



Mission ~~Im~~Possible

Turning IPv4 Off in an Enterprise Network

Jen Linkova, furry@google.com

Motivation

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

Network Overview

- SLAAC-only (no DHCPv6 for address assignment)
- NAT64/DNS64 to access IPv4-only destinations
 - NAT64 at the site edge
 - Router Advertisements options for DNS64 and PREF64
- Centralized DHCPv4 infrastructure
- Wired ports: 802.1x + dynamic vlan assignment

Previously on...

2020: IPv6-only Guest WiFi and wired networks

Dedicated IPv4-enabled SSID and wired vlan for fallback

Reclaimed a lot of IPv4 addresses

More details: ["The Day I Broke All the Treadmills" RIPE81 presentation](#)

IPv6-Only Guest: Lessons Learned

Dedicated SSID/VLAN: not a good idea

- Confusing for users
- Higher IPv4 consumption
- Lower visibility to issues
- Scalability concerns
- Operational complexity

We need something better!

IPv6-mostly Network

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

Client Indicates IPv6-only Capability

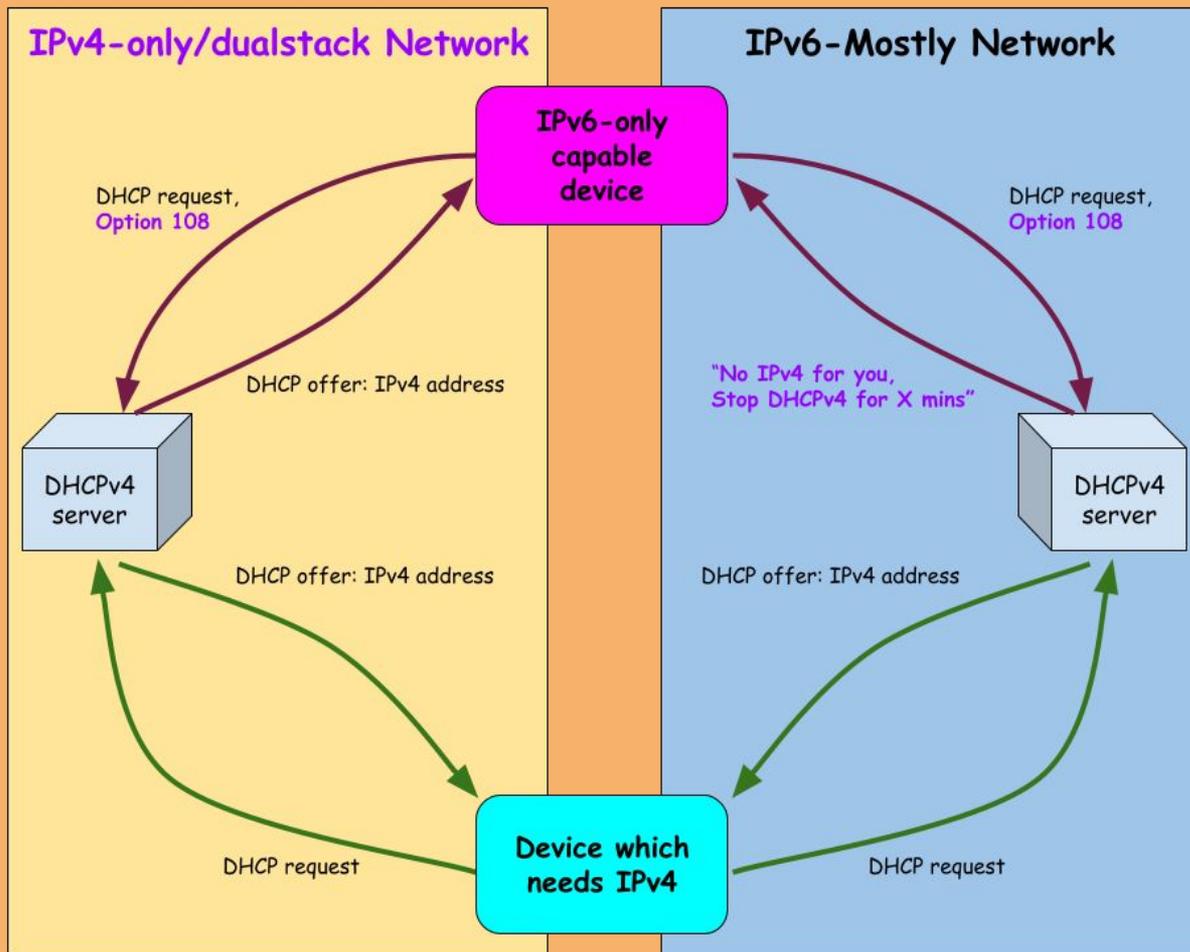
```
graph TD; A[Client Indicates IPv6-only Capability] --> B[Server checks if the given network supports IPv6-only clients]; B --> C[IPv6-Only Capable client on IPv6-Only capable network No IPv4 allocated]; B --> D[All other cases: IPv4 Allocated];
```

