# 464XLAT: Breaking Free of IPv4

Cameron.Byrne @ T-Mobile.com

NANOG 61

June 2014

# Goals of Talk

1. Declare victory for IPv6
2. Explain IPv6-only approach at T-Mobile US
3. Discuss risks related to IPv4-only operations for content providers and App makers

# Background

- T-Mobile US is a GSM / UMTS / LTE provider in the USA with 49 Million subscribers
- In 2008, T-Mobile launched the first Android phone. This dramatically changed the mobile data dynamics – more devices, connected for a longer time, all needing IP addresses
- T-Mobile embraced the concept of IPv6-only, since dual-stack required IPv4 that was not available
- NAT64 / DNS64 was a good solution that did not require IPv4 on each client, but some applications failed to work on IPv6-only networks. It is not acceptable to break Spotify or Whatsapp, applications that require IPv4
- T-Mobile, in partnership with NEC and JPIX, documented 464XLAT in the IETF as RFC6877 to overcome the limitations of NAT64 by adding a NAT46 into the client (CLAT)
- Android 4.3 introduced support for 464XLAT in October 2013
- **T-Mobile US changed the default settings for all Android 4.3+ phones to be IPv6-only / 464XLAT**

# Results Are Important

- T-Mobile US launched 8 Android phone models with 464XLAT as the default in the last 8 months, all Android 4.3+ phones will be 464XLAT in the future at T-Mobile US
- 8 million unique IPv6 subscribers in the first 8 months are active on the network
- http://www.worldipv6launch.org/measurements/ measurements show 27% of all T-Mobile connections to dual-stack sites are now IPv6
- **Over 50% of IPv6-user traffic is end-to-end IPv6 (no translation needed) ← *This saves money and makes the network simpler***

# 27% of T-Mobile US Connections use IPv6 to Dual-Stack content

www.worldipv6launch.org/measurements/

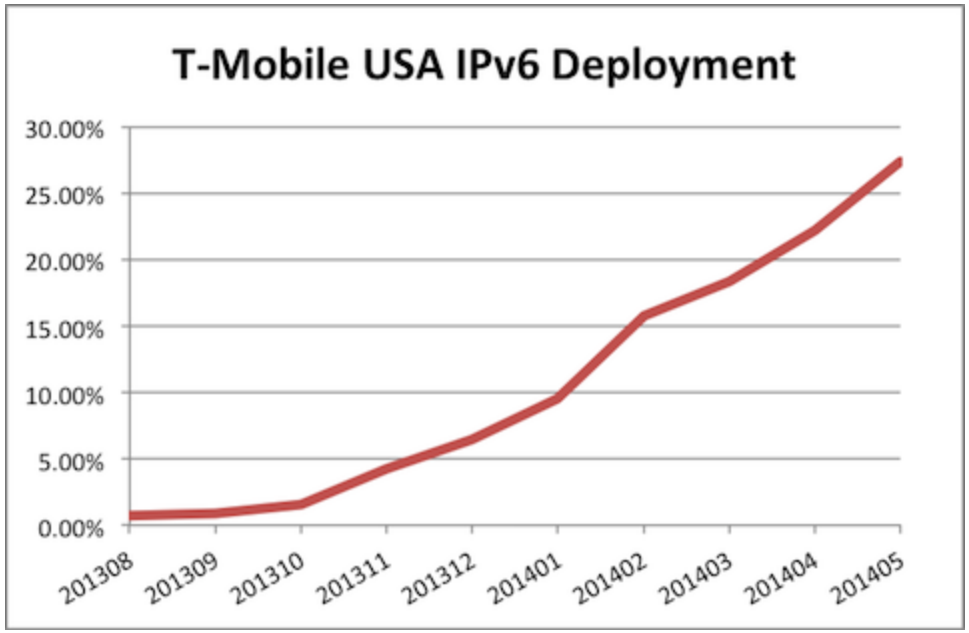
**Network operator measurements, 20th May 2014 (notes)**

Show 25 entries  Search: T-Mobile

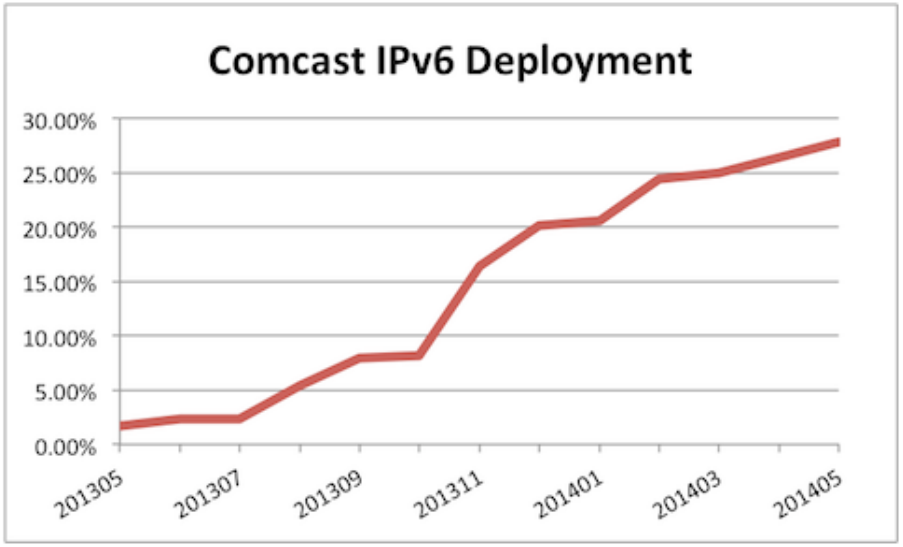
| Participating Network | ASN(s) | IPv6 deployment |
|---|---|---|
| T-Mobile USA | 21928 | 27.37% |

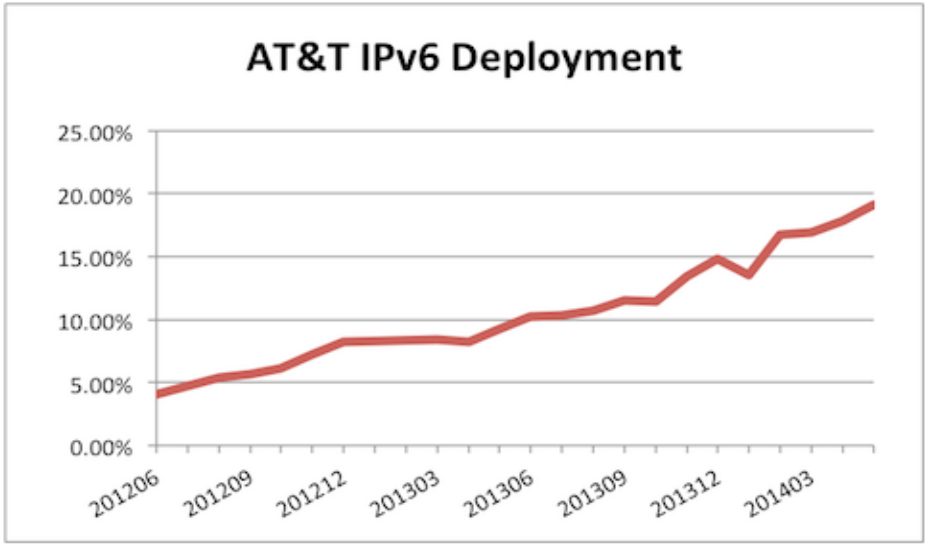
Showing 1 to 1 of 1 entries (filtered from 207 total entries)

