Deploying a Large Scale Enterprise Wi-Fi Network Using IPv6

Steve Tam, Wireless Engineer
Anusha Roy, Wireless Engineer

October 2025

Meta

Intro

- This presentation is intended to:
  - Provide some firsthand insight on lessons learned from migrating a enterprise wireless network to using IPv6
- Assumes some basic understanding of how IPv6 works
  - Does not cover how to potentially implement IPv6 for the rest of your network

Agenda

- Why IPv6?
- Meta's corporate Wi-Fi network
  - General overview & current scale
- Topology & timeline
- Migrating infrastructure
- Migrating clients
- Lessons Learned

Why IPv6?

Why IPv6? (for the Wi-Fi network)

- Meta has been a heavy adopter of IPv6, both externally & internally
  - All of our internal user networks are either dual-stacked or IPv6-only
  - Development servers are IPv6-only

- In 2018, we began running out of private IPv4 space
  - Only the 172.16.0.0/12 and 192.168.0.0/16 ranges are used for the corporate network

Why IPv6? (cont'd)

- In order to keep opening up new offices, we needed to be aggressive with migrating to IPv6

- From an operator perspective - simplified deployments
  - /64 for each AP subnet = 2^64 addresses available
  - No calculations needed upfront to right-size the AP subnet
  - No need to resize the AP subnet when expanding a site

Why IPv6? (cont'd)

- Same goes for clients - less of a need to constantly add IPv4 space as client counts go up
  - Especially in a world where devices are increasingly connecting over Wi-Fi
  - Flash crowds are a regular occurrence

Overview of Meta's Wi-Fi network

NANOG™

Overview / Scale

- Covers all Meta locations globally:
  - Offices in 90+ cities
  - 28 data centers
  - 76k employees

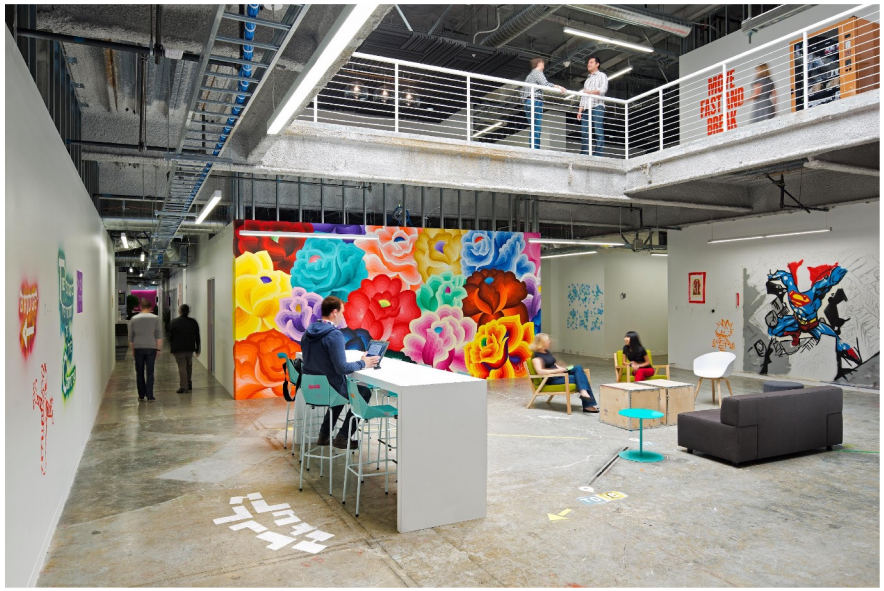
- ~100k wireless clients daily
  - macOS, Windows, Linux, Chro
  - iOS, Android
  - Wearables (Oculus, Orion) and robots

Overview / Scale

- Multi-vendor network:
  - Using the top two wireless LAN vendors by market share
  - Primarily on-prem & controller-based
- ~40k+ wireless access points
- ~400+ wireless controllers
- Using both vendor-provided network management systems (NMS) as well as homegrown ones