Server checks if the given network supports IPv6-only clients

IPv6-Only Capable client on
IPv6-Only capable network
No IPv4 allocated

All other cases:
IPv4 Allocated

RFC8925: Use DHCPv4 to Turn IPv4 Off



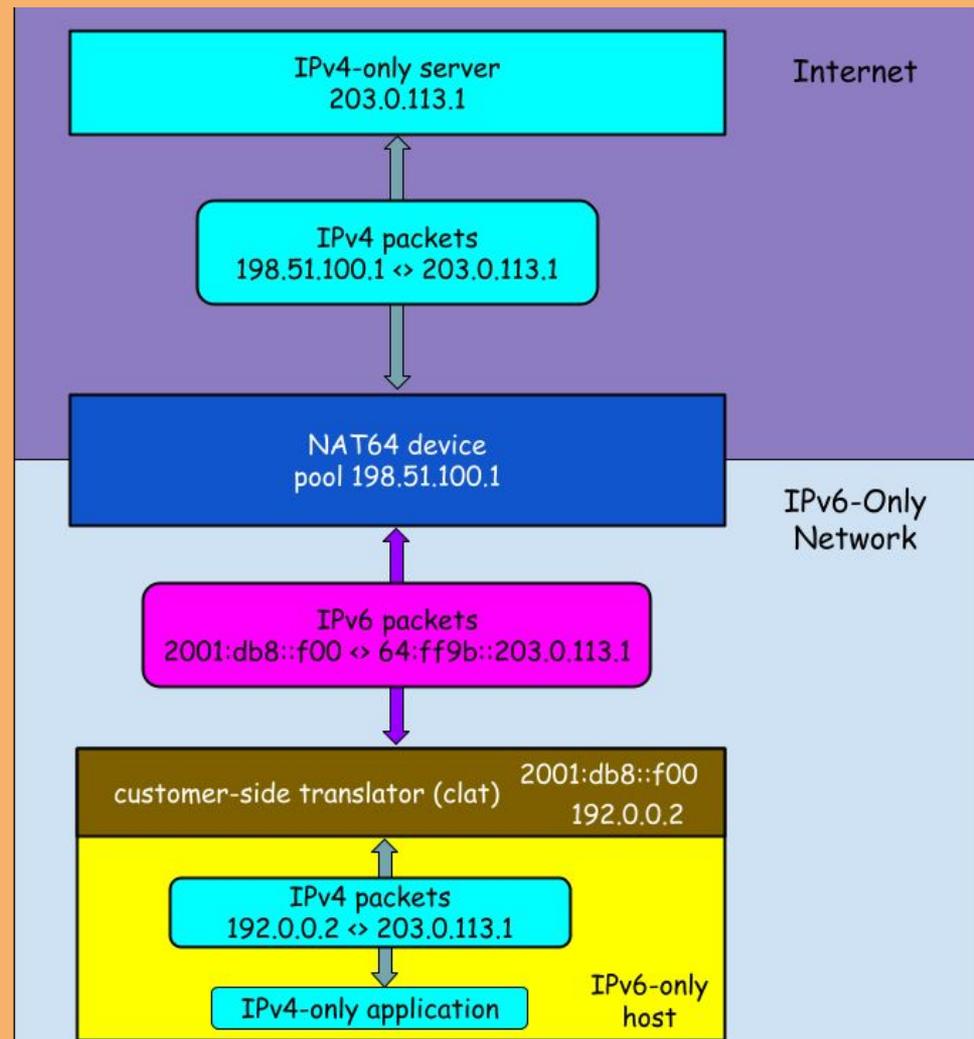
464XLAT (RFC6877)

DNS64 doesn't help if applications:

- Do not use DNS ("IPv4-literals")
- Only lookup IPv4 addresses
- Fail to operate w/o IPv4 address
- Uses DNSSEC

Solution: 464XLAT

- Provide applications with a private IPv4 address
- needs NAT64 only, no need for DNS64
 - DNSSEC-compatible



Project Scope

Network Infrastructure across all offices globally:

- Corporate WiFi and IPv4-enabled (fallback) Guest WiFi
- Wired user-facing segments

Devices migrated to IPv6-Only:

- All Android, iOS (15+), MacOS 13+
 - send DHCPv4 Option 108
 - support 464XLAT and PREF64
- Opt-in for selected ChromeOS and Linux devices

Rollout Schedule: March - Nov 2023

- Pilot in 3 locations for 2 months
- Extended pilot in 5 locations for 1 month
- “Stop the bleeding”: enable IPv6-mostly for greenfields
- Incremental rollout in 4 months, enabling Option 108 per subnet (10, 15, 25, 50, 60, 70, 80, 90, 100% of all networks)