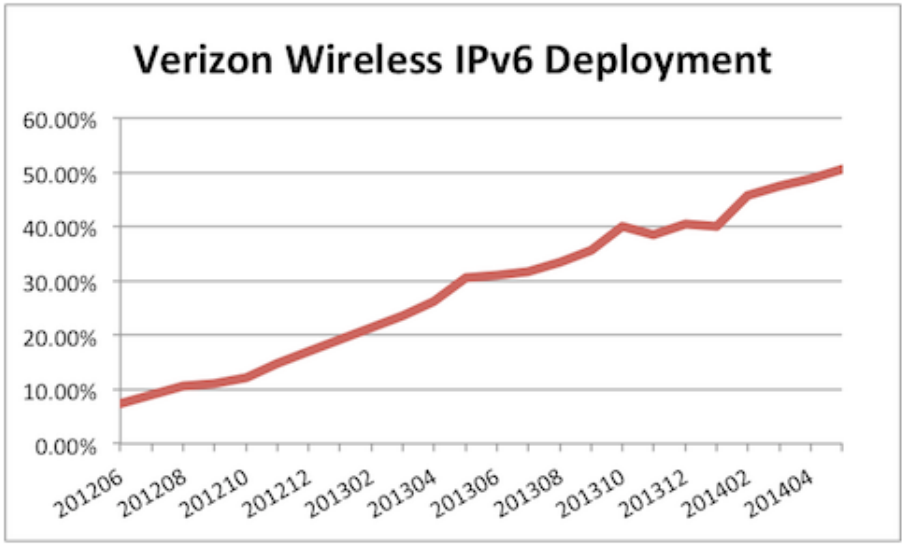
First Previous 1 Next Last

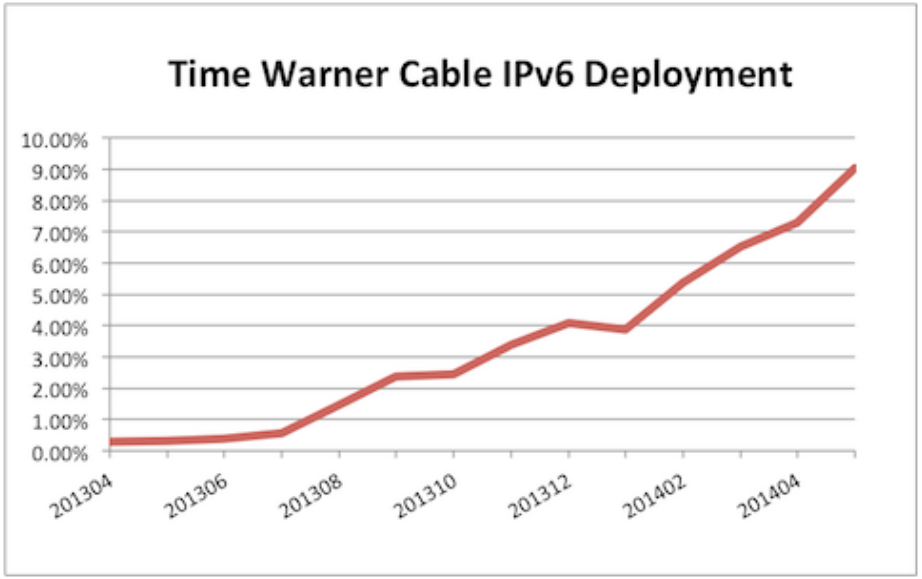


# Other major networks on IPv6



# 464XLAT allows for full functionality on IPv6-only networks

- Dual-stack does not solve the IPv4 number scarcity issue
- IPv6-only + NAT64/DNS64 is very good, but not good enough for full IPv4 replacement (web and email work, but Skype does not work)
- IPv6-only + 464XLAT
  - Solves IPv4 numbering issue by not assigning IPv4 to clients
  - Decouples edge growth from IPv4 availability
  - IPv4-only applications like Skype work on an IPv6-only network because 464XLAT translates IPv4 on the phone to IPv6 on the network

# Why not MAP or DS-lite

- Mobile networks don’t use DHCP, so no way to setup MAP or DS-lite without some heavy lifting in protocols and standards
- Purely stateless solutions like MAP require many IPv4 addresses to be statically assigned to the MAP domain
- Stateful NAT64 allows greater multiplexing of IPv4 addresses, even port overloading to get beyond 64,000 sessions per IPv4

# IPv6 deployment is achievable

- T-Mobile USA did not spend any CapEx on IPv6
- Only introduce 464XLAT on new phones, so we do not disrupt any existing services, leverage normal phone QA process
- Innovative thinking helps reduce deployment costs (hash 128 bit numbers into 32 bit fields in billing records)
- IPv6 will save money in your network (less NAT/CGN, no need to buy IPv4 addresses, …)

# Which Platforms Supports 464XLAT Today?

- YES
- Android 4.3+

- NO
- Blackberry
- Apple
- Windows Phone (?)

