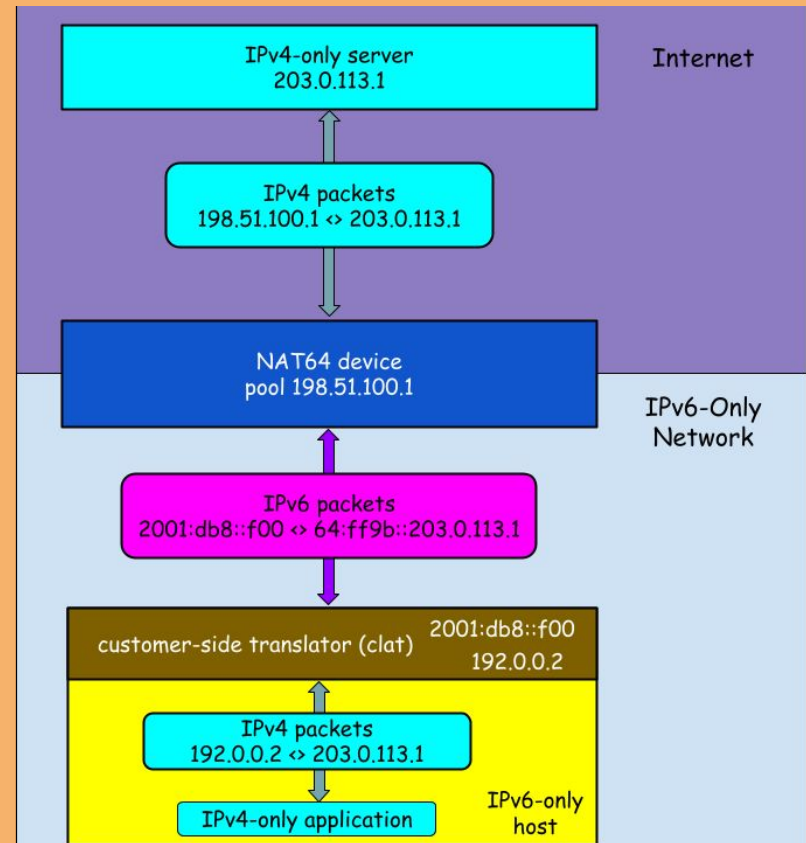
464XLAT (RFC6877)

DNS64 doesn't help if applications:

- Do not use DNS ("IPv4-literals")
- Only lookup IPv4 addresses
- Fail to operate without IPv4 address
- Use DNSSEC

Solution: CLAT

- Provide applications with an IPv4 address
- No IPv6 support in applications needed
- Needs NAT64 only, no need for DNS64
- DNSSEC-compatible



```
furry@Wintermute:~>ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
ping: sendto: No route to host
ping: sendto: No route to host
Request timeout for icmp_seq 0
ping: sendto: No route to host
Request timeout for icmp_seq 1
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 0 packets received, 100.0% packet loss
furry@Wintermute:~>
```

Without CLAT

```
furry@Wintermute:~>ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=119 time=3.965 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=119 time=4.784 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=4.112 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.965/4.287/4.784/0.357 ms
furry@Wintermute:~>
```

With CLAT

Signalling NAT64 Prefix via DNS (RFC7050)

```
furrry@Wintermute:~>dig +short @2001:4860:4860::8888 ipv4only.arpa a
192.0.0.171
192.0.0.170
furrry@Wintermute:~>dig +short @2001:4860:4860::8888 ipv4only.arpa aaaa
furrry@Wintermute:~>
furrry@Wintermute:~>dig +short @2001:4860:4860::64 ipv4only.arpa aaaa
64:ff9b::c000:ab
64:ff9b::c000:aa
furrry@Wintermute:~>
```

special ipv4-only name

NAT64 prefix (64:ff9b::96) + IPv4 address in hex

Signalling NAT64 Prefix in Router Advertisements

```
Wintermute:~ furry$ ipconfig getra en0
RA Received 11/01/2024 03:59:33 from fe80::42:1, length 96, hop limit 0, lifetime 3600s, reachable 0ms,
retransmit 0ms, flags 0x40=[ other ], pref=medium
    prefix info option (3), length 32 (4): 2001:67c:64:42::/64, Flags [ auto ], valid time 86400s,
pref. time 3600s
    pref64 option (38), length 16 (2): 64:ff9b::/96 lifetime 3600s
    rdns option (25), length 24 (3): lifetime 86400s, addr: 2001:67c:64:53::53:1
    source link-address option (1), length 8 (1): 00:50:56:bb:b5:0b
```