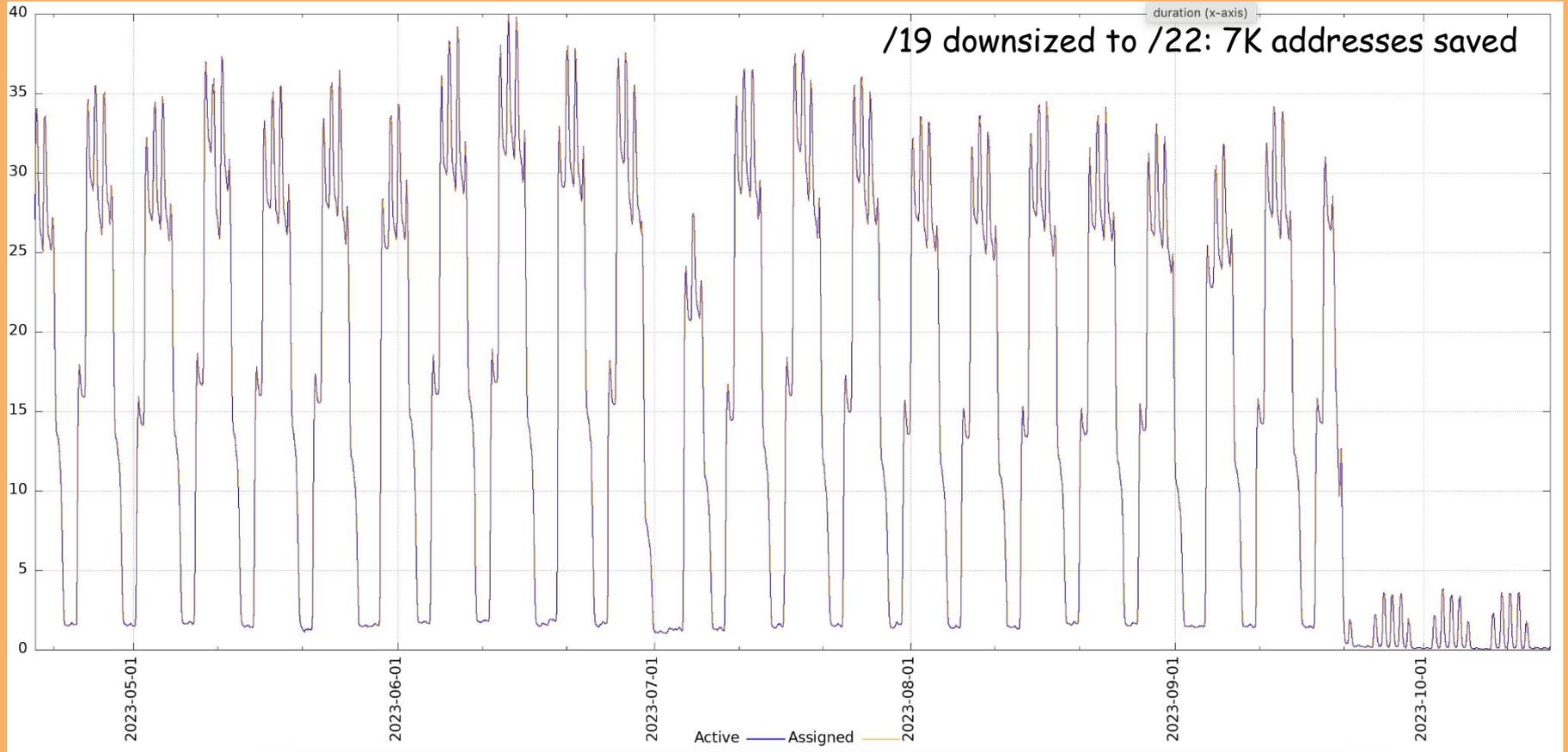
WE ARE HERE



Results

- No blocking issues found
 - *A few cosmetic issues: all fixed in MacOS Sonoma*
- DHCPv4 utilization drops by 3-4 times (average) on WiFi
- Expecting to reclaim at least 300K addresses

A Random Network: DHCP Utilization Drop



Lesson Learned #0

The only way to get IPv6 deployed:
to run out of (private) IPv4

Lesson Learned #1: "You Know Nothing, Jon Snow"

You do not really operate IPv6 until you turn IPv4 off

- Happy Eyeballs hide the problems
 - *"My workstation loses IPv6 DNS for a few mins after waking up"*
- Users do not report issues
- Issues are not getting fixed