# THE TECHNICAL DETAILS

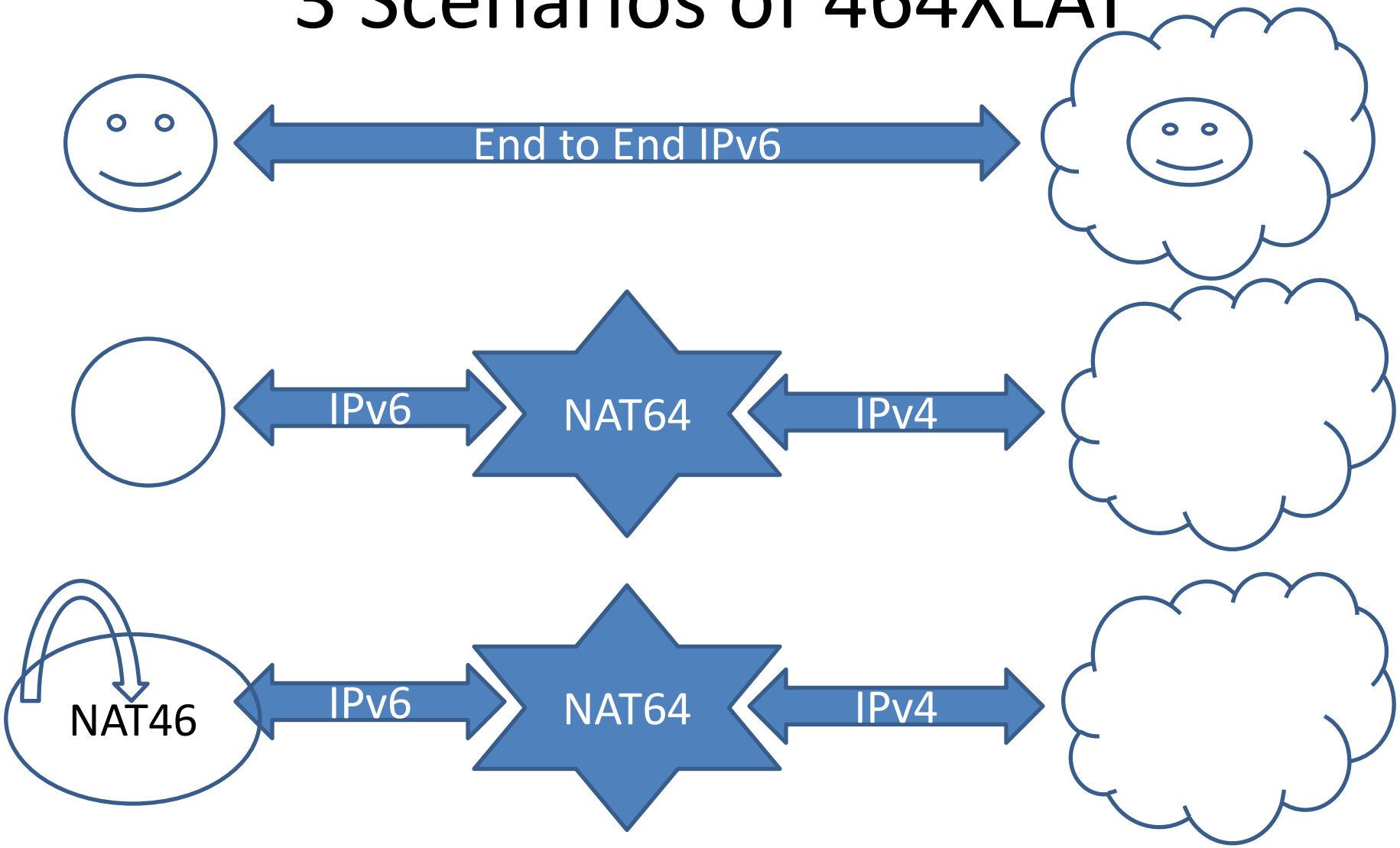
# 464XLAT is just a set of building blocks

- Stateless NAT64 (RFC6145)
  - Client side translation CLAT (NAT4->6)
- Statefull NAT64 (RFC6146)
  - Provider site translation PLAT (NAT6->4)
- DNS64 (RFC 6147)
  - When the FQDN does not have a AAAA record, DNS64 dynamically creates one that allows the client to use IPv6 and the network translates from IPv6 to IPv4 at the NAT64
- Prefix64 Discovery (RFC 7050)
  - Queries for the well-known FQDN ipv4only.arpa, which is by definition IPv4-only. If there is a AAAA response provided, then it is known that a DNS64 is in the path

# 3 Scenarios in a 464XLAT network

1. End-to-end IPv6:  Facebook, Google, Wikipedia, Yahoo, Youtube … **IPv6->IPv6**
2. Application supports IPv6 (web browser) but the server is only IPv4 (www.myspace.com, www.twitter.com, www.amazon.com,  …), so DNS64/NAT64 translates **IPv6->IPv4**
3. Application does not support IPv6 (Whatsapp, Spotify, …), the client must provide a stateless NAT46 to the application and stateful NAT64 must be in the network: **IPv4->IPv6->IPv4**

# 3 Scenarios of 464XLAT



# How does Stateless NAT64 work?

- Algorithmically map IPv4 addresses to IPv6 addresses, bidirectional, 1 to 1
  - Not dynamic
  - Deterministic
  - Maps all of IPv4's 32 bits into an IPv6 /96 (or larger prefix)
- Defined in RFC6145
- Example
  - 2001:db8::10.1.1.1 <-> 10.1.1.1
  - 2001:db8::10.2.2.2 <-> 10.2.2.2
  - 2001:db8::www.example.com <-> ipv4 www.example.com