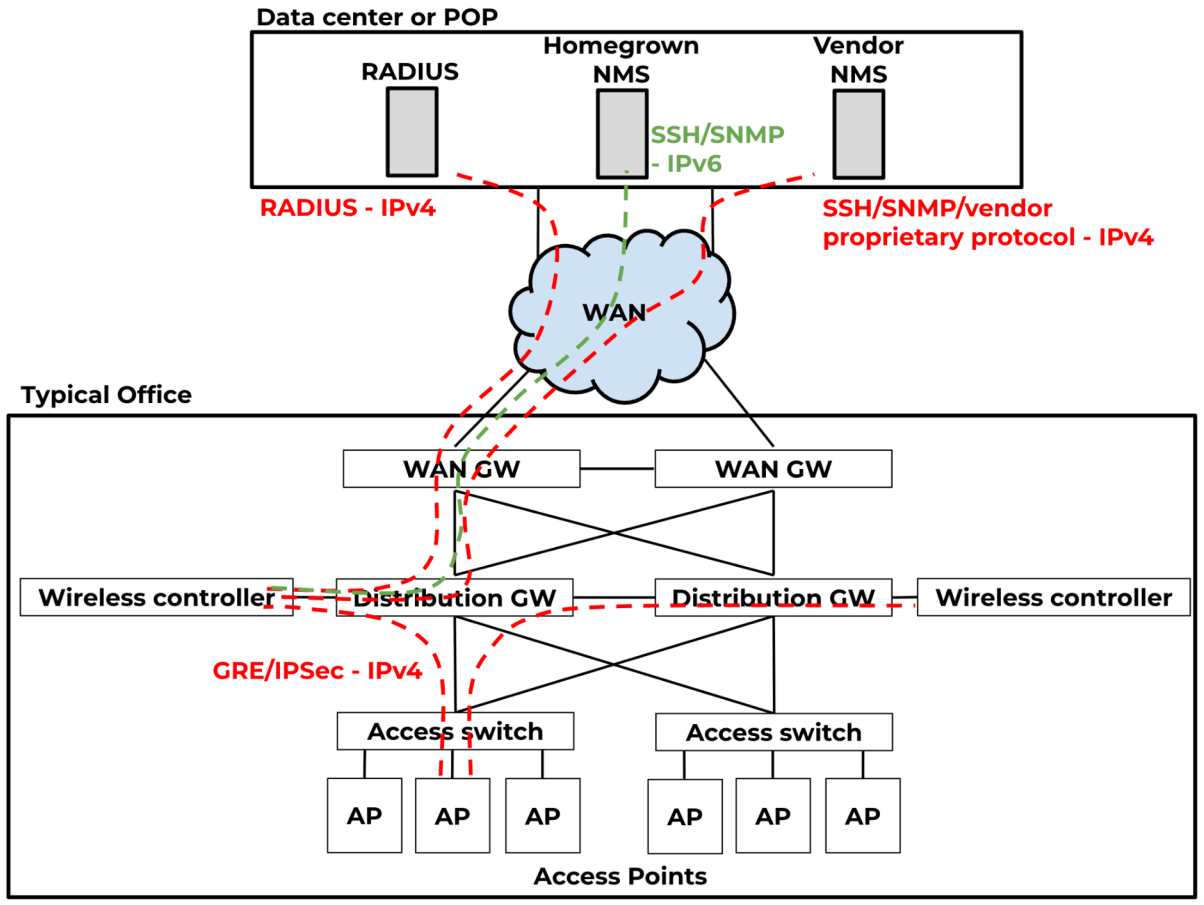
The Meta Corporate WLAN

- 3 main SSIDs:
  - Employee (802.1x auth)
  - Guest (PSK)
  - Lab/Test (PSK + MAC auth)



Topology / Timeline

Before - pre-2018

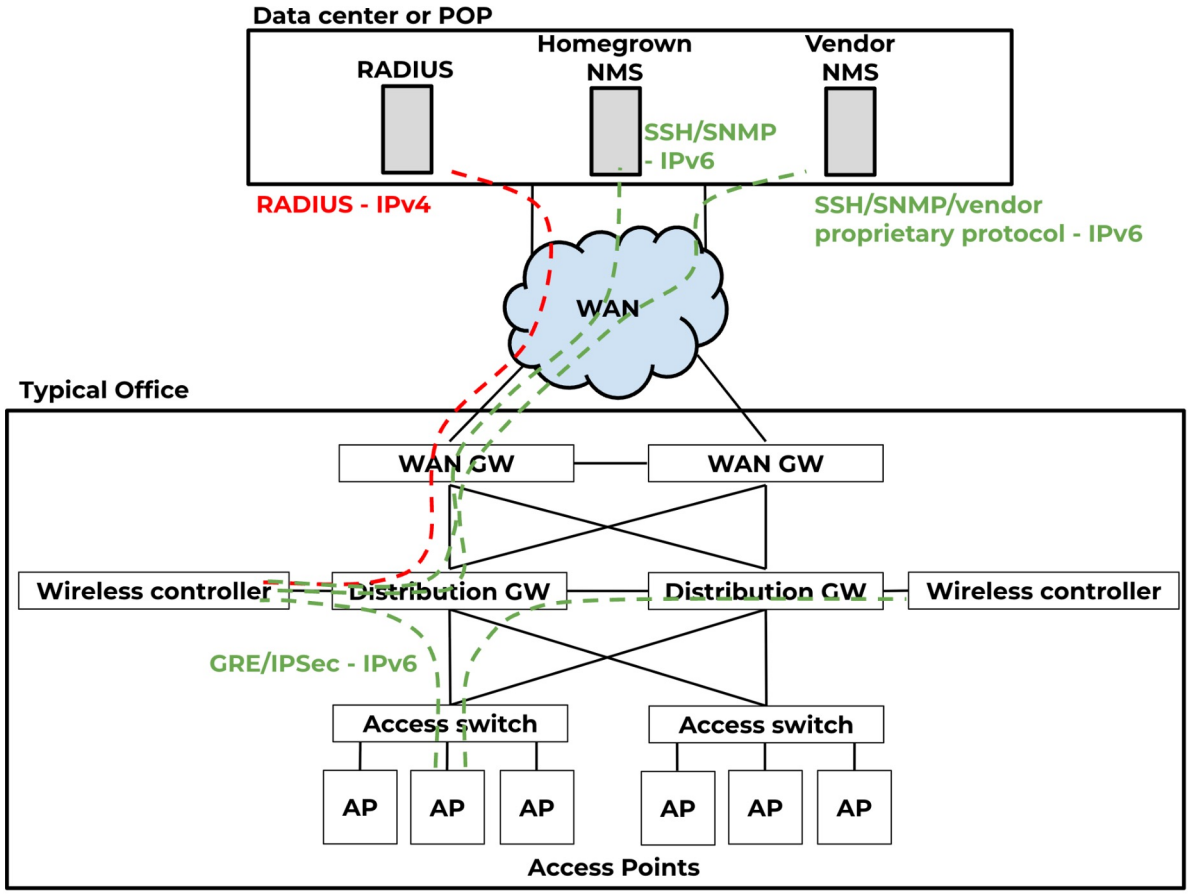


Timeline (Vendor 1)