Discovery #1: ~~Duck~~ Host Test

Dual-stack network segment
192.0.2.0/24, 2001:db8:1::/64

192.0.2.100

2001:db8:1::192

A device which
looks like a host
and
behaves like a host,
it's probably a host

..or is it a router?

dual-stack network segment
192.0.2.0/24, 2001:db8:1::/64

IPv6-mostly
migration

IPv6-mostly network segment
2001:db8:1::/64

192.0.2.100 2001:db8:1::192

Nat 10.0.0.0/24 ↔ 192.0.2.100

10.0.0.0/24

tethered system

Tethered system

192.0.2.100 2001:db8:1::192

~~Nat 10.0.0.0/24 ↔ 192.0.2.100~~

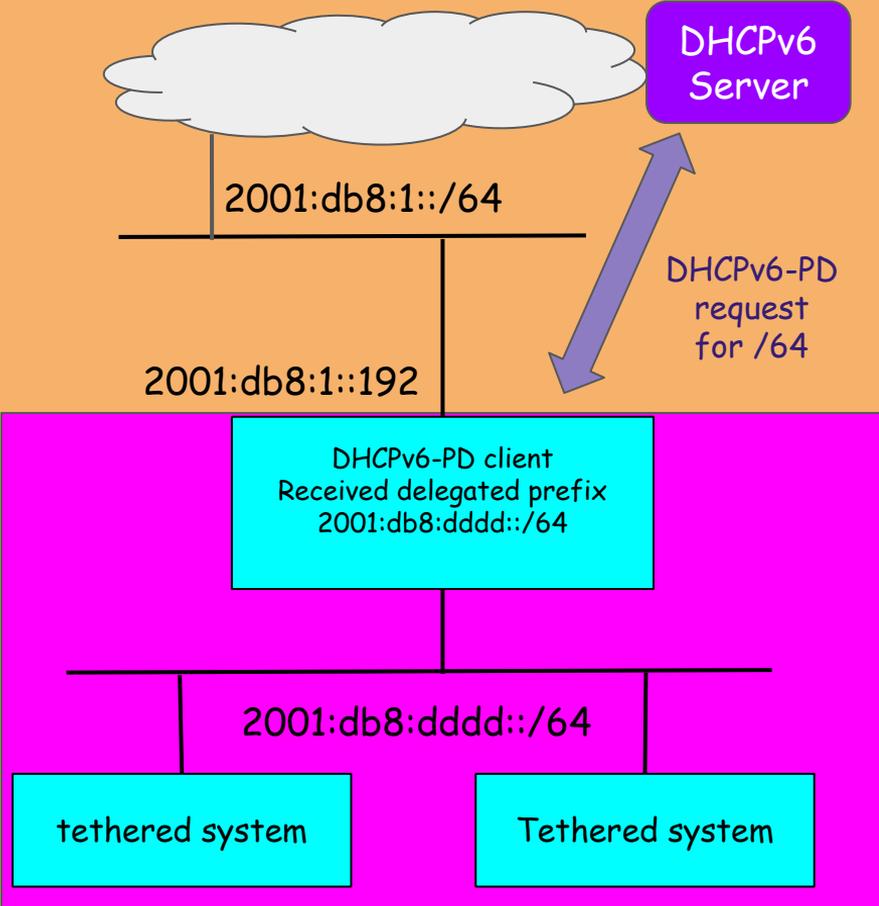
Broken connectivity

10.0.0.0/24

tethered system

Tethered system

Solution: DHCPv6-PD



Other Interesting Issues (see Appendix)

- IPv6 disabled (or set to link-local) on endpoints
- Extension Headers blocked: Fragmentation and ESP
- ESP/IPSec: various issues with firewalls/NAT64
- Devices with 10+ IPv6 addresses: blocked by WiFi
- Clients moving between VLANs (renumbering)
 - Rule 5.5 of Default Address Selection is crucial
- Devices losing IPv6 in 5 secs after RAs
 - WiFi APs getting ND proxy wrong..
- Packets from 192.0.0.2 on wire (fixed)
- Traceroute to ipv4 addresses: only '*' (work in progress)

RFCs Published

- [RFC 8781](#)
 - Discovering PREF64 in Router Advertisements
- [RFC 8925](#)
 - IPv6-Only Preferred Option for DHCPv4
- [RFC 9131](#)
 - Gratuitous Neighbor Discovery: Creating Neighbor Cache Entries on First-Hop Routers

New Drafts

- Using DHCPv6-PD to Allocate Unique IPv6 Prefix per Client in Large Broadcast Networks ([draft-ietf-v6ops-dhcp-pd-per-device](#))
- 464 Customer-side Translator (CLAT): Node Recommendations ([draft-link-v6ops-claton](#))
- Using Subnet-Specific Link-Local Addresses ([draft-link-v6ops-gulla](#))

Next Steps

2024: Migrate ChromeOS and Linux endpoints

The screenshot shows the ChromeOS Experiments page. At the top left, the word "Experiments" is displayed. A blue box in the top right corner contains the text "ChromeOS 114 and above". Below the title, there are two columns: "Available" and "Unavailable". The "Available" column contains the experiment "Enable RFC8925 (prefer IPv6-only on IPv6-only-capable network)". The description for this experiment reads: "Let ChromeOS DHCPv4 client voluntarily drop DHCPv4 lease and prefer to cooperate IPv6-only, if the network is also IPv6-only capable. - ChromeOS". Below the description is the code "#enable-rfc-8925". To the right of the code is a dropdown menu with the following options: "Default" (selected), "Enabled", and "Disabled".