DNS Choice

Client can use any DNS

No DNS64

DNSSEC-compatible

Atomic Config

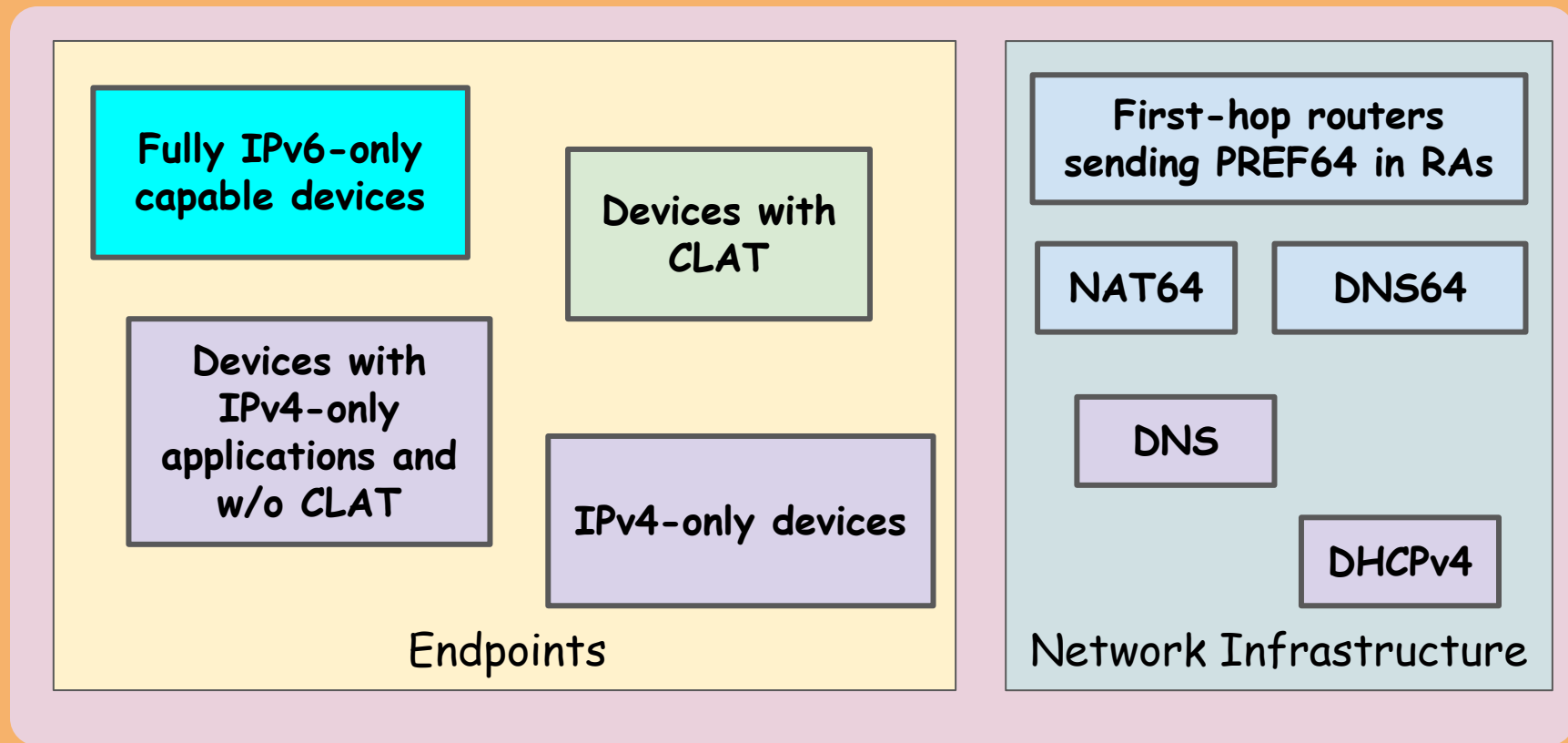
All IPv6 Config in
a single packet

Updatability

Ability to signal new
prefixes, if needed

See RFC 8781 and RFC 9872 for details.

