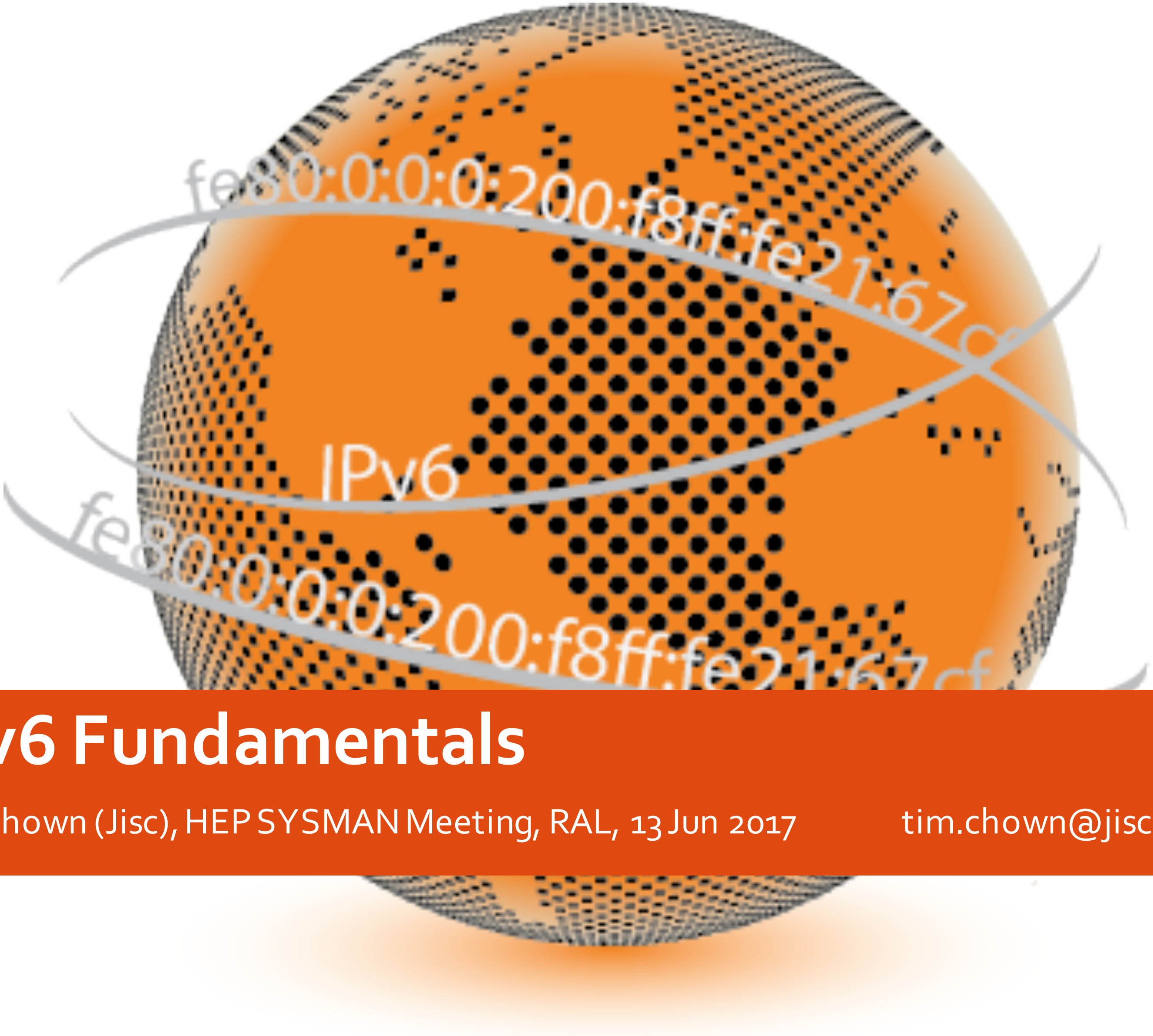


# IPv6 Fundamentals

Tim Chown (Jisc), HEP SYSMAN Meeting, RAL, 13 Jun 2017 tim.chown@jisc.ac.uk

# IPv4 status (exhaustion!)

# 40 years of IP networking

- IPv4 has lasted some 40 years or so

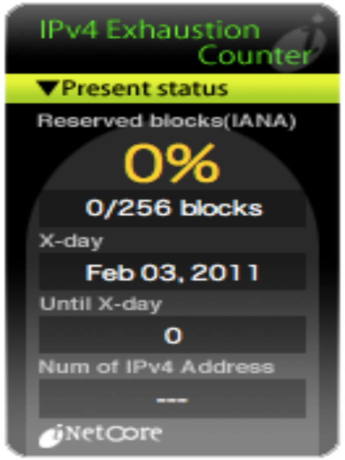
- In the 1970's IP networking was just an 'experiment'
  - A handful of computers taking part
- The designers chose to use 32 bits for IPv4 addresses
  - 4.3 billion (2^32) addresses is enough for an experiment, right?
  - A quite reasonable decision at the time by Vint Cerf and Bob Khan.