Experiments

ChromeOS 114 and above

Available Unavailable

Enable RFC8925 (prefer IPv6-only on IPv6-only-capable network)

Let ChromeOS DHCPv4 client voluntarily drop DHCPv4 lease and prefer to cooperate IPv6-only, if the network is also IPv6-only capable. - ChromeOS

#enable-rfc-8925

Default ▾

Default

Enabled

Disabled

QUESTIONS?

Appendix
(Time-Permitting Slides)

Lesson Learned #2: Extension Headers

Make sure Extension Headers are permitted

Especially

- Fragment Header
- ESP Header
 - Used by IPSec
 - VPNs
 - WiFi Calling

On the Importance of Checksum

NAT64 and IPv4 UDP packets with zero checksum:

Corrupted IPv6 checksum

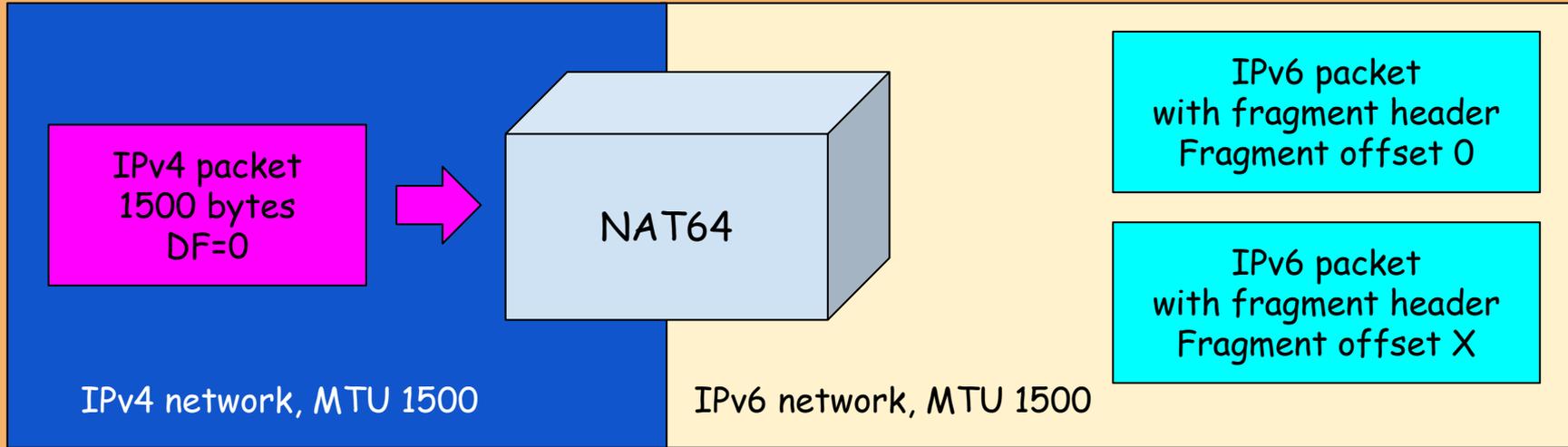
Unexpected journey: 'there and back again'

Firewalls permitting outgoing IPSec traffic

...do not create the state for the return traffic

...for there are no ports!

Discovery #3: Fragmentation Strikes Back



Caveats:

some NAT64 platforms use "1280" as a default size for translated packets instead of IPv6-only interface MTU.