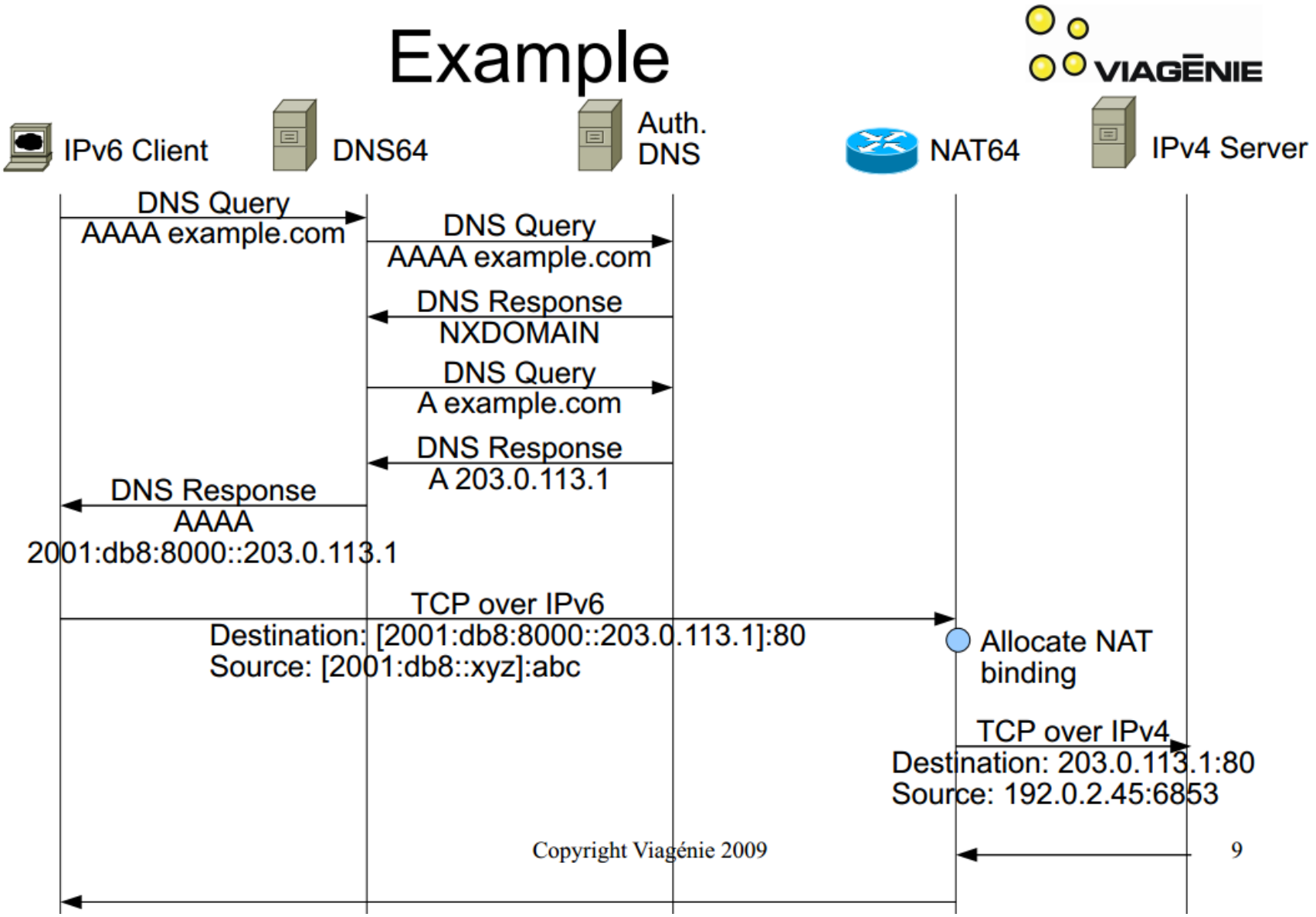
# How does Stateful NAT64 work?

- Dynamically translate IPv6 packets to IPv4 packets
  - Dynamic
  - Not deterministic (translation based on available IPv4 pool)
  - Translation state is short-lived and based on session creation and termination
- Defined in RFC6146
- Example
  - Before translation
    - TCP source 2001:db8:abcd::ffff port 555 # client address
    - TCP destination 2001:db8:1234::10.1.1.1 port 80 # NAT64 address
  - After translation
    - TCP source 192.168.1.1 port 555 # 192.168.1.1 available from NAT64 pool
    - TCP destination 10.1.1.1 port 80 # Last 32 bits of IPv6 destination

# How does DNS64 work?

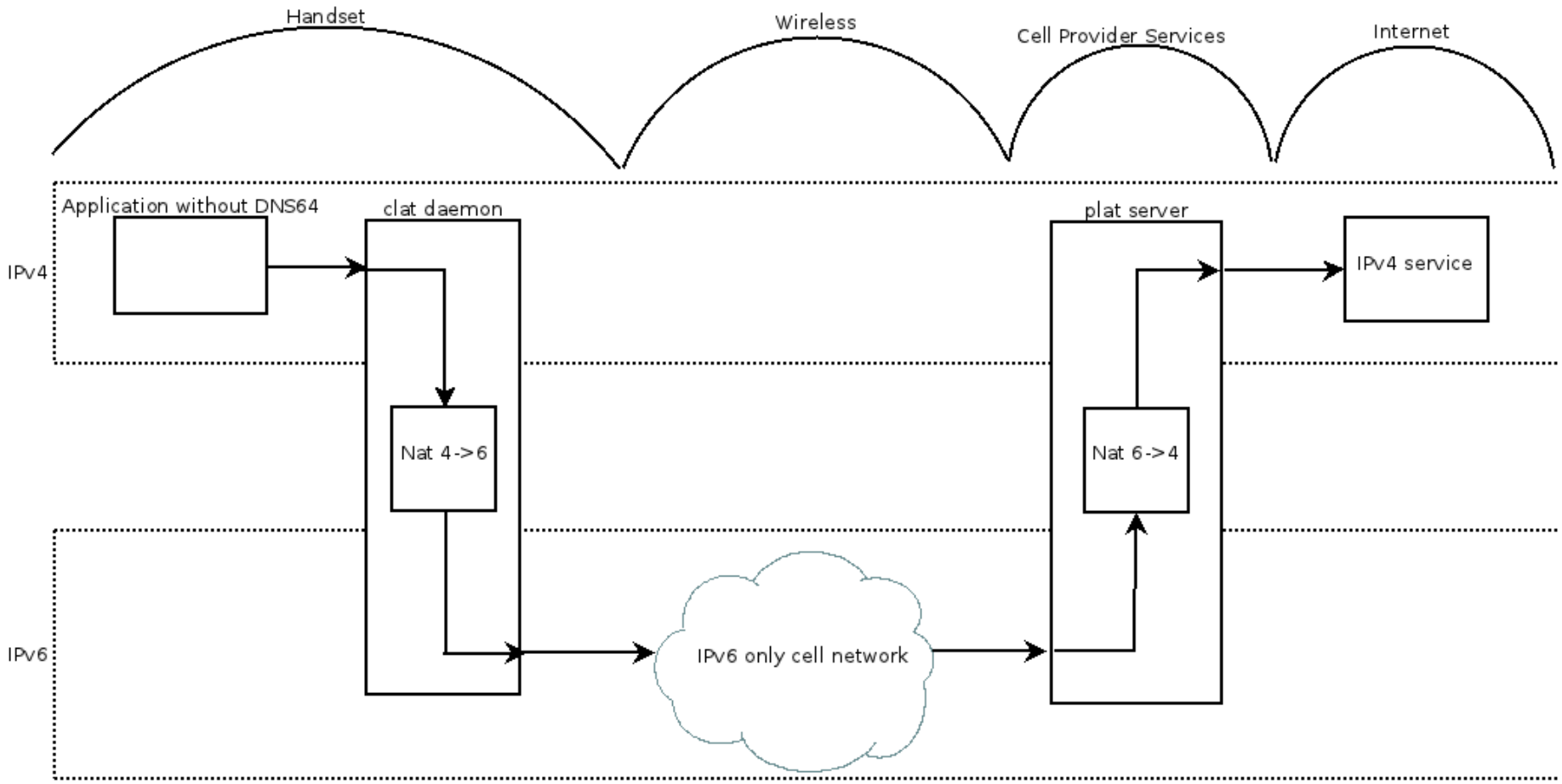
- When an FQDN does not have a AAAA record, the DNS64 will synthetically create one based on a network defined Pref64
- The pref64 is a prefix hosted on the NAT64 for translation
- Example without DNS64
  - Query = a and aaaa for www.example.com
  - Answer = a = 10.1.1.1, aaaa = NO ERROR
- Example with DNS64
  - Query = a and aaaa for www.example.com
  - Answer = a = 10.1.1.1 AND aaaa = 2001:db8::10.1.1.1