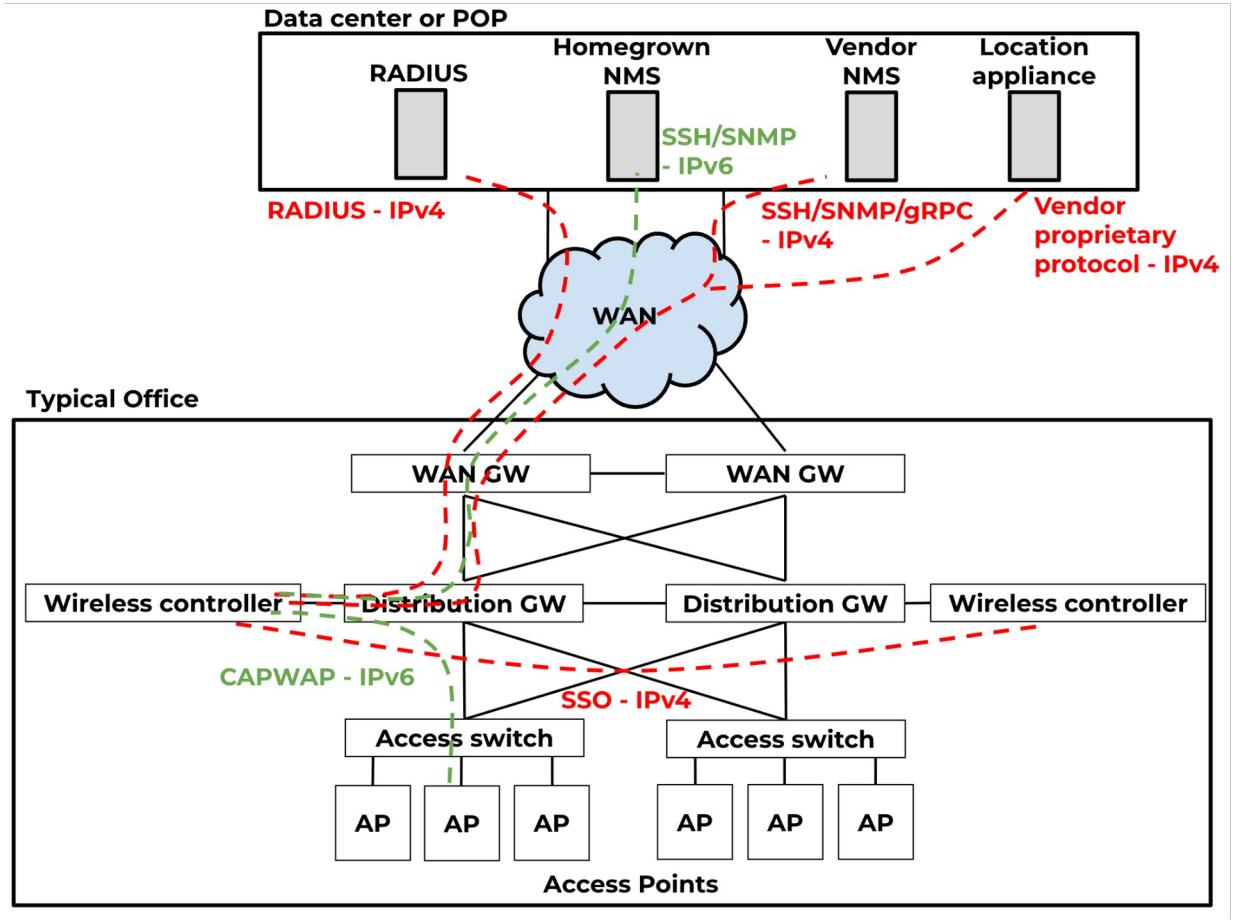
- **June 2018** - First IPv6-only pilot office
- **July 2018** - Guest network goes IPv6-only for new sites
- **Aug 2018** - Declaration that all new office builds go IPv6-only for the AP VLAN
- **Dec 2018** - First HQ site migrated to using IPv6 for APs

- **Feb 2019** - Vendor 1's NMS begins using IPv6 to controllers
- **Dec 2019** - Completed migration of all Vendor 1's APs to IPv6

Vendor 1 (end of 2019)



Vendor 2 (early 2020)