Lesson Learned #3: Don't Disable IPv6

- "just disable IPv6 and see if it helps" wasn't a good idea.
- Had to automate enabling IPv6 on managed devices
- No way to fix it at scale for BYOD

Discovery #4: Hidden Limits

Host addresses: link-local, temporary, stable, 464XLAT



More addresses in case of virtual systems (ChromeOS: up to 20)



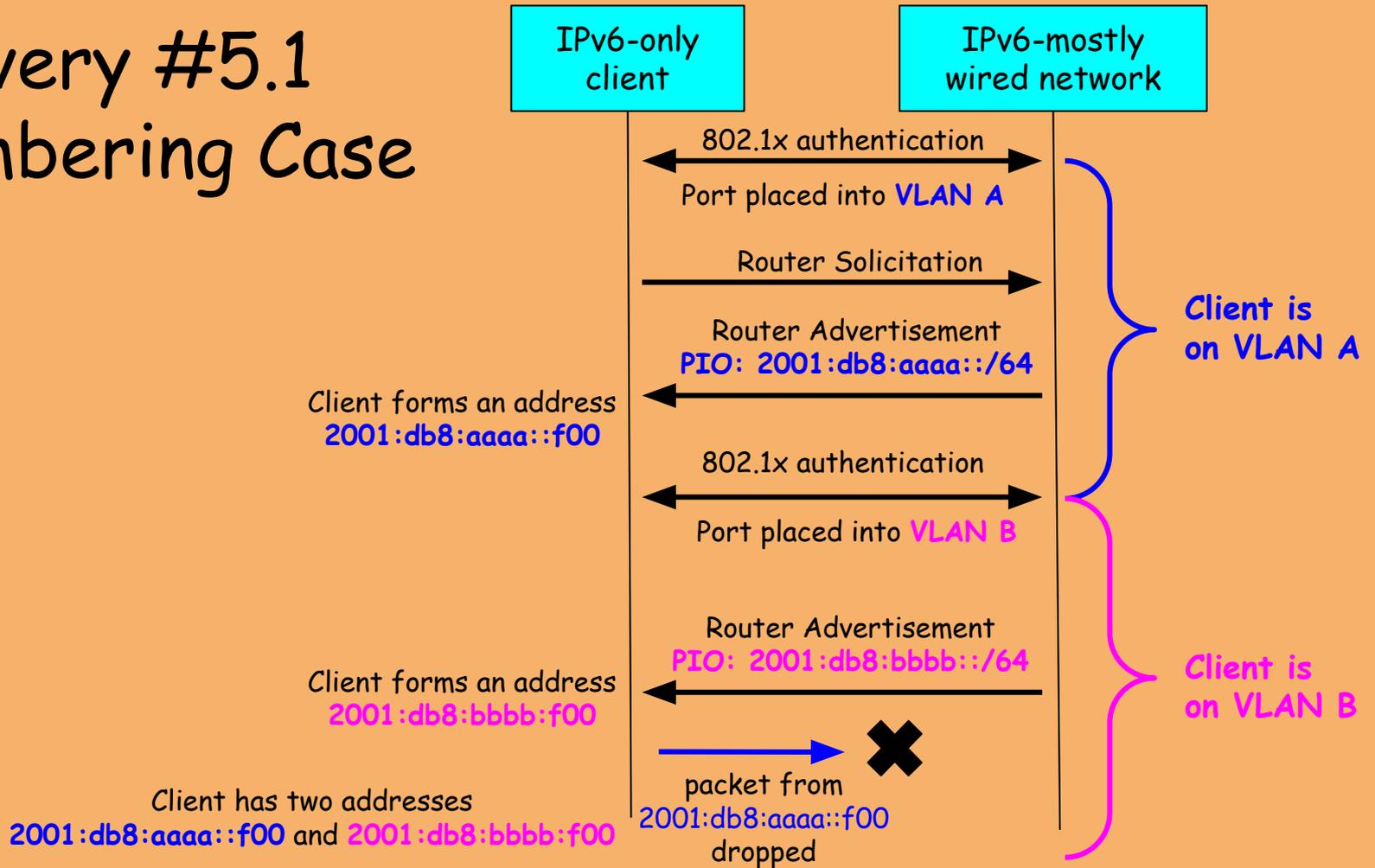
WiFi APs limit number of IPv6 addresses/client (limit can be as low as 7)