# Example




Copyright Viagénie 2009

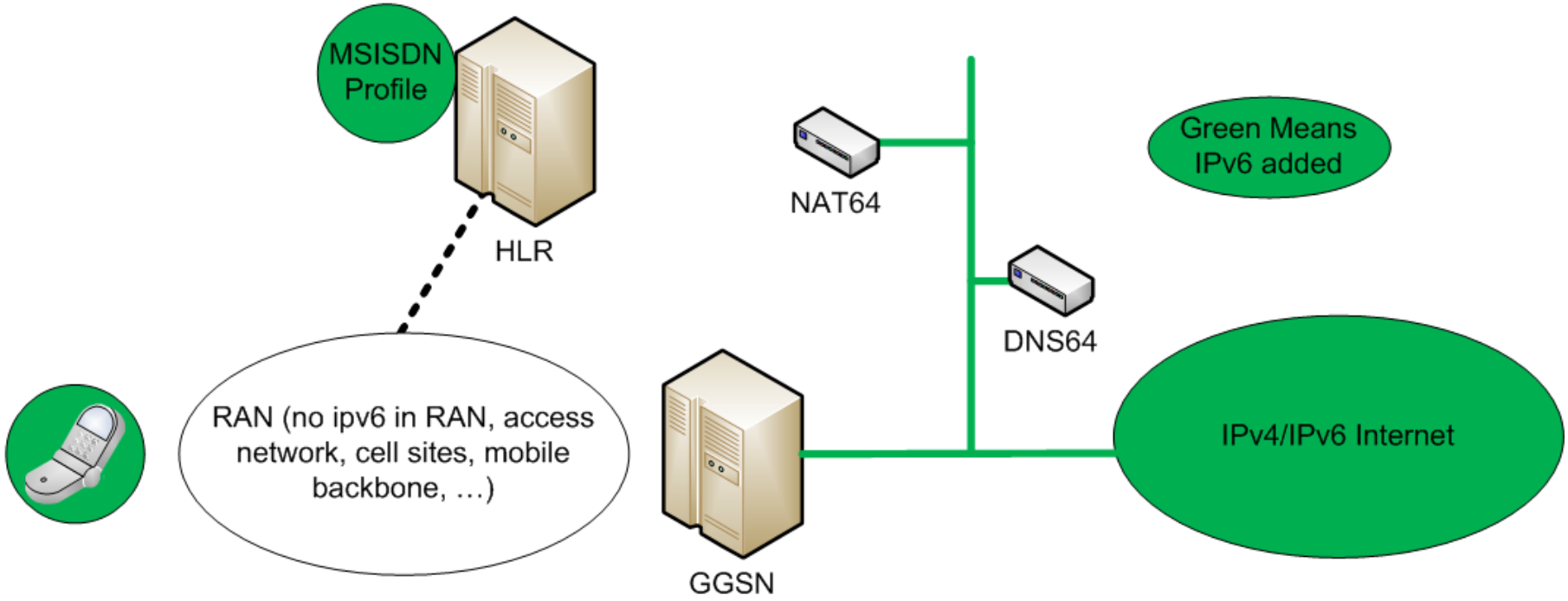
# How to make EVERYTHING work on IPv6-only?



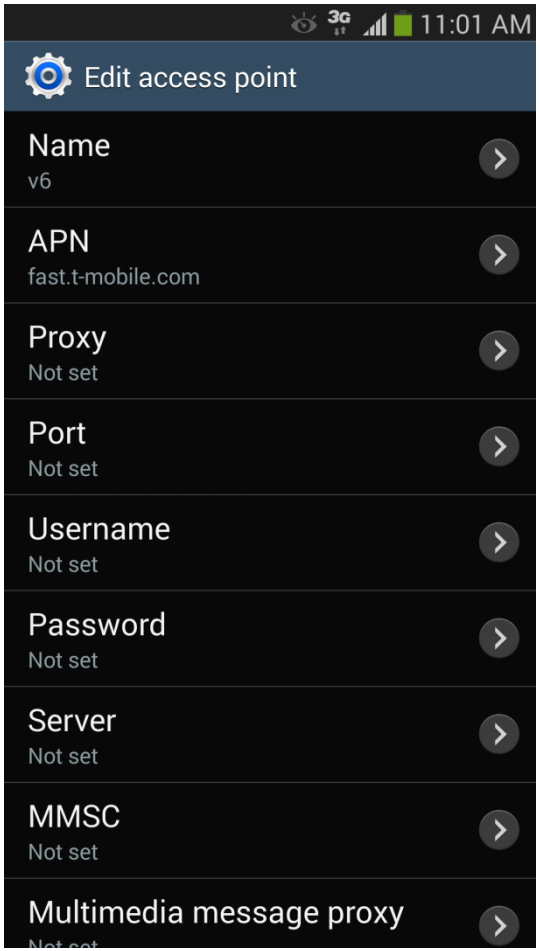
# Zoom Out: What does this look like in the context of 3GPP GSM / UMTS / LTE ?

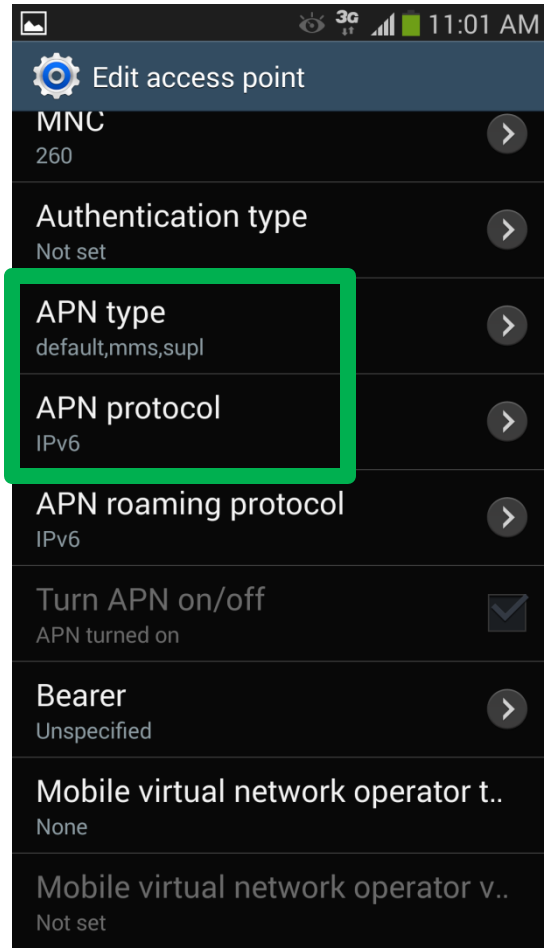
# High Level View of IPv6 deployment:

Phone, HLR profile, GGSN, NAT64, IPv6 ISP



# Zoom in: What does the phone configuration look like: APN Settings



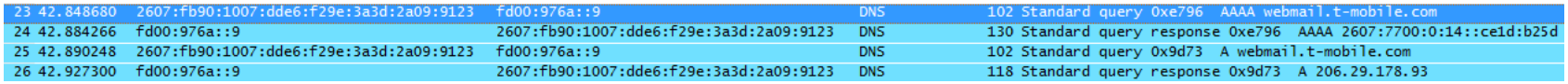




In Android 4.3, "APN Protocol IPv6" for the "APN Type default" triggers the use of 464XLAT by default

IPv6 = 464XLAT

# TIME FOR WIRESHARK

# Like most things, we start with DNS

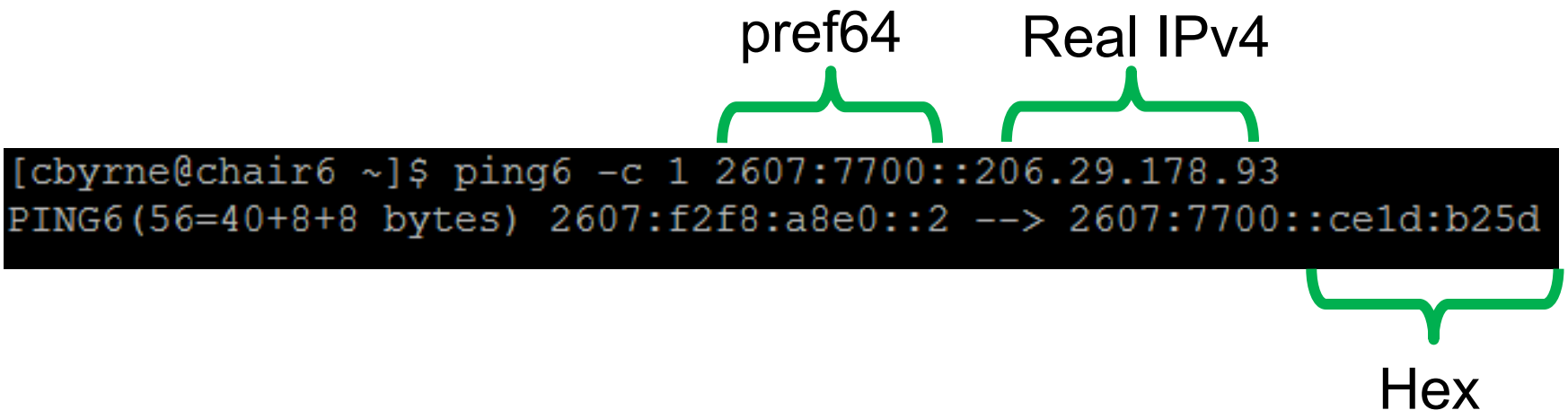


| | | | | | |
|---|---|---|---|---|---|
| 23 | 42.848680 | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | fd00:976a::9 | DNS | 102 Standard query 0xe796 AAAA webmail.t-mobile.com |
| 24 | 42.884266 | fd00:976a::9 | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | DNS | 130 Standard query response 0xe796 AAAA 2607:7700:0:14::ce1d:b25d |
| 25 | 42.890248 | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | fd00:976a::9 | DNS | 102 Standard query 0x9d73 A webmail.t-mobile.com |
| 26 | 42.927300 | fd00:976a::9 | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | DNS | 118 Standard query response 0x9d73 A 206.29.178.93 |

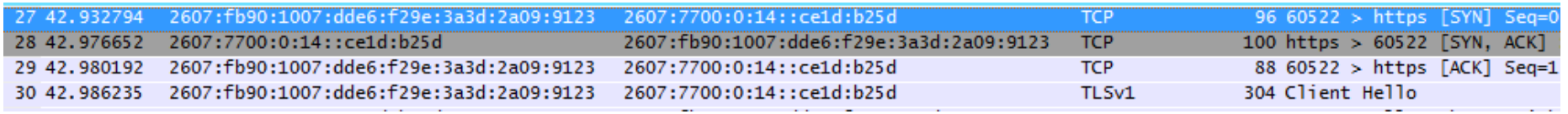
- The client is IPv6-only towards the network, but the host OS thinks it is dual-stack since it has an IPv4 CLAT interface and a native IPv6 radio interface
- So, the client does a query for DNS “A” and “AAAA” records
- The DNS64 responds with a synthesized AAAA and the real A
- The synthesized AAAA = Pref64 + real IPv4

# Quick Check

- Does the synthesized AAAA match the pref64 + real A?



# Next, the UE selects the IPv6 DNS response, and starts TCP



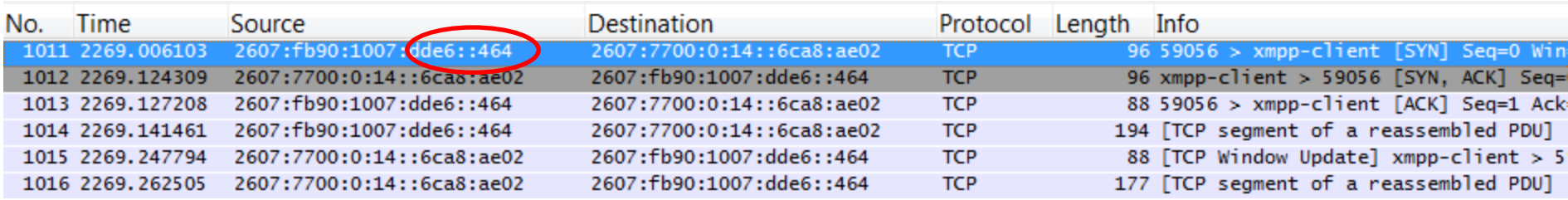
| | | | | | |
|---|---|---|---|---|---|
| 27 | 42.932794 | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | 2607:7700:0:14::ce1d:b25d | TCP | 96 60522 > https [SYN] Seq=0 |
| 28 | 42.976652 | 2607:7700:0:14::ce1d:b25d | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | TCP | 100 https > 60522 [SYN, ACK] |
| 29 | 42.980192 | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | 2607:7700:0:14::ce1d:b25d | TCP | 88 60522 > https [ACK] Seq=1 |
| 30 | 42.986235 | 2607:fb90:1007:dde6:f29e:3a3d:2a09:9123 | 2607:7700:0:14::ce1d:b25d | TLSv1 | 304 Client Hello |