Migration Landscape



How to Migrate to IPv6-Only?

Dedicated
IPv6-only
SSID/VLAN

Not a good idea

1. User Confusion

2. High IPv4 Consumption

3. Low Issue Visibility

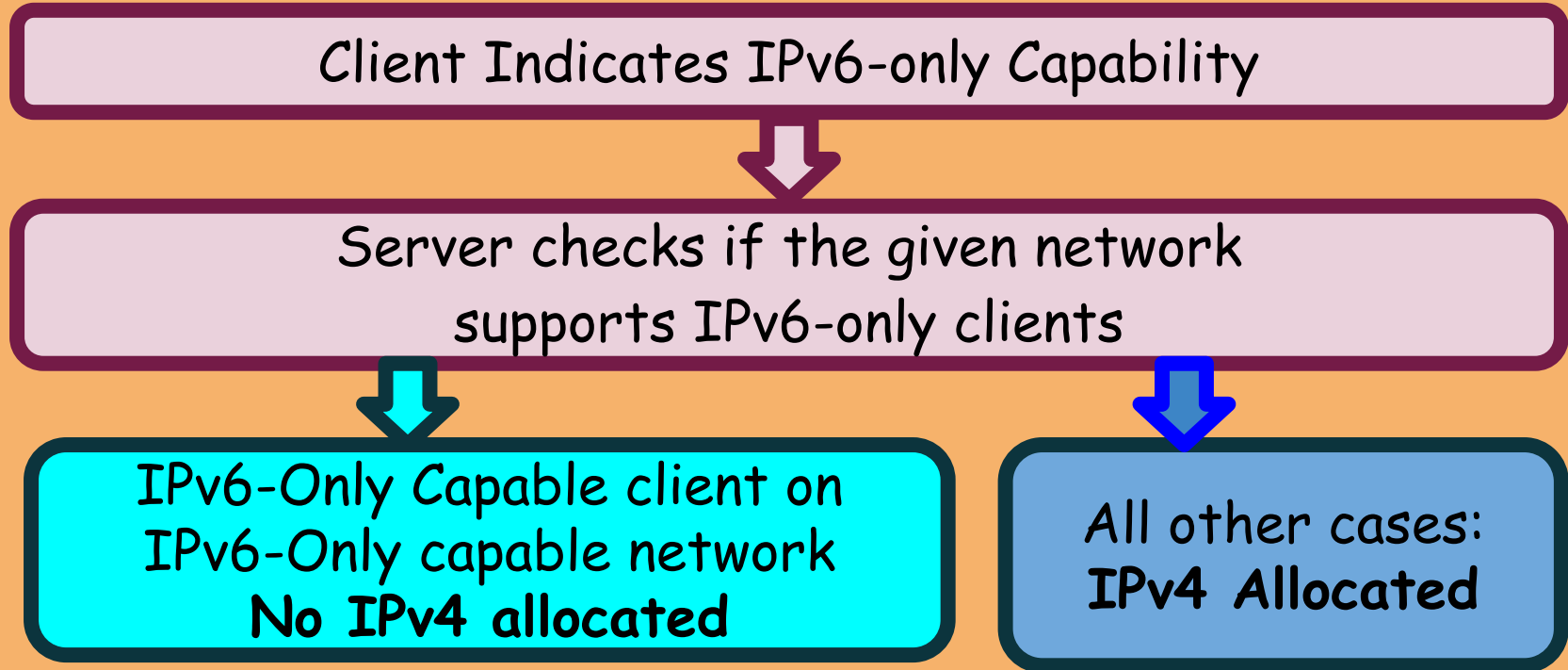
4. Scalability Issues

5. Operational Complexity

We needed something better!