- IPv4 address notation was agreed
  - Addresses written in the 'dotted quad' form, e.g., 192.0.2.1

- Today, with IoT and other IP growth areas, 32 bits is clearly not enough

# IPv4 address space status – exhausted!

- The IANA IPv4 global address pool was exhausted in Feb 2011
  - There is no new, unused IPv4 address space left to give to RIRs



IPv4 Exhaustion Counter

▼Present status

Reserved blocks(IANA)

0%

0/256 blocks

X-day

Feb 03, 2011

Until X-day

0

Num of IPv4 Address

---

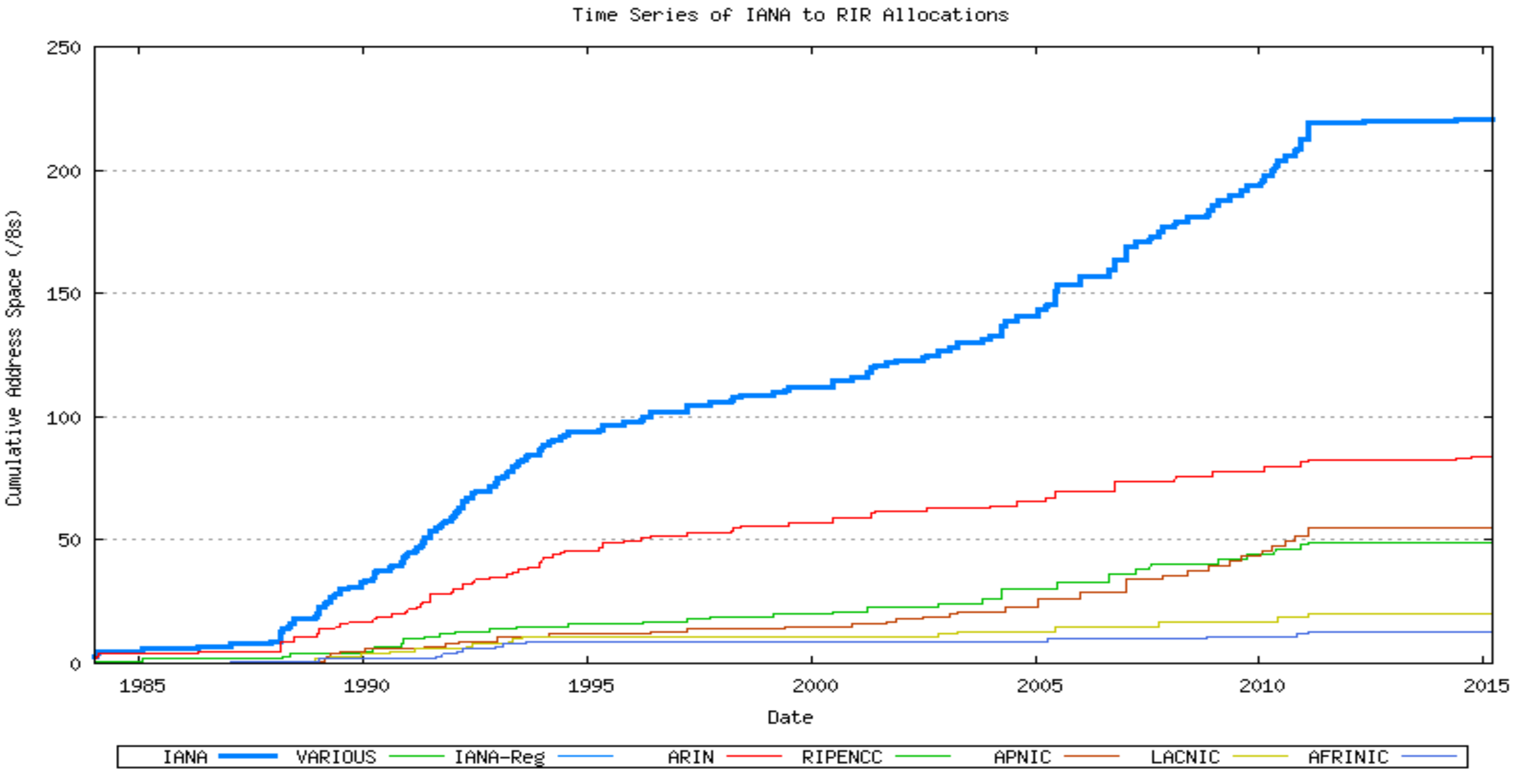
NetCore

- The Regional Internet Registries (RIRs) have varying levels of reserves of address space
  - APNIC and RIPE NCC are on their last /8, and rationing heavily
  - ARIN ran out completely in September 2015