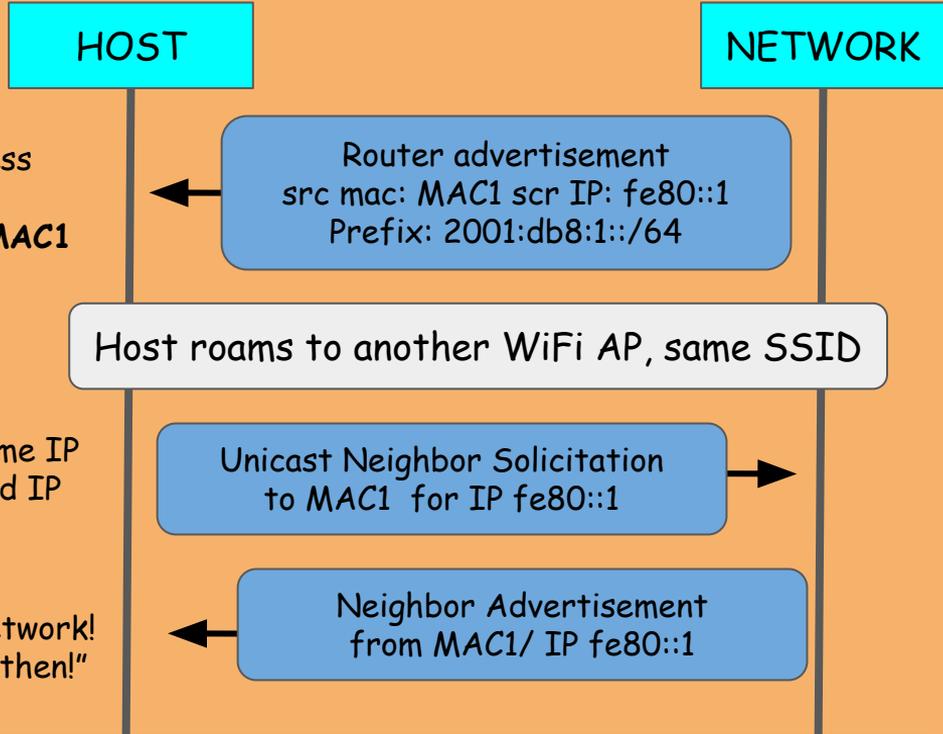
IPv6 addresses randomly lose connectivity

Discovery #5.1

Renumbering Case



Detecting Network Attachment (RFC6059)



Client forms an address
`2001:db8:1::f00`
Default gw: `fe80::1, MAC1`

Host roams to another WiFi AP, same SSID

?? Am I still connected to the same IP link? Have my gateway's MAC and IP changed??

"Oh, I'm on the same network!
Ill keep the addresses then!"

The host should send a Router Solicitation and check that /64 is the same but....

VRRPv3

datatracker.ietf.org/doc/html/rfc5798#section-7.3

attached to.

7.3. Virtual Router MAC Address

The virtual router MAC address associated with a virtual router is an IEEE 802 MAC Address in the following format:

IPv4 case: 00-00-5E-00-01-{VRID} (in hex, in Internet-standard bit-order)

The first three octets are derived from the IANA's Organizational Unique Identifier (OUI). The next two octets (00-01) indicate the address block assigned to the VRRP for IPv4 protocol. {VRID} is the VRRP Virtual Router Identifier. This mapping provides for up to 255 IPv4 VRRP routers on a network.

IPv6 case: 00-00-5E-00-02-{VRID} (in hex, in Internet-standard bit-order)

The first three octets are derived from the IANA's OUI. The next two octets (00-02) indicate the address block assigned to the VRRP for IPv6 protocol. {VRID} is the VRRP Virtual Router Identifier. This mapping provides for up to 255 IPv6 VRRP routers on a network.

7.4. IPv6 Interface Identifiers

IPv6 routers running VRRP MUST create their Interface Identifiers in the normal manner (e.g., "Transmission of IPv6 Packets over Ethernet Networks" [RFC2464]). They MUST NOT use the virtual router MAC address to create the Modified Extended Unique Identifier (EUI)-64 identifiers.

