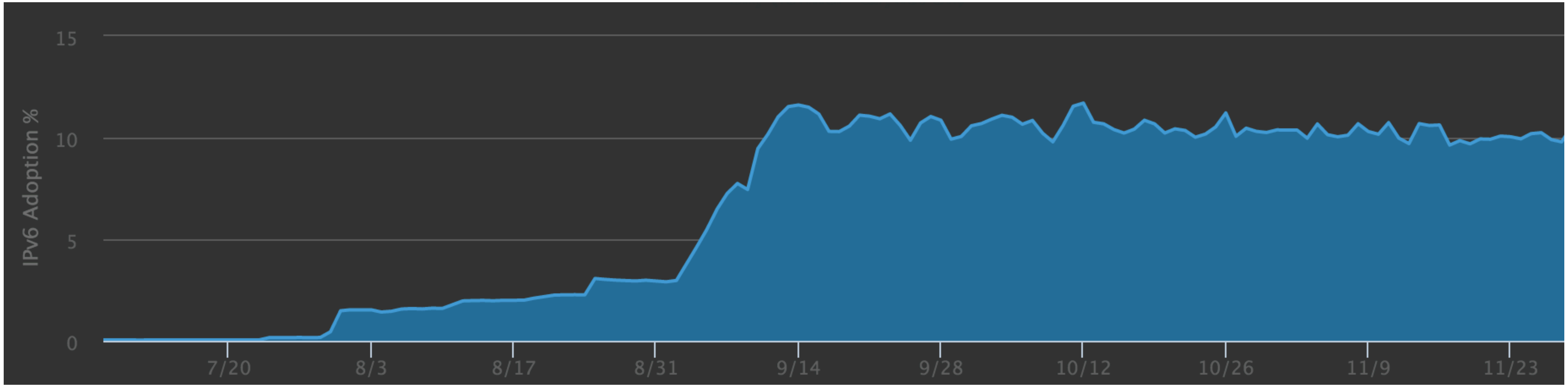
IPv6 – Recursive DNS

Dual Stack Client Behaviour

Lucky Man

Where We Got To

- ~1M Dual Stack Users September 2015



- Why did we stop?

IPv4 or IPv6?

- RDNSS load increase did not track our IPv6 enablement program.
- We know about Happy Eyeballs (RFC6555)
  - Fire TCP SYN to both v4 and v6 addresses.
  - Delay v4 SYN to give v6 a head start.

- But when does an OS or application ask for a AAAA?
  - Linux, Windows & OSX: When the host has a non-link-local, or non-Teredo address.
  - ULA – Unique Local Addressing.