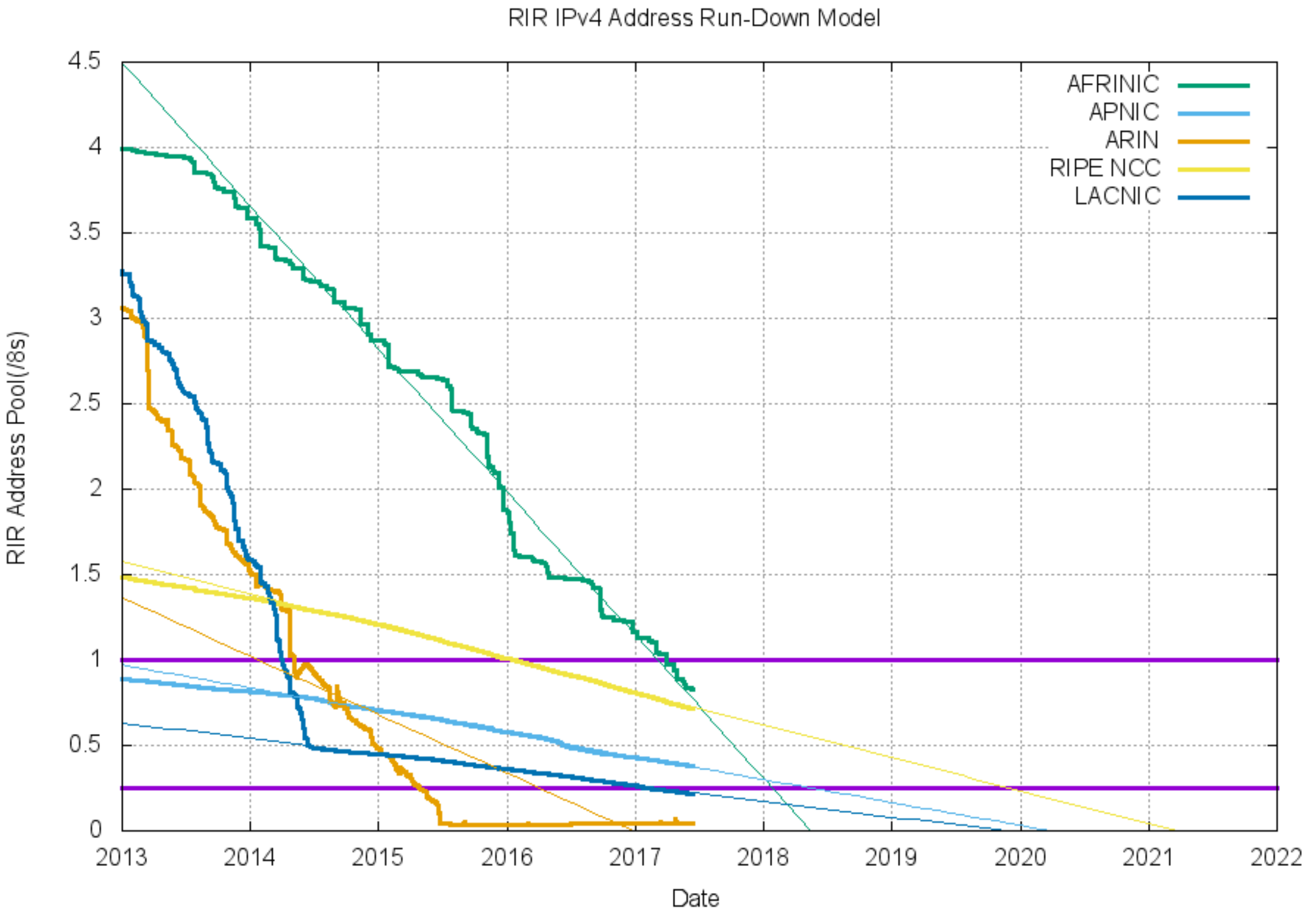
- RIPE NCC is using a 'Last /8' policy: this means the max IPv4 allocation is a /22 (1,024 addresses)
  - So ISPs (including Jisc) have no new supply of significant address space from their RIR
  - Jisc can no longer give (say) a /20 to a new university
  - But existing IPv4 deployments still work, of course; the sky has not fallen (yet)

- See http://www.potaroo.net and http://ipv4.potaroo.net for many, many charts
  - Excellent resource maintained by Geoff Huston

# IPv4 Allocations over time by IANA



# IPv4 run-out projection



# Address exhaustion

- What are the impacts of IPv4 address exhaustion?