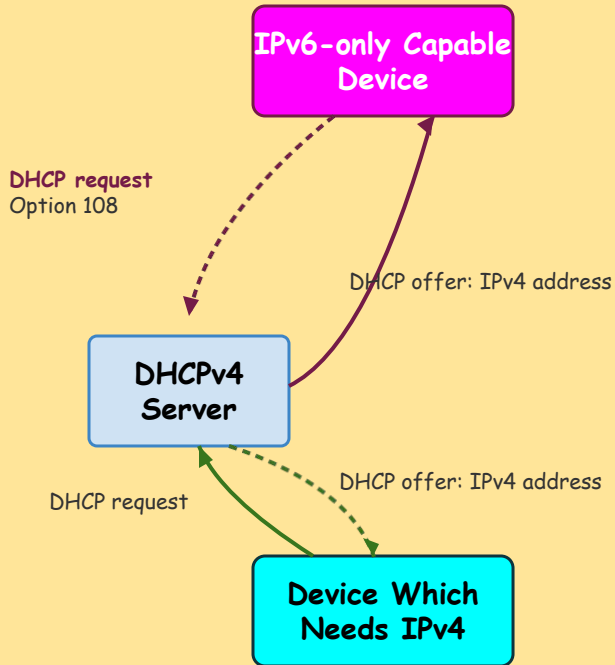
IPv6-Mostly Network

A network enabling co-existence of IPv6-only and IPv4-enabled devices

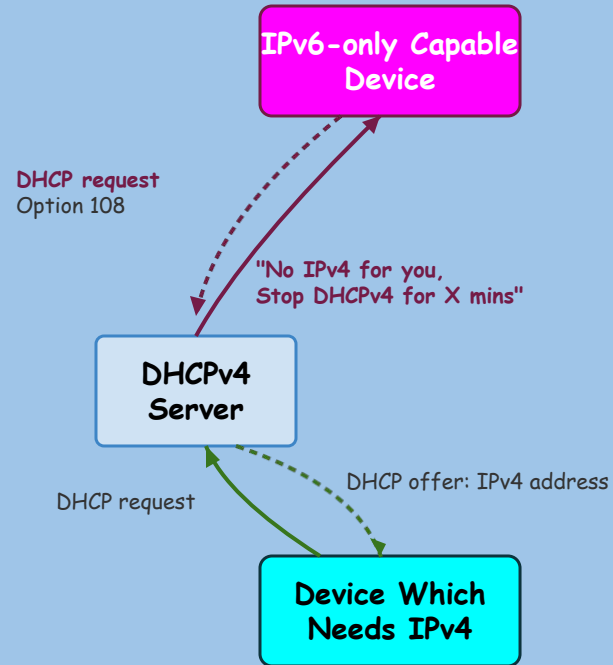


RFC8925: Use DHCPv4 to Turn IPv4 Off

IPv4-only / Dual-Stack Network



IPv6-Mostly Network



RFC8925 in a Nutshell

Client

Sends Option 108

Included in Parameter
Request List of
DHCPDISCOVER and
DHCPREQUEST

Server

- **Legacy Network**
Do not return Option 108
- **IPv6-Mostly Network**
Return Option 108 (stops
client DHCPv4)

No New Features Needed From DHCP Servers

How to Migrate to IPv6-Mostly

01 Deploy NAT64

02 Enable PREF64 in Router Advertisements

03 Configure DNS64 (Optional)

04 Enable DHCP Option 108
For the given pool/subnet

Endpoints sending Option 108 by default become IPv6-only

05 Enable Option 108 in managed Opt-In Clients, if any

Project Scope: Enabling Option 108

Network Infra

- SLAAC-only, no DHCPv6
- NAT64 & DNS64
 - NAT64 at site edge
 - RAs for DNS64/PREF64
- Central DHCPv4
- Wired Ports
 - 802.1x authentication
 - Dynamic VLAN

Endpoints

- Workstations, laptops, phones
- Guest devices
- Both wireless and wired

Option108 sent by default by:

- Apple (MacOS, iOS)
- Android
- ChromeOS (since 2024)

Manual Opt-in for Linux

Rollout Schedule: March–December 2023

00 Starting Point: NAT64, DNS64 and sending PREF64 in RAs rolled out

01 Pilot: 3 engineering offices (2 months)

02 Extended Pilot: 5 locations (1 month)

03 "Stop the bleeding": Enable IPv6-mostly mode for greenfields

04 Incremental Rollout: 5 months
10%, 15%, 25%, 50%, 70%, 90%, 100% of all existing subnets)
Apple and Android devices became IPv6-only; other OSes kept IPv4. ChromeOS: 2024

Results

Compatibility & Issues

- No blocking issues found
- Cosmetic bugs fixed in macOS 15
- Phones work flawlessly