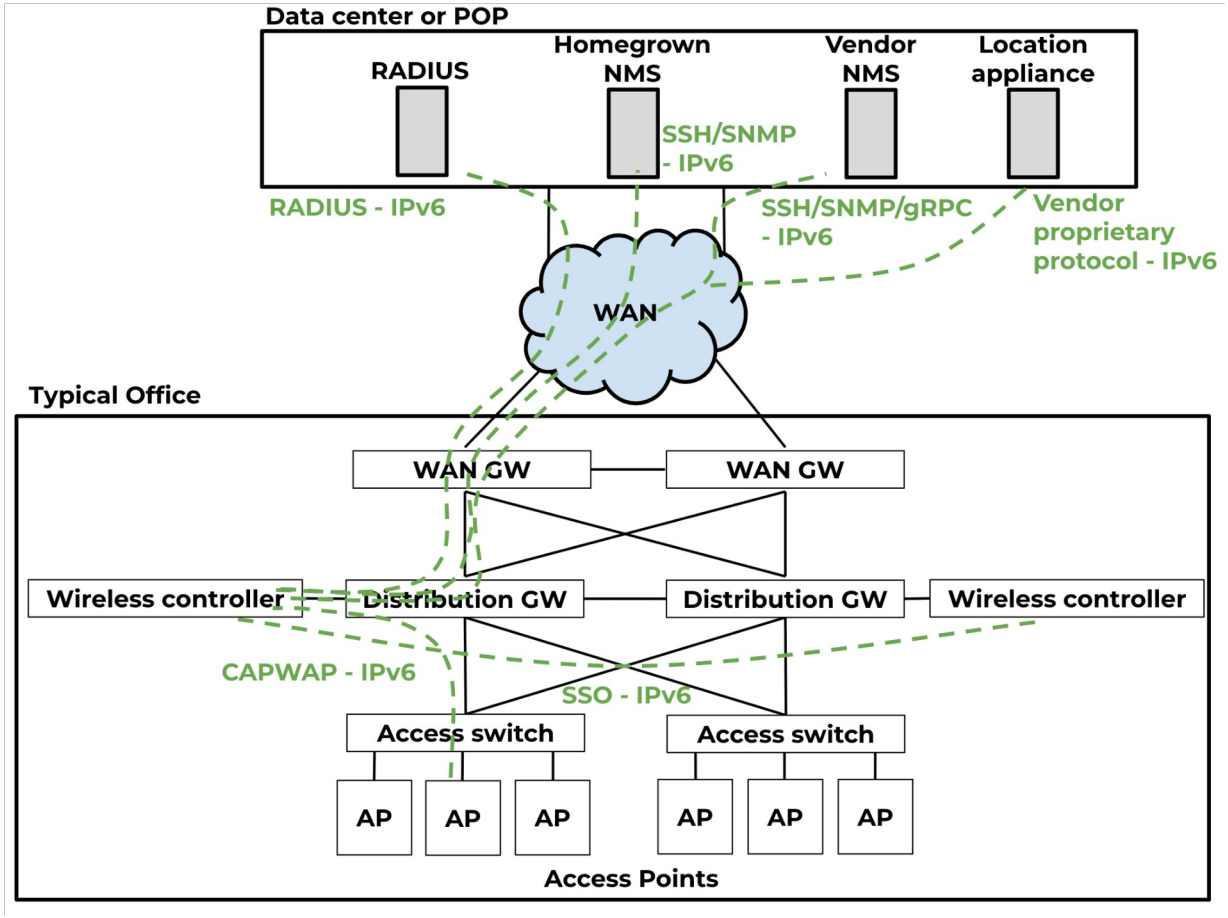


NANOG™

Timeline (Vendor 2)

- **Jan 2020** - APs using IPv6 to controllers
- **Feb 2021** - Controller SSO & NMS using IPv6
- **Mar 2021** - Location appliance using IPv6
- **Feb 2022** - RADIUS authentication over IPv6

Vendor 2 (by early 2022)



About RADIUS…

- It's from one of the two vendors that we use for WiFi
- Took a number of years for it to fully gain IPv6 support

- When using IPv4, we added controllers/switches just once:
  - By using a broad /12 or /16 range in a device group
  - Helps avoid adding them one-by-one each time a new one is deployed

About RADIUS…

- With IPv6 “support” initially added:
  - We had to add controllers/switches individually by /128
  - That’s 400+ controllers and 13k+ switches
  - Ended up building a script to add them by API
  - Support for IPv6 address ranges in a device group was added later

Migrating Infrastructure

Controller Discovery

- Typical methods:
  - Static
  - DNS
  - DHCPv6 option 52
    - Example:

```
subnet6 fd00:0:0:100::/64 {
   range6 fd00:0:0:100::1000 fd00:0:0:100::2000;
   option dhcp6.capwap-ac-v6 fd00::100:192:168:1:50;
}
```

    - Depending on your DHCPv6 server, raw options may be required

Controller Discovery (Vendor 1)

- APs successfully discovering a controller using DHCPv6 option 52:

```
Enable IPv6 for the Conductor v6 discovery
Enabling DHCPv6 ...
```

```
Running ADP...Done. Conductor is fd00::100:192:168:1:50
conductor is changed from 0 to fd00::100:192:168:1:50, cleanup cached info for old conductor
AP rebooted Fri Aug  1 00:40:36 UTC 2025; System cmd at uptime 0D 0H 4M 26S: uap conversion successful
```

Controller Discovery (Vendor 1)

- AP joined to the controller over IPv6:

```
(WLC) [mynode] #show ap database long

AP Database
-----------
Name              Group    AP Type  IP Address          Status     Flags  Switch IP
----              -----    -------  ----------          ------     -----  ---------
00:4e:35:c6:93:4e  default  535      fd00:0:0:100::1136  Up 4m:54s  rIL    fd00::100:192:168:1:50
```

Controller Discovery (Vendor 1)

- AP joined to the controller over IPv6:

```
(WLC) [mynode] #show ap consolidated-provision info ap-name 00:4e:35:c6:93:4e

ap name: 00:4e:35:c6:93:4e
ipv4 address type: dynamic
ipv4 address: 192.168.1.168
ipv4 netmask: 255.255.255.0
ipv4 gateway: 192.168.1.1
ipv4 lease: 6912
ipv4 dhcp server: 192.168.1.1
ipv4 dns server: 8.8.8.8, 8.8.4.4
ipv6 address type: dynamic(DHCPv6)
ipv6 address: fd00:0:0:100::1136
ipv6 lease: 2591831
ipv6 dns server: 3ffe:501:ffff:100:200:ff:fe00:3f3e
ipv6 gateway: fe80::92ec:77ff:fe8f:59c1
ipv6 dhcp option52: fd00::100:192:168:1:50
conductor preference: IPv4
Protocol in Use: IPv6(NO_IPV4_MASTER)
conductor: fd00::100:192:168:1:50
conductor discover type: DHCPv6
previous lms: none
lms addrs [0]: fd00::100:192:168:1:50
```

Controller Discovery (Vendor 2)

- APs successfully discovering a controller using DHCPv6 option 52:

```
CAPWAP State: Discovery
Got WLC address fd00::100:192:168:1:50 from DHCPv6.
Discovery Request sent to fd00::100:192:168:1:50, discovery type DHCP(2)
Discovery Request sent to 255.255.255.255, discovery type UNKNOWN(0)
Discovery Request sent to ff02::18c, discovery type UNKNOWN(0)
Discovery Response from fd00::100:192:168:1:50
No IPv4 AP Mgr in IPv4 pref mode. Try IPv6 mode...

CAPWAP State: DTLS Setup

CAPWAP State: Join
Sending Join request to fd00::100:192:168:1:50 through port 5248
Join Response from fd00::100:192:168:1:50
AC accepted join request with result code: 0
Received wlcType 0, timer 30
TLV ID 2216 not found
TLV-DEC-ERR-1: No proc for 2216
RTNETLINK answers: No such file or directory

CAPWAP State: Image Data
AP image version 8.10.104.96 backup 0.0.0.0, Controller 17.8.0.144
Version does not match.
```