# Impact of IPv4 address exhaustion?

- Includes:
  - Some organisations may possibly be “encouraged” to return addresses
  - Increased address space trading/leasing – market at $10/IP or more
  - Increased use of NAT
    - True end-to-end networking difficult, or impossible
    - Increased complexity in network management
    - Accountability issues, potential for overlapping private address space
  - Introduction of Carrier Grade NAT by ISPs
    - Home DSL router has an ISP-private IP on its ‘public’ interface
    - Can be recognised by use of reserved 100.64.0.0/10 prefix (RFC 6598)
    - Likely to have a negative impact on applications, esp. inbound
  - Use of other forms of address sharing
    - Customers might get an address and a range of port numbers to use

# IPv6 Protocols

# IPv6 features?

- The solution to IPv4 exhaustion is IPv6
- NAT has bought us some time, and is now widely deployed in most home networks, and many SME and enterprise/campus networks
- NAT has an adverse effect on network operations, especially end-to-end

- But what IPv6 features or benefits are you aware of already?
  - Thoughts?

# IPv6 features

- Key new features of IPv6
  - **128-bit address space**
  - Host autoconfiguration through "Stateless Address Autoconfiguration" (SLAAC)
  - SLAAC allows devices to generate their own IP address without a DHCP server
- Implicit features
  - LOTS of addresses – so no **need** to use host-based NAT
- Over-hyped (and not really true…)
  - Improved QoS
  - Improved security

# Aside: about IP protocols and standards

» The Internet works thanks to use of commonly agreed protocols

» The Internet Engineering Task Force (IETF) develops IP-related protocols (amongst other protocols higher up the stack)

- › Meets three times annually, and uses mail lists
- › Operates by consensus in Working Groups
- › Any individual can write an Internet draft
- › Internet draft documents discussed and progressed if supported through WG adoption to RFC status if published
- › Now over 8,000 RFCs published

» See http://www.ietf.org

# The IPv6 core protocol

- » IETF work began in the mid-1990's on the protocol that became IPv6

- » Led to publication of RFC 2460 in 1998
  - › This core specification has remained largely unchanged for nearly 20 years
    - – Except for some security-specific updates
    - – Undergoing a (minor) revision this year, largely to include pointers to more recent and relevant RFCs (see draft-ietf-6man-rfc2460bis-13)
  - › Defines header format, including 128-bit addresses, and packet processing

- » The IPv6 address format is defined in RFC 4291
  - › Describes what the addresses look like
  - › (This RFC has been updated once, and is getting a refresh alongside RFC 2460-bis)

# The IPv6 address format

- An IPv6 address is 128 bits
  - But how do we write an IPv6 address?
  - Using dotted decimals like IPv4 would be very long!

- It was agreed that addresses are written as eight sets of four hexadecimal characters, e.g.
  - 2001:0db8:0000:0000:baad:cafe:1234:5678

- To abbreviate, you can omit any leading zeros
  - 2001:db8:0:0:baad:cafe:1234:5678

- And you can replace **one** series of :0: fields with ::
  - 2001:db8::baad:cafe:1234:5678
  - … why only one?

Jisc

# Example 1…

- How can you abbreviate the following IPv6 address?
  - 2001:0db8:0000:0000:0000:0000:0000:0c50
  - A.        2001:0db8:0:0:0:0:0:0c50
  - B.        2001:0db8::0c50
  - C.        2001:db8::c50
  - D.        2001:db8::c5
  -

(taken from http://www.ripe.net/lir-services/training/material)

# Example 2…

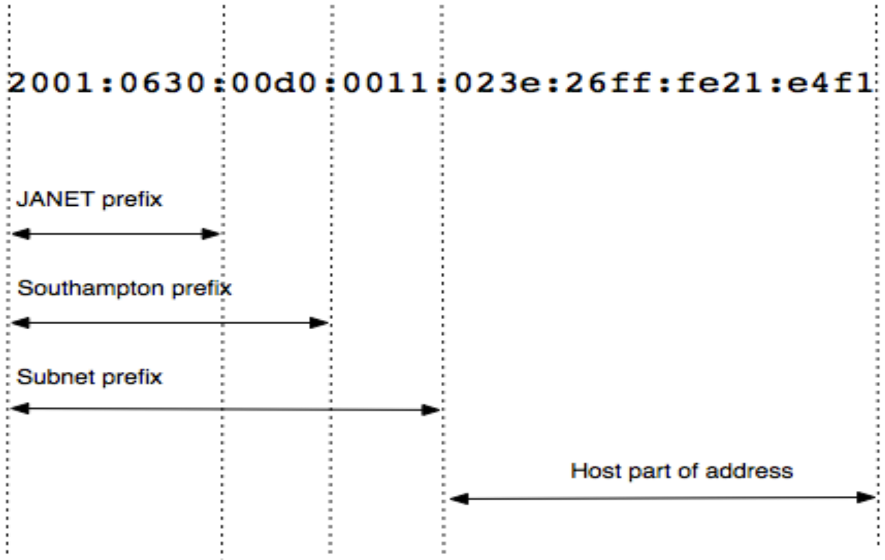
- How can you abbreviate the following IPv6 address?
  - 2001:0db8:0000:0000:b450:0000:0000:00b4