Router Interface 1 configuration

subnet 2001:db8:1:cafe::/64

vrrp-id 101

Router Advertisement

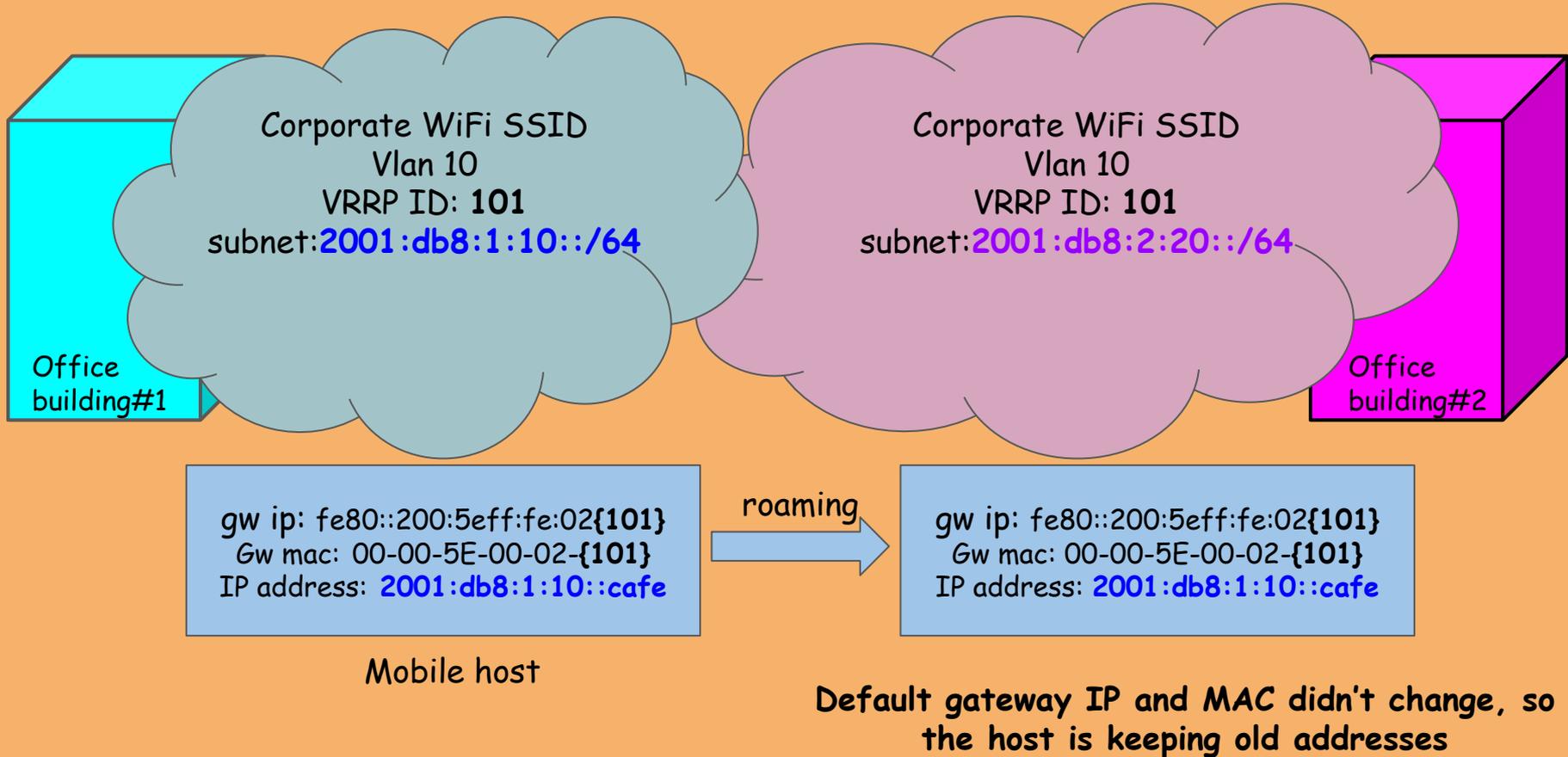
src mac: 00-00-5E-00-02-{101}

src ip: fe80::200:5eff:fe:02{101}

All segments with the same VRRP ID
have the same virtual router MAC

Some implementations violate "MUST"

Discovery #5.2 Roaming Case



Solution: RFC6724, Rule 5.5

IPv6-only client

IPv6-mostly network

Client forms an address
2001:db8:aaaa::f00
next-hop: fe80::a

Router Advertisement
from fe80::a
PIO: 2001:db8:aaaa::/64

Client's on
VLAN A



Client moves to vlan B

Client forms an address
2001:db8:bbbb:f00, next-hop fe80::b

Router Advertisement
from fe80::b
PIO: 2001:db8:bbbb::/64

Client's on
VLAN B



Client has two addresses
2001:db8:aaaa::f00, next-hop fe80::a
2001:db8:bbbb:f00, next-hop fe80::b

next-hop fe80::a masked as unreachable

Neighbor Discovery
for fe80::a

failure



next-hop: fe80::b

Packets from 2001:db8:bbbb::f00

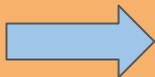


Rule 5.5: Prefer addresses in a prefix advertised by the next-hop.

"Globally Unique" Link-Local Addresses

Before: configure VRRP group ID only, link-local VIP encodes ID

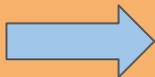
```
Router Interface 1 configuration
subnet 2001:db8:1:cafe::/64
vrrp-id 101
```



```
Router Advertisement
src mac: 00-00-5E-00-02-{101}
src ip: fe80::200:5eff:fe:02{101}
```

After: configure subnet 64 bit prefix as interface-id for link-local VIP

```
Router Interface 1 configuration
subnet 2001:db8:1:cafe::/64
vrrp-id 101
virtual-link-local: fe80::2001:db8:1:cafe
```



```
Router Advertisement
src mac: 00-00-5E-00-02-{101}
src ip: fe80::2001:db8:1:cafe
```

The Curious Case of Rip Van Winkle

- “My workstation loses IPv6 DNS for a few mins after waking up”
- Rootcause:
 - Router lifetime and RDNSS lifetime: 3600 secs
 - Device sleeps for > 1hr
 - A bug in the OS: DNS expires, the router is not!

Disappearing Routers

- Device loses IPv6 connectivity soon after connecting
- Obtain it back in a matter of minutes, loses it again
- Root cause:
 - WiFi AP with ND proxy: clears 'R' bit in NA