Controller Discovery (Vendor 2)

- AP joined to the controller over IPv6:

```
WLC#sh wireless stats ap join summary
Number of APs: 1

Base MAC        Ethernet MAC    AP Name                                  IP Address                                      Status
-----------------------------------------------------------------------------------------------------------------------------
04eb.409f.9f80  04eb.409e.2724  AP04EB.409E.2724                         fd00::200:4ced:6526:d555:328e                   Joined

WLC#
```

Migrating Clients

Key Considerations for Migrating Clients to IPv6

- Host Initialization
  - Devices can't instantly identify if a network is IPv4-only, dual-stack, or IPv6-only
  - To ensure seamless connectivity, they typically configure both IPv4 and IPv6 right away
- Disabling IPv4
  - Some devices don’t support IPv6
  - IPv6-only networks require DNS64 and NAT64 to reach IPv4 websites

IPv6 Allocation: SLAAC & DHCPv6

- IPv6 address allocation for clients:
  - DHCPv6 for stateful assignment
    - Android doesn't support DHCPv6
    - DHCPv6 is basically like DHCPv4
  - SLAAC for stateless auto-address configuration
    - RDNSS for advertising DNS servers (not all platforms support this)
- Can mix using SLAAC & DHCPv6 with the managed-config and other-stateful-config flags in router advertisements

Large Subnet/VLANs on IPv6

- Broadcast/multicast traffic can cause network congestion and poor performance
  - IPv6 is more “chatty”- relies on multicast for essential functions like Neighbor Discovery
- On wireless, multicast is treated like broadcast—sent to all clients, often at the lowest data rate
- As the number of clients grow, so can the amount of broadcast/multicast traffic

Neighbor Discovery Caching

- Wireless controllers implement ND caching
- Controller keeps a cache of known IPv6 clients/neighbors
- When a device sends a Neighbor Solicitation (NS):
  - Controller checks its cache
  - If the target is known, the controller responds directly or forwards only to the relevant client
  - Prevents NS flooding to all clients on the VLAN & conserves airtime

Meta's Wi-Fi User Networks

- Employee = dual stacked w/ strides to go IPv6-only
  - Mobile devices (phones): IPv6-only on Wi-Fi
  - Laptops: new offices are IPv6-only; older sites still dual-stacked
- Have better control over the device mix which tends to be all IPv6-capable

Meta's Wi-Fi User Networks

- Guest = dual-stacked
  - Tried IPv6-only for awhile but had to go back and add IPv4
  - No control over device mix
    - External vendors bring in laptops with IPv6 disabled & network settings locked down by IT policy
- Lab = dual-stacked
  - Some IPv4-only IoT & wearable devices to deal with

Migrating Clients

- For IPv6-only networks, we rely on DNS64 & NAT64

- Some fun IPv6-only related issues previously encountered:
  - **Fedora** - network manager drops every ~45 seconds without an IPv4 address
  - **iOS 14.7** - iPhone drops on Wi-Fi every ~60 seconds
  - **Android 15** - drops due to low RA lifetime values (<3 minutes)
  - **Microsoft Outlook app on Android** - unable to sync/send emails on an IPv6-only network