  - A. 2001:db8::b450::b4
  - B. 2001:db8::b450:0:0:b4
  - C. 2001:db8::b45:0000:0000:b4
  - D. 2001:db8:0:0:b450::b4

Note: 2001:db8::/32 is the reserved IPv6 documentation prefix

# Breaking an address down

- IPv6 allocations are made through the RIRs just as they are (or were!) for IPv4
  - The default IPv6 allocation to an ISP is a /32 IPv6 prefix, e.g. Janet has 2001:630::/32
  - A larger ISP, such as Sky UK, can obtain a larger block of address space
- A typical prefix breakdown for a university site might be:



# Site IPv6 prefixes

- As stated, the default allocation for a site, such as a campus, is a /48 IPv6 prefix
  - In practice, a home network may get less, e.g., a /56
- Such prefixes are Provider Assigned/Aggregated (PA), from the ISP
  - This means if a customer changes ISP, they will be given a new, different prefix
  - Which means the customer will have to renumber

- RIRs also offer Provider Independent (PI) allocations
  - These are /48 in size
  - See https://www.ripe.net/publications/docs/ripe-684, which describes the policy
  - Good for customers, but will cause larger global IPv6 routing tables

- Or you can apply to become a Local Internet Registry (LIR), and receive a /32
  - QMUL and UCL have done this

# IPv6 Address Types

- There are two general classes of IPv6 addresses

- Unicast
  - Same as IPv4, but with the addition of link-local addresses
  - More on those in a moment…

- Multicast
  - Inherent to the IPv6 protocols, in particular Neighbour Discovery (ND) (RFC 4861)
  - All multicast addresses fall under ff00::/8
  - IPv6 does not have an IP subnet broadcast addresses
  - It uses link-local multicast within subnets instead
  - So beware any (very) old hub/switch devices

# IPv6 unicast address scopes

- Global addresses
  - Unique globally, routed globally. Just like IPv4.
- Unique Local Addresses (ULAs) (RFC 4193)
  - Used within a site, not routed externally
  - Uses reserved prefix under fc00::/7
  - The other prefix bits are random to make a /48 prefix that is probabilistically unique for the site
  - A **bit** like IPv4 RFC1918 private addresses, but not designed to be used with NAT
- Link-local addresses
  - Unique on a subnet, not forwarded by routers
  - Uses reserved prefix under fe80::/10
  - A **bit** like IPv4 169.254.0.0/16 space (RFC 3927)

# Multiple IPv6 addresses per host

- **So in IPv6, hosts are usually multi-addressed**
  - Invariably with at least a link-local IPv6 and a global IPv6 address
- ULAs may be used **as well as** global addresses
  - Offers stable internal addressing for a site if your global prefix changes
  - Devices inside a routed site can prefer to use their ULA addresses
  - Again, they are **NOT** designed to be used for IPv6 NAT
- Currently it **seems** that no universities are using ULAs
  - Existing IPv6 deployments use global addresses provided by Jisc
- ULAs have been proposed for some uses, e.g. in IPv6 homenets
  - The ISP prefix is more likely to change in those scenarios, so internal address stability is desirable

# Picking addresses: IPv6 Address Selection

- Defined in RFC 6724
  - Updates original guidance in RFC 3484
- Used to allow a host to – for example – pick an appropriate source address to use with a given destination address
  - Match scopes where possible
  - e.g. it should use a ULA source to talk to a ULA destination
  - Do not use a link-local source to talk to a global destination (why?)
- However, the multiple address issue is also a challenge for network management and monitoring
  - Tracking which addresses belong to which devices
  - Eric will likely mention this later; if not, ask him ☺

# The loopback address…

Zazzle

# IPv6 Neighbour Discovery protocol

- IPv6 employs a suite of protocols known as ‘Neighbour Discovery’, which take the form of either link-local unicast or link-local multicast ICMPv6 messages

- Nodes can send Router Advertisement (RA) messages, to let hosts know about various properties of the link they serve

- Nodes can send Router Solicitations to request any routers to send an RA

- Neighbour Solicitation (NS) and Neighbour Advertisement (NA) messages provide the equivalent of the IPv4 ARP function (i.e., IP to MAC address lookup)