- From the client perspective, this is a native IPv6 end-to-end flow
- But, we know that the DNS is a synthesized AAAA and the client is actually sending its packets to the NAT64 for IPv6->IPv4 stateful translation
- This is just DNS64 / NAT64, no client-side translation needed for  this scenario

# The full case of 464XLAT double translation: WhatsApp

```
Queries
  e8.whatsapp.net: type AAAA, class IN
      Name: e8.whatsapp.net
      Type: AAAA (IPv6 address)
      Class: IN (0x0001)
Answers
  e8.whatsapp.net: type AAAA, class IN, addr 2607:7700:0:14::b8ad:a1ba
      Name: e8.whatsapp.net
      Type: AAAA (IPv6 address)
      Class: IN (0x0001)
      Time to live: 48 minutes, 25 seconds
      Data length: 16
      Addr: 2607:7700:0:14::b8ad:a1ba
  e8.whatsapp.net: type AAAA, class IN, addr 2607:7700:0:14::3216:e142
      Name: e8.whatsapp.net
      Type: AAAA (IPv6 address)
      Class: IN (0x0001)
      Time to live: 48 minutes, 25 seconds
      Data length: 16
      Addr: 2607:7700:0:14::3216:e142
  e8.whatsapp.net: type AAAA, class IN, addr 2607:7700:0:14::6ca8:ae02
      Name: e8.whatsapp.net
      Type: AAAA (IPv6 address)
      Class: IN (0x0001)
      Time to live: 48 minutes, 25 seconds
      Data length: 16
      Addr: 2607:7700:0:14::6ca8:ae02
```

# SYN is sent from the CLAT address



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1011 | 2269.006103 | 2607:fb90:1007:dde6::464 | 2607:7700:0:14::6ca8:ae02 | TCP | 96 | 59056 > xmpp-client [SYN] Seq=0 Win |
| 1012 | 2269.124309 | 2607:7700:0:14::6ca8:ae02 | 2607:fb90:1007:dde6::464 | TCP | 96 | xmpp-client > 59056 [SYN, ACK] Seq= |
| 1013 | 2269.127208 | 2607:fb90:1007:dde6::464 | 2607:7700:0:14::6ca8:ae02 | TCP | 88 | 59056 > xmpp-client [ACK] Seq=1 Ack |
| 1014 | 2269.141461 | 2607:fb90:1007:dde6::464 | 2607:7700:0:14::6ca8:ae02 | TCP | 194 | [TCP segment of a reassembled PDU] |
| 1015 | 2269.247794 | 2607:7700:0:14::6ca8:ae02 | 2607:fb90:1007:dde6::464 | TCP | 88 | [TCP Window Update] xmpp-client > 5 |
| 1016 | 2269.262505 | 2607:7700:0:14::6ca8:ae02 | 2607:fb90:1007:dde6::464 | TCP | 177 | [TCP segment of a reassembled PDU] |

Remember, we set the clatd.conf to use the IID of ::464 for CLAT translations

- IPv6 is widely deployed today!

1 google.com
Enables users to search the world's information, including webpages, images, and videos. Offers... More

2 facebook.com
A social utility that connects people, to keep up with friends, upload photos, share links and ... More

3 youtube.com
YouTube is a way to get your videos to the people who matter to you. Upload, tag and share your... More

4 yahoo.com
A major internet portal and service provider offering search results, customizable content, cha... More

5 amazon.com
Amazon.com seeks to be Earth's most customer-centric company, where customers can find and disc... More

6 linkedin.com
A networking tool to find connections to recommended job candidates, industry experts and busin... More

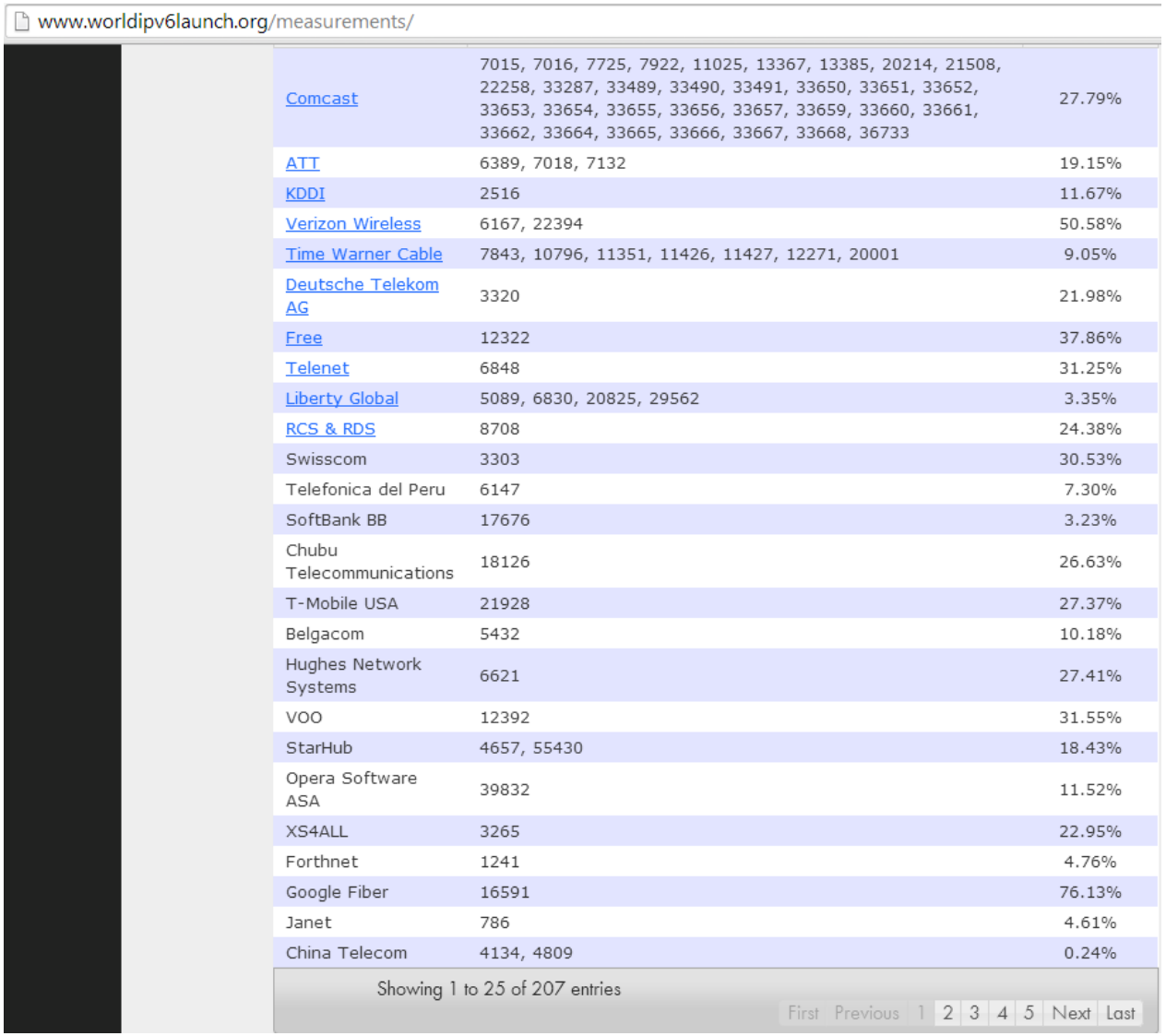
7 wikipedia.org
A free encyclopedia built collaboratively using wiki software. (Creative Commons Attribution-Sh... More