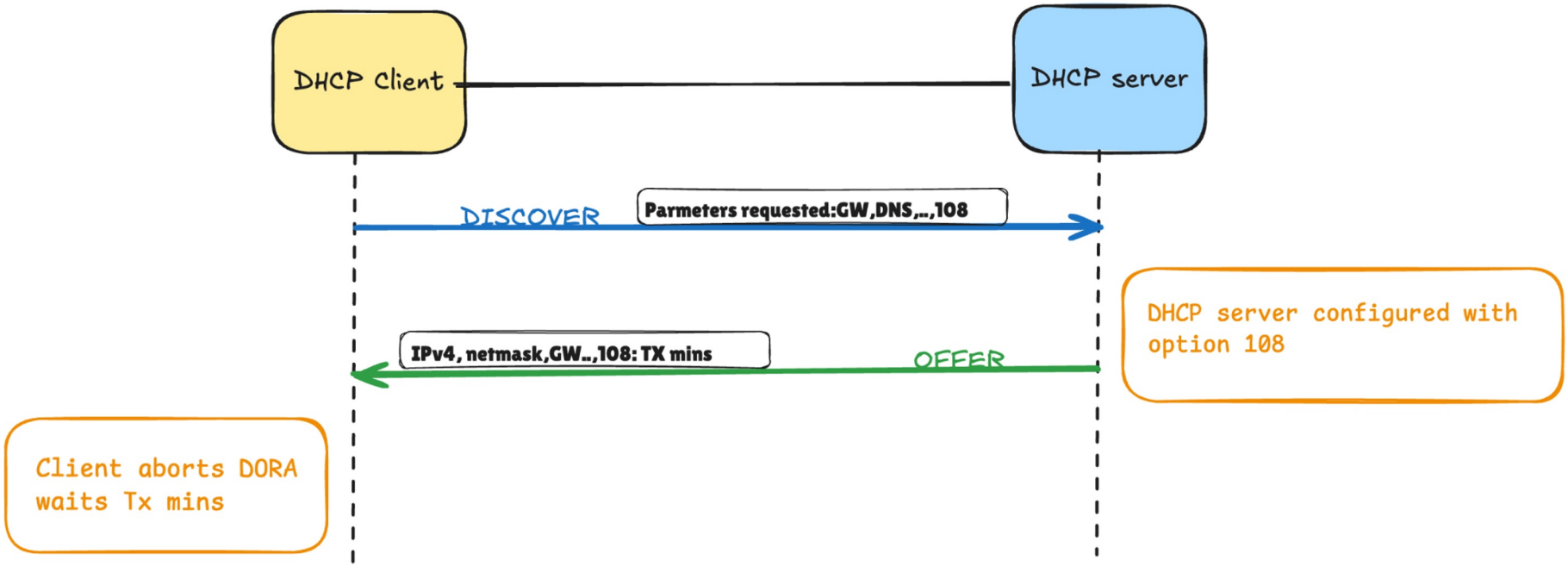
DHCPv4 option 108

- Meta's Wi-Fi network is predominantly dual-stacked
  - But can turn into single stack during busy hours
  - IPv4 over-utilization causes client issues
    - Hard WiFi disconnects on Linux
    - Applications not relying on DNS64
    - No IPv6 fallback if IPv4 isn't available
  - Fixing those issues is a game of whack-a-mole

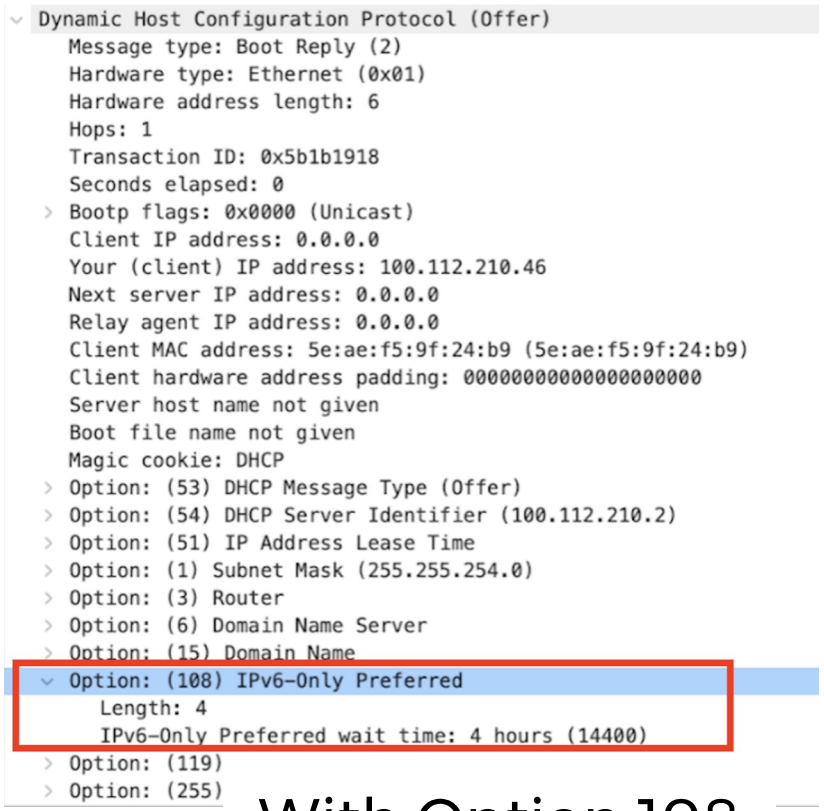
DHCPv4 option 108 → What is it?

- Tells the client to disable its IPv4 stack
- Rides over IPv4, thus requires some available leases
- Clients also enable CLAT (Customer-Side Translator)

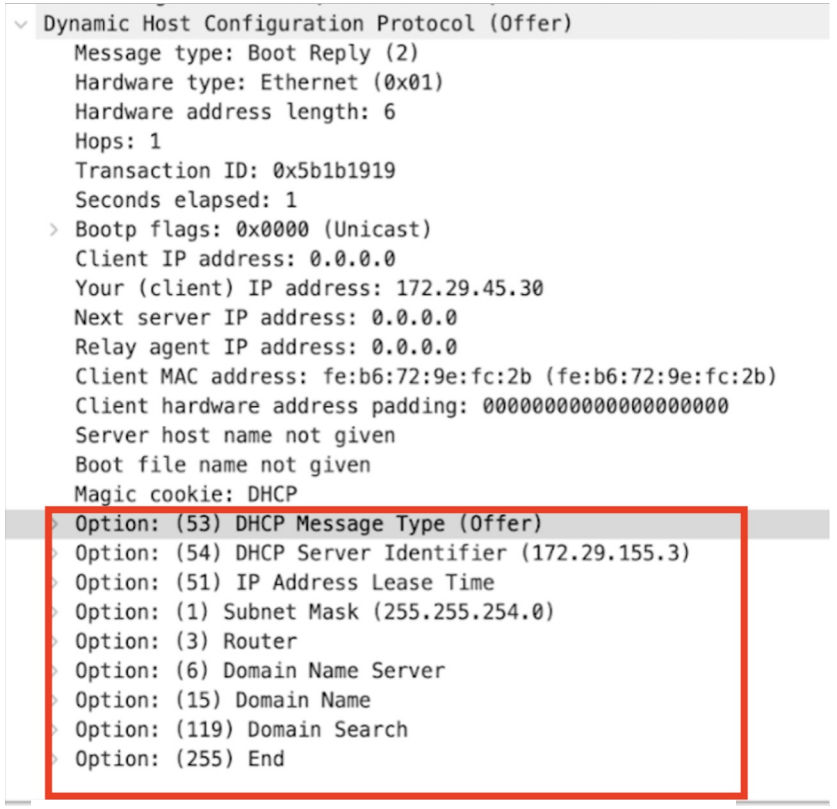
DHCPv4 option 108 → What is it?