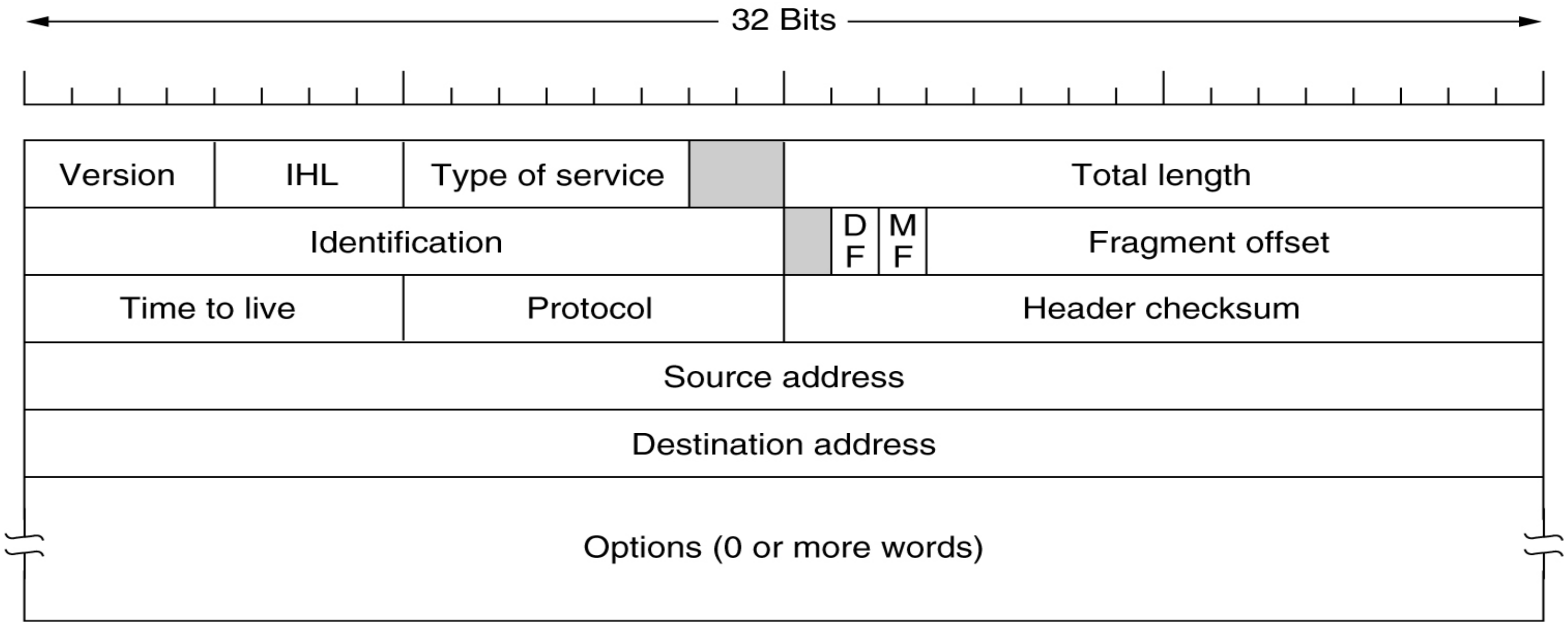
- Router Redirects can be sent to a host to tell it about a better first-hop router to get to a destination

# IPv6 packet headers

- IPv6 by design streamlines the main IP header
  - So the IPv6 header has less fields than the IPv4 header
  - The header is also now a fixed size
  - The header is still longer though, due to the 128-bit addresses

- For additional functions/options, IPv6 uses optional Extension Headers, inserted by the sender between the main header and the payload
  - Used when needed, e.g. for fragmentation, or IPsec
  - So you will see a ‘chain’ of main header, optional headers, then the payload
  - In most cases, you just see the main header and payload