8 ebay.com
International person to person auction site, with products sorted into categories.

9 twitter.com
Social networking and microblogging service utilising instant messaging, SMS or a web interface.

10 bing.com
Search engine developed by Microsoft. Features web, image, video, local, news, and product search.

# Major eye-ball networks have enabled IPv6 – T-Mobile US, Comcast, Verizon, AT&T, …

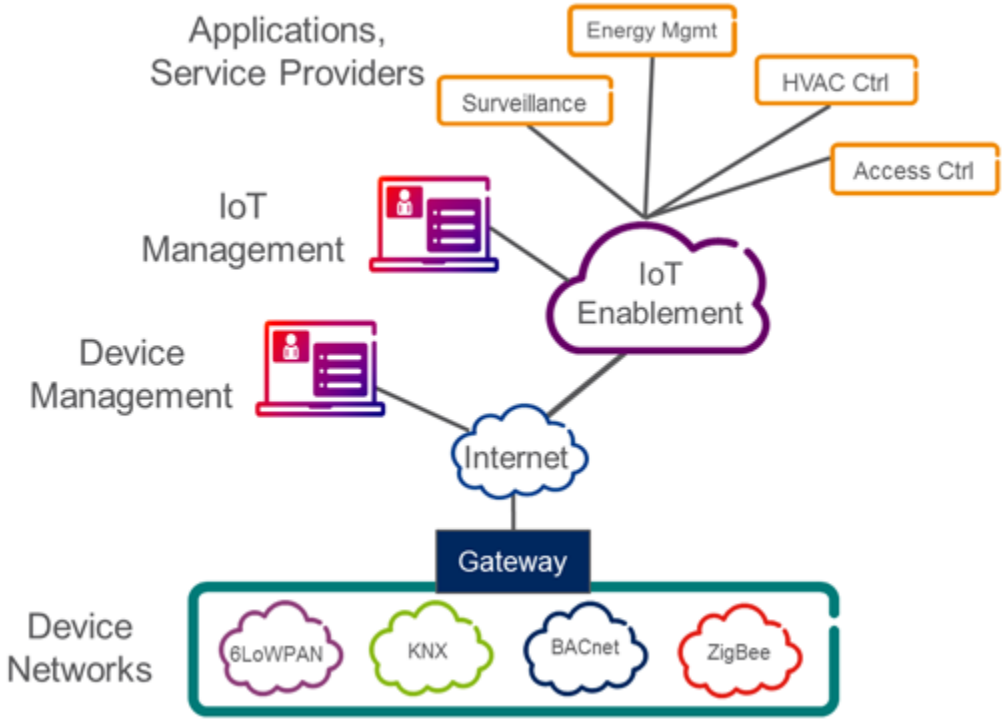



| | | |
|---|---|---|
| Comcast | 7015, 7016, 7725, 7922, 11025, 13367, 13385, 20214, 21508, 22258, 33287, 33489, 33490, 33491, 33650, 33651, 33652, 33653, 33654, 33655, 33656, 33657, 33659, 33660, 33661, 33662, 33664, 33665, 33666, 33667, 33668, 36733 | 27.79% |
| ATT | 6389, 7018, 7132 | 19.15% |
| KDDI | 2516 | 11.67% |
| Verizon Wireless | 6167, 22394 | 50.58% |
| Time Warner Cable | 7843, 10796, 11351, 11426, 11427, 12271, 20001 | 9.05% |
| Deutsche Telekom AG | 3320 | 21.98% |
| Free | 12322 | 37.86% |
| Telenet | 6848 | 31.25% |
| Liberty Global | 5089, 6830, 20825, 29562 | 3.35% |
| RCS & RDS | 8708 | 24.38% |
| Swisscom | 3303 | 30.53% |
| Telefonica del Peru | 6147 | 7.30% |
| SoftBank BB | 17676 | 3.23% |
| Chubu Telecommunications | 18126 | 26.63% |
| T-Mobile USA | 21928 | 27.37% |
| Belgacom | 5432 | 10.18% |
| Hughes Network Systems | 6621 | 27.41% |
| VOO | 12392 | 31.55% |
| StarHub | 4657, 55430 | 18.43% |
| Opera Software ASA | 39832 | 11.52% |
| XS4ALL | 3265 | 22.95% |
| Forthnet | 1241 | 4.76% |
| Google Fiber | 16591 | 76.13% |
| Janet | 786 | 4.61% |
| China Telecom | 4134, 4809 | 0.24% |

Showing 1 to 25 of 207 entries

First Previous 1 2 3 4 5 Next Last

# Scale: Internet of things needs IPv6
# Scale:  Cloud needs IPv6



# Lesson Learned

1. IPv6-only works
2. IETF works for operators (making sausage is not pretty, but it works)
3. Breaking stuff is not ok
4. Question the answers (if dual-stack, then what?)
5. Don’t boil the ocean
    1. Tether is still IPv4-only
    2. Old phone are still IPv4

# Summary

- IPv4 does not fit the business needs to grow the edge of our networks fueled by growth from internet of things and cloud
- IPv6 works today and is deployed on some of the largest edge networks, backbones, and content clouds
- 464XLAT allows networks to grow without many public IPv4 addresses
- With IPv6-only networks being deployed, IPv4 is now a legacy liability …going the way of Windows XP

**Big Picture:** We must avoid the Internet's largest growth engines (mobile, cloud, "things") from being indefinitely tied to scarce IPv4 and fragile stateful NAT44.