User Experience

- Improved due to IPv6 becoming a first-class citizen

DHCPv4 Utilization

- Drops 3x to 4x on WiFi segments

Reclaimed Resources

- 500K+ IP addresses freed

Some Issues Encountered

Disabled IPv6

Devices with disabled or link-local only IPv6

Hosts Acting as Routers

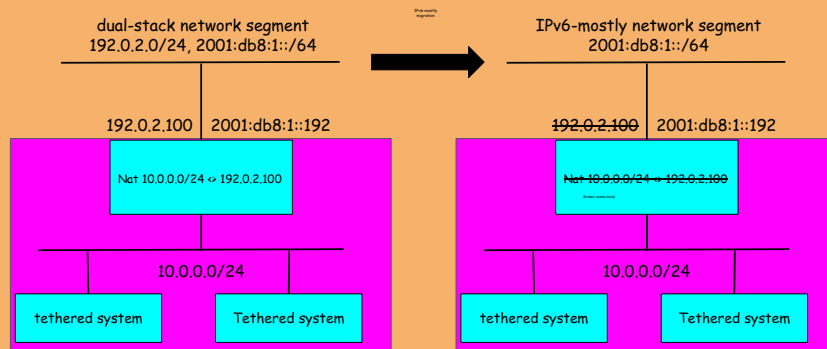
Extending connectivity downstream (tethering) breaks in IPv6-mostly.

Filtered Extension Headers

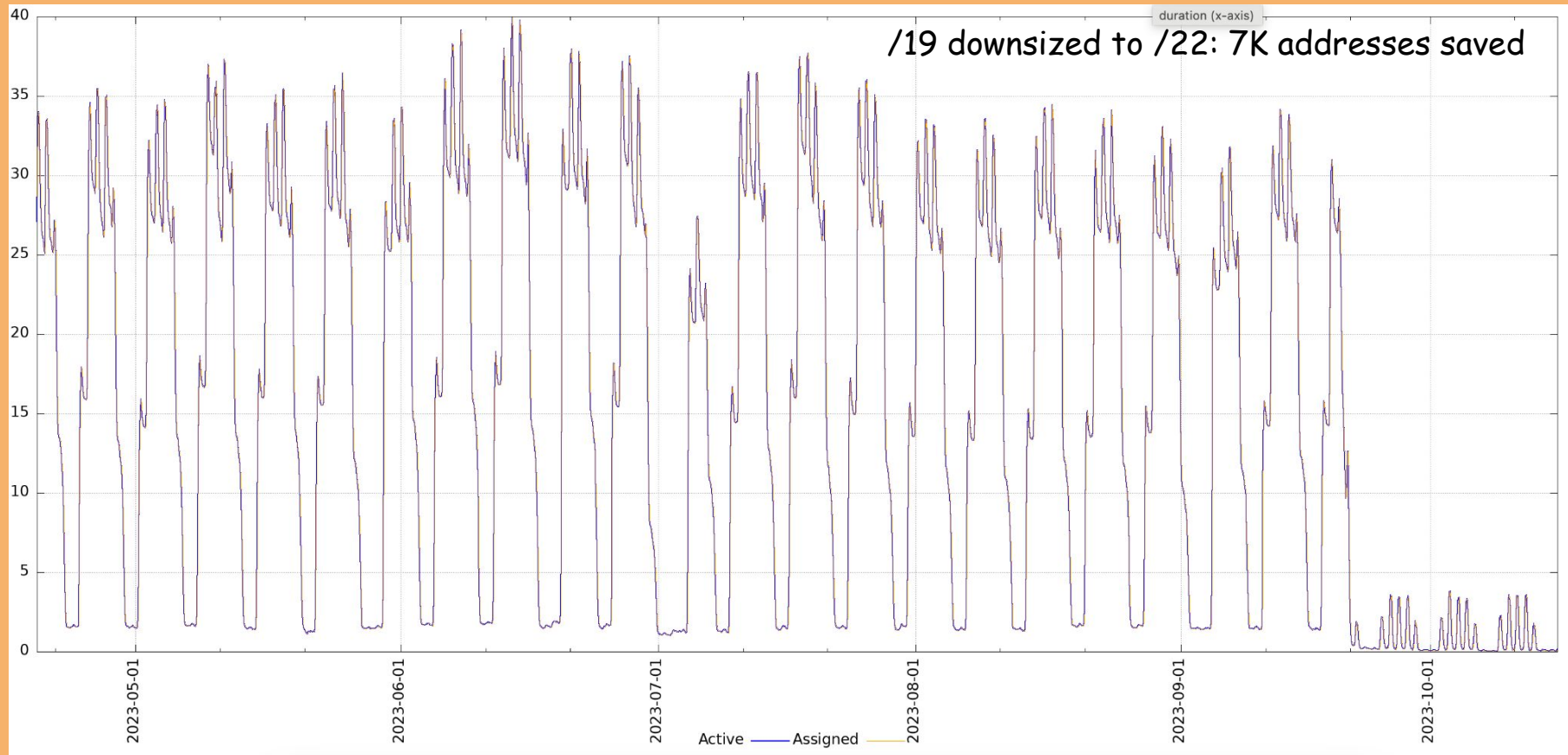
- ESP used in VPN and WiFi Calling
- Fragment headers in fragmented packets