# IPv4 packet header format



32 Bits

Version | IHL | Type of service | Total length

Identification | DF | MF | Fragment offset

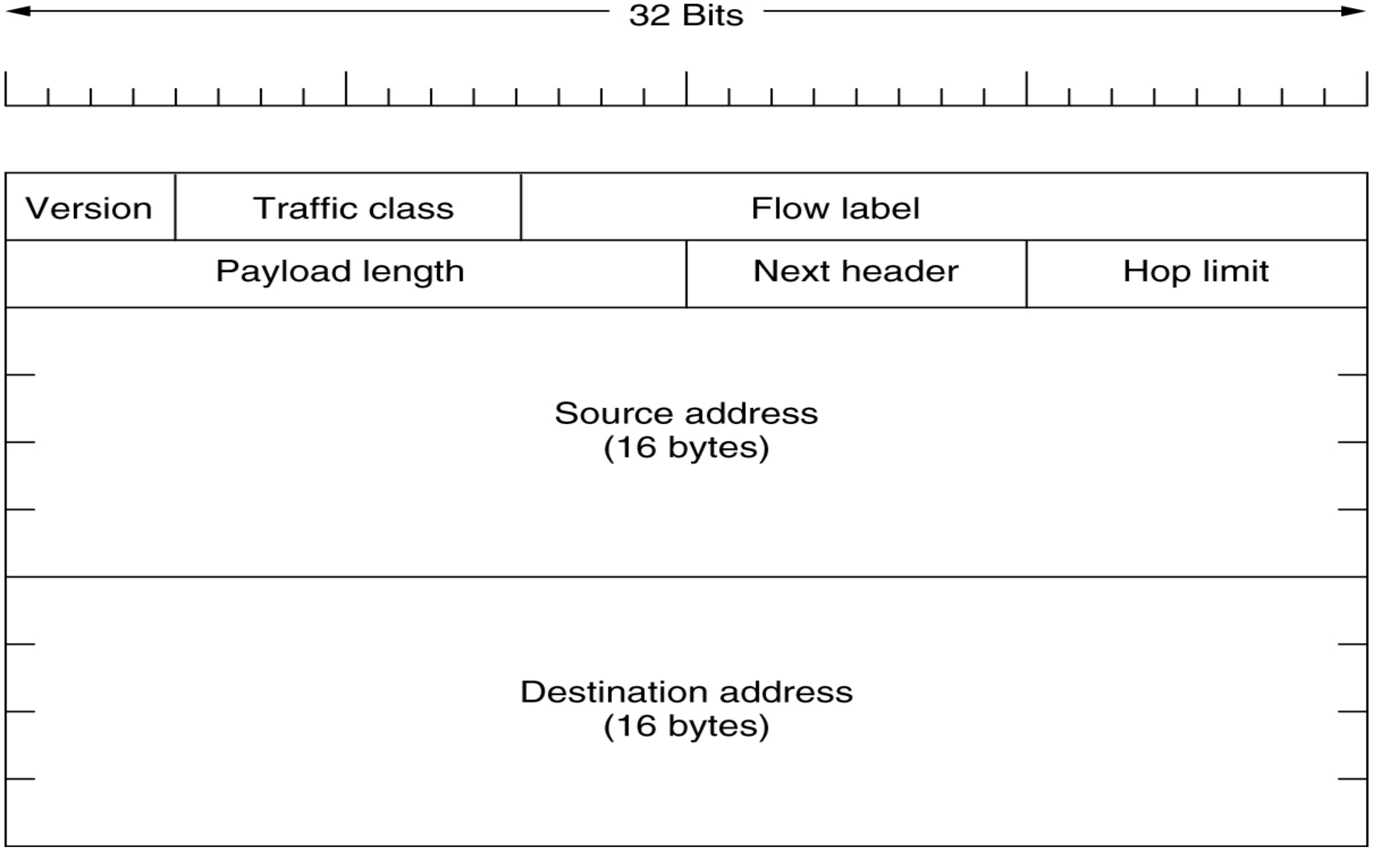
Time to live | Protocol | Header checksum

Source address

Destination address

Options (0 or more words)

# IPv6 packet header format



# IPv4 compatibility?

- IPv6 is a new IP protocol, with 128 bit addresses
- The packet headers are clearly different

- IPv6 is thus not directly compatible with IPv4
- An IPv4-only device therefore cannot send an IPv4 packet directly to an IPv6 device

- We'll return to how this issue is handled later on when we look at IPv6 transition / integration with IPv4

- Devices can however run both IPv4 and IPv6 together (known as dual-stack), and then choose which protocol to use

# More on IPv6 Extension Headers

- Indicated by Next Header field, e.g.:
  - Hop-by-hop header
  - Destination options header
  - Routing header
  - Fragmentation header
  - Authentication and ESP headers

- In principle, new Extension Headers can be defined
- In practice, firewall implementations can make this problematic, as they drop unknown header types, which new headers will be
- Experiments have shown that packets with certain IPv6 EHs may be dropped by various devices in access networks, or by site firewalls (see RFC 7872)
  - Not a problem for 'normal' traffic

# Handling fragmentation in IPv6

- Fragmentation in IPv6 is only performed by the end hosts
  - Uses the optional IPv6 fragmentation header
  - Fragmentation is **not** performed by routers in the network

- Thus hosts must be able to establish the path MTU
  - Implies the ICMPv6 messages used for PMTU discovery must not be filtered
  - See RFC 4890 for ICMPv6 filtering recommendations
  - Don’t just blindly drop all ICMPv6 traffic at your site border!