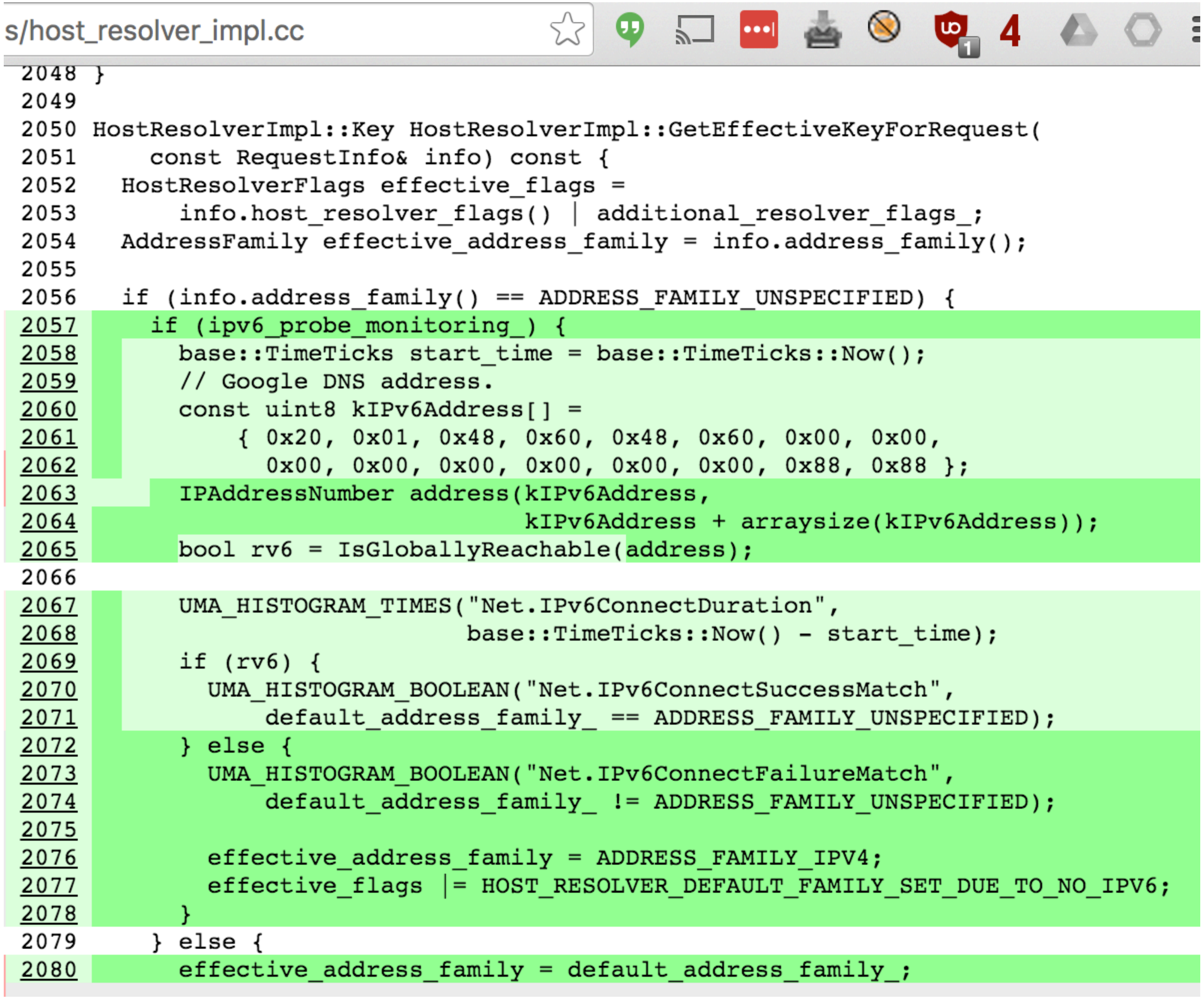
- All of our CPEs advertise a ULA prefix
  - Allows intraLAN IPv6 communication even if the WAN drops, or GUA prefix changes.

Conversely

- Android & Chrome really don’t like ULA
  - Won’t even ask for a AAAA when only a ULA is present without a GUA
  - When presented with a AAAA for a ULA address, will prefer global IPv4 over IPv6 ULA source.

- Actually Chrome is quite broken:
  - Checks for a route to Google’s v6 DNS to decide if it should attempt an IPv6 connection.

WTF Chrome



```
}

HostResolverImpl::Key HostResolverImpl::GetEffectiveKeyForRequest(
    const RequestInfo& info) const {
  HostResolverFlags effective_flags =
      info.host_resolver_flags() | additional_resolver_flags_;
  AddressFamily effective_address_family = info.address_family();

  if (info.address_family() == ADDRESS_FAMILY_UNSPECIFIED) {
    if (ipv6_probe_monitoring_) {
      base::TimeTicks start_time = base::TimeTicks::Now();
      // Google DNS address.
      const uint8 kIPv6Address[] =
          { 0x20, 0x01, 0x48, 0x60, 0x48, 0x60, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x88, 0x88 };
      IPAddressNumber address(kIPv6Address,
                              kIPv6Address + arraysize(kIPv6Address));
      bool rv6 = IsGloballyReachable(address);

      UMA_HISTOGRAM_TIMES("Net.IPv6ConnectDuration",
                          base::TimeTicks::Now() - start_time);
      if (rv6) {
        UMA_HISTOGRAM_BOOLEAN("Net.IPv6ConnectSuccessMatch",
            default_address_family_ == ADDRESS_FAMILY_UNSPECIFIED);
      } else {
        UMA_HISTOGRAM_BOOLEAN("Net.IPv6ConnectFailureMatch",
            default_address_family_ != ADDRESS_FAMILY_UNSPECIFIED);

        effective_address_family = ADDRESS_FAMILY_IPV4;
        effective_flags |= HOST_RESOLVER_DEFAULT_FAMILY_SET_DUE_TO_NO_IPV6;
      }
    } else {
      effective_address_family = default_address_family_;
```

Conclusions & Where We’re At Now

- You’re going to need more RDNSS capacity.
- Or perhaps **draft-vavrusa-dnsop-aaaa-for-free** eventually?
  - “Providing AAAA records for free with QTYPE=A”

- OSX’ Safari
  - Reports that Safari can get “stuck” when host has only a ULA and no GUA or v6 default route.
  - Haven’t been able to reproduce in the lab yet.

- Other applications/devices also behave poorly with only ULA (no default route or GUA)
  - Xbox One – Fifa16
  - Sony Bravia TV