With and Without Option 108



With Option 108

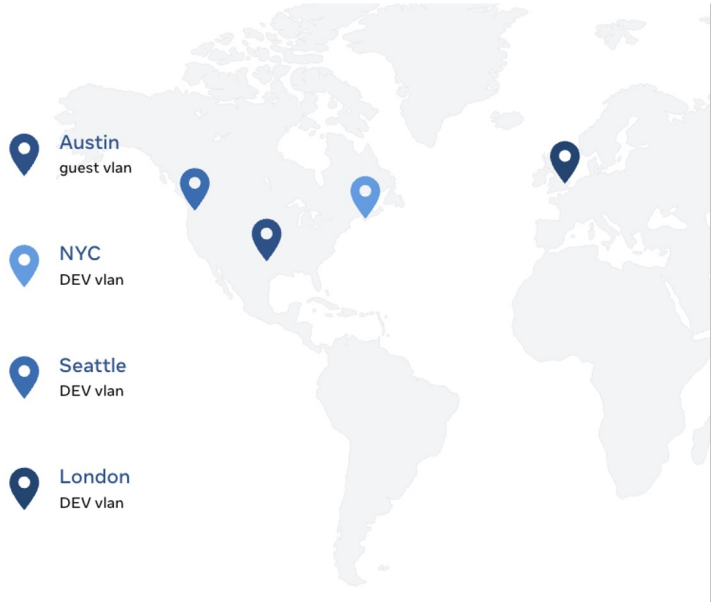


Without Option 108

NANOG™

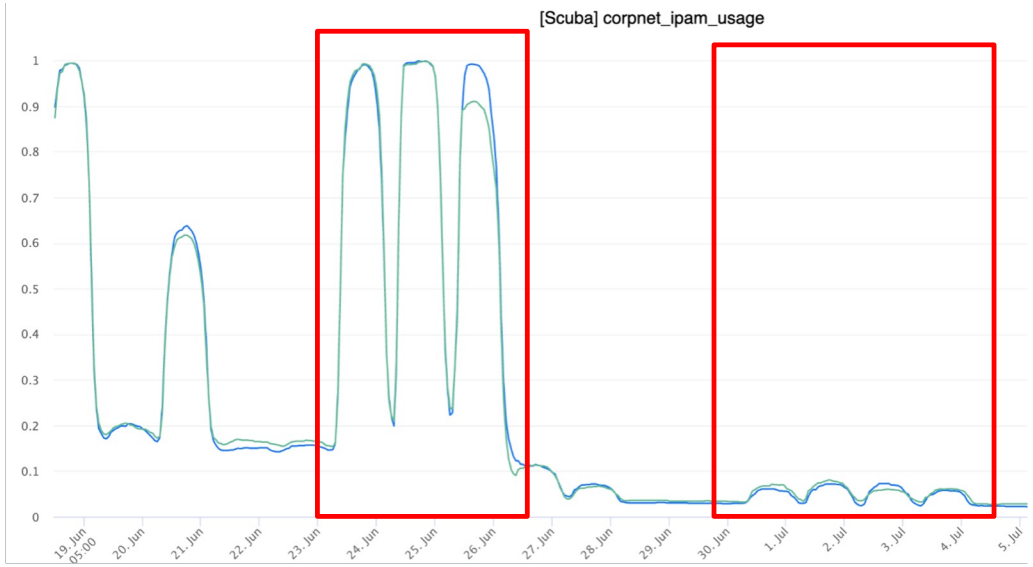
DHCP option 108 → Enablement

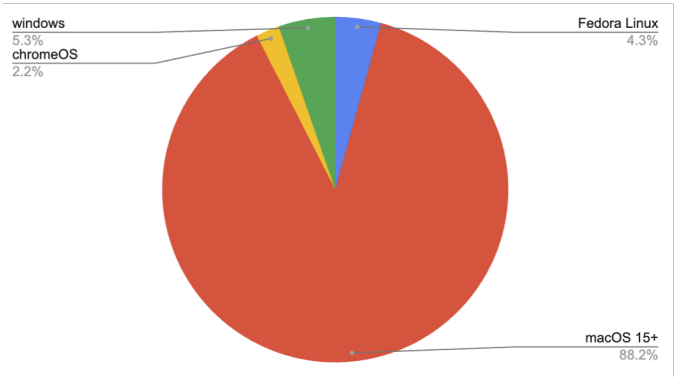
- On overutilized subnets:
  - Compared utilization before/after
  - Tested most common applications
  - Listened for people screaming
- Tested client platforms
  - MacOS 13+
  - Windows 10+
  - Linux Fedora
  - ChromeOS
  - iOS 16+, Android 14+