Multiple Addresses per Device

- APs block additional IPv6 addresses
- Multiple EVPN Type-2 routes



WiFi Segment: DHCP Utilization Drop



IPv6-Mostly: Summary

Feasibility

- Smooth incremental migration
- Deployable NOW
- Already deployed in some enterprises and public WiFi

IPv4 Consumption

- 75-80% of hosts become IPv6-only
- Higher numbers expected as Windows implements CLAT/Option 108

Other Benefits

- Simplified operations
- Improved user experience

Systems Supporting IPv6-Mostly by Default

Apple

(iOS, MacOS)

Android

ChromeOS

NetworkManager 1.58 Released With 464XLAT CLAT Support & GENEVE Interface

Written by [Michael Larabel](#) in [Linux Networking](#) on 20 July 2026 at 04:12 PM EDT. [3 Comments](#)



NetworkManager 1.58 is out today with the latest feature enhancements for bettering wired and wireless network management on Linux.

NetworkManager 1.58 adds support for 464XLAT Customer-side Translator (CLAT) to create a virtual IPv4 interface for intercepting outbound IPv4 traffic from legacy applications and translate the headers into IPv6. 464XLAT is for helping IPv6 transition to allow devices on IPv6-only networks to still access legacy IPv4 services and applications.



r/ipv6 · 1y ago
UnderEu



NetworkManager v1.52 finally adds support for RFC 8925 - "IPv6-only preferred" DHCPv4 option



In November 2025, we announced the private preview of Windows CLAT, marking a major milestone in helping customers operate successfully on IPv6-only and IPv6-mostly networks.

Since then, we've worked closely with customers across government, enterprise, and the broader internet community to validate the feature, refine the experience, and ensure it's ready to scale.

If you're interested in the background on why Windows CLAT exists, how it works, and the standards it's built on, we encourage you to read the original announcement:

[Windows CLAT Enters Private Preview: A Milestone for IPv6 Adoption](#)

Public Preview Now Available

Today, we're thrilled to announce that **Windows CLAT is now available in public preview.**

The public preview is available starting with [Windows Insider Canary](#) build #29599-1000, and will continue to become available in other Windows Insider Channels over the coming weeks. This expands access beyond the private preview cohort and enables a broader set of customers to evaluate Windows CLAT in their environments and provide feedback as we continue toward general availability. For public preview, we also now support [Group Policy Object \(GPO\)](#) to configure Windows CLAT.

Conclusions

Not truly operating IPv6 while IPv4 is active!

Happy Eyeballs Hide Issues

- Users do not report problems
- Network bugs remain unfixed

"Workstation loses IPv6 DNS for minutes after waking up"

References

[draft-ietf-v6ops-6mops](#), Deployment and Operations Considerations

RFC 8781

Discovering PREF64 in
Router Advertisements

RFC 8925

IPv6-Only Preferred
Option for DHCPv4

RFC 9131

Gratuitous ND: Neighbor
Cache on First-Hop

RFC 9663

DHCPv6 Prefix
Delegation per Client

RFC 9762

Signal DHCPv6-PD
Availability via RAs

RFC 9872

Discover IPv6 Prefix for
Address Synthesis

QUESTIONS?

Ask away!