DHCP option 108 in action

- Employee network





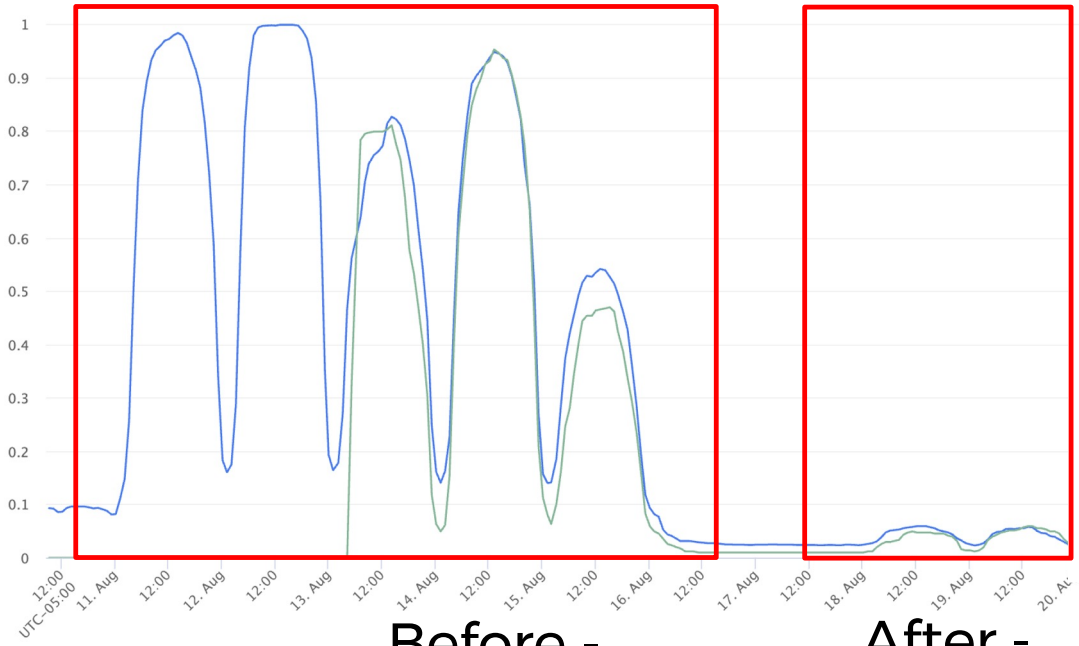
Before - 100% IPv4 utilization on a /21

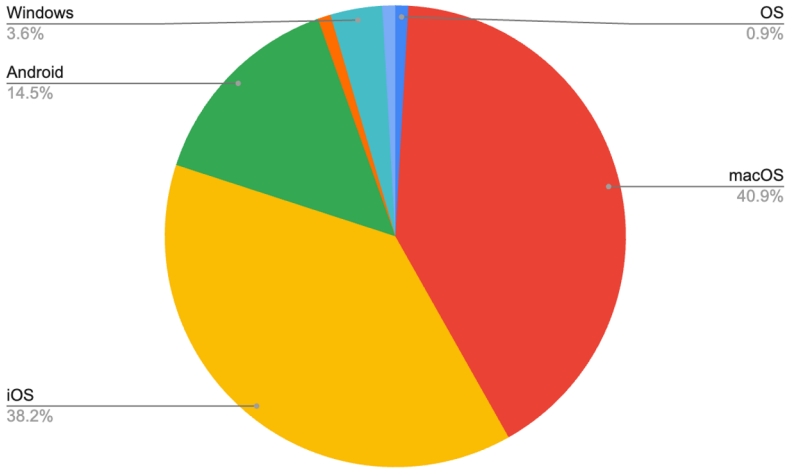
After - ~5% IPv4 utilization

NANOG™

DHCP option 108 in action

- Guest network





Before - 100% IPv4 utilization

After - ~3% IPv4 utilization

NANOG™

DHCP option 108 → Takeaways

- Effective solution in an Apple/Android mostly environment

platform	Version	Implements DHCP option 108	Supports IPv4 literals
Apple macOS	Ventura and above	Yes	Yes with CLAT (as long as PREF64 is learned via rfc7050)
Apple iOS	16 and above	Yes	Yes with CLAT (as long as PREF64 is learned via rfc7050)
Google Android	14 and above	Yes	Yes
Google chromeOS	136 and under	Yes	No (bug 389342045)
Windows	10 & 11	No	N/A
Linux Fedora	42 and under	No	N/A

DHCP option 108 → Takeaways

- It reduces the need for IPv4
- More unknowns in guest vlans
- Easy to deploy

Lessons Learned

Lessons Learned

1. For both vendors that we use, it was important that:
    - APs have a factory image which would allow them to join a controller on an IPv6-only network
        - Specifically: IPv6 & DHCPv6 option 52 support out of the box
    - Promises of “we’ll load the right image before the APs ship to your site” were not always kept

Lessons Learned

2. Most vendors treat IPv6 support as something they add support for later, and not from day 1
   - Even if they have folks contributing to the RFCs for IPv6
   - But they eventually got there after much nudging

Lessons Learned

3. Doing IPv6-only for clients is tricky if you don't have full control over your client base
    - Even so, we've run into weird issues & dependencies, especially when going from one major OS version to another

Lessons Learned

4. Issues with fragmentation over IPv6 & ACLs
    - Some fragmentation started happening once we migrated our APs to using IPv6
    - Fragments would be lost if not explicitly allowed in ACLs (especially during EAP-TLS auth, which would timeout)

Lessons Learned

```
3968 EAP      1126 Request, TLS EAP (EAP-TLS)
3970 EAP       102 Response, TLS EAP (EAP-TLS)
3971 EAP      1126 Request, TLS EAP (EAP-TLS)
3972 EAP       102 Response, TLS EAP (EAP-TLS)
3973 EAP      1126 Request, TLS EAP (EAP-TLS)
3974 EAP       102 Response, TLS EAP (EAP-TLS)
3975 TLSv1.2   297 Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
3976 IPv6     1510 IPv6 fragment (off=0 more=y ident=0x0000962b nxt=47)
3977 EAP       148 Response, TLS EAP (EAP-TLS)
3980 EAP       138 Request, TLS EAP (EAP-TLS)
3981 IPv6     1510 IPv6 fragment (off=0 more=y ident=0x0000962c nxt=47)
3982 EAP       148 Response, TLS EAP (EAP-TLS)
3983 EAP       138 Request, TLS EAP (EAP-TLS)
3984 IPv6     1510 IPv6 fragment (off=0 more=y ident=0x0000962d nxt=47)
3985 EAP       148 Response, TLS EAP (EAP-TLS)
3990 EAP       138 Request, TLS EAP (EAP-TLS)
```

Lessons Learned

5. DHCP option 108 may make more sense on our employee network; less so for guest
   - Due to better control over clients on the employee network

Thank you

NANOG™