- IPv6 links must have an MTU of at least 1280 bytes
  - For Ethernet, the MTU will usually be 1500 bytes
  - You may have scenarios where you want to exploit a larger MTU, e.g. 8192 or 9000

# Address allocation and management

- We've seen what IPv6 packets look like, and the format of an IPv6 address

- We now need methods to
  - Get allocations of IPv6 address blocks to use within our site
    - As seen earlier, the Janet Service Desk allocates /48s to Janet-connected sites
  - Decide the method of configuring addresses on hosts in our site

# Configuring IPv6 hosts

- Given an allocation, there are two choices for configuring hosts
  - (in addition to manual configuration, if you prefer that)

- DHCPv6
  - **Largely** similar to DHCPv4
  - Familiar model, arguably helps to support accountability
  - Stateful – the server holds lease information for each address used by a host

- Stateless Address Autoconfiguration (SLAAC)
  - Defined in RFC 4862. New for IPv6. Allows hosts to essentially pick their own address.
  - Introduces new management challenges
  - Still requires stateless DHCPv6 for additional configuration parameters, e.g. NTP server

# IPv6 Autoconfiguration

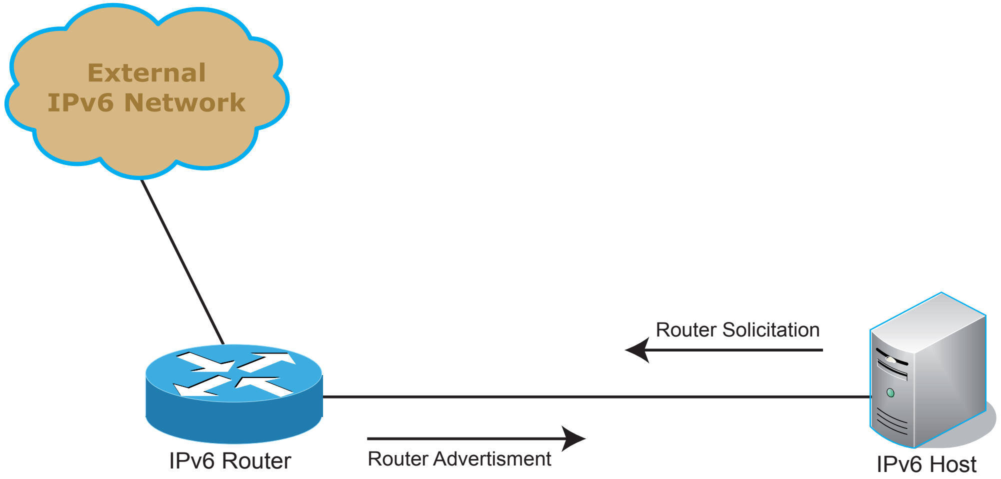
- IPv6 nodes can use SLAAC to determine their
  - IP address
  - Default gateway
  - (Optionally) DNS resolver

- SLAAC works by routers sending link-local multicast Router Advertisement (RAs)

- An RA message contains information that may include:
  - On-link prefix(es), with preferred/valid lifetimes
  - The link Maximum Transmission Unit (MTU) ; typically 1500 for Ethernet
  - An indication of the availability of DHCPv6; M = stateful DHCPv6 available, O = stateless DHCPv6 available
  - A-flag; A = 1 means configure address with SLAAC; A = 0 means do not configure address with SLAAC
  - (Optional) DNS resolver information (RFC 8106)

# RA operation

- The Router Advertisement is multicast on the local subnet
  - Its (link-local) source address implies the default router address
  - SLAAC works by appending a 64-bit Extended Unique Identifier (EUI-64) interface identifier to the 64-bit network prefix to form the host's 128-bit IPv6 address
  - The EUI-64 interface identifier is formed by taking the 48-bit MAC address and inserting 16-bits of padding ('fffe') in the middle, and then toggling the 'universal' bit.



# Autoconfiguration example

- For example:
  - Host MAC address = 08:00:20:9c:14:66
  - Network prefix = 2001:630:80:2::/64
  - Address = 2001:630:80:2:0a00:20ff:fe9c:1466

- Note:
  - The 48-bit MAC address requires the 16-bit 'fffe' padding to build a 64-bit EUI
  - The universal/local bit is inverted (hence 'oa')
  - Key principle is to form the address by using the prefix from the RA appended with the device's MAC address (with the padding) to form the 128-bit IPv6 host address

- SLAAC means all IPv6 hosts use 64-bit links; i.e. every host network is a /64
- See RFC 7421 for more discussion of "Why 64 bits?"

Jisc

# Privacy concerns with SLAAC…?

- A host autoconfiguring in different visited networks could be traced by its fixed 64-bit interface identifier (IID)
  - i.e., the last 64 bits would be the same wherever the device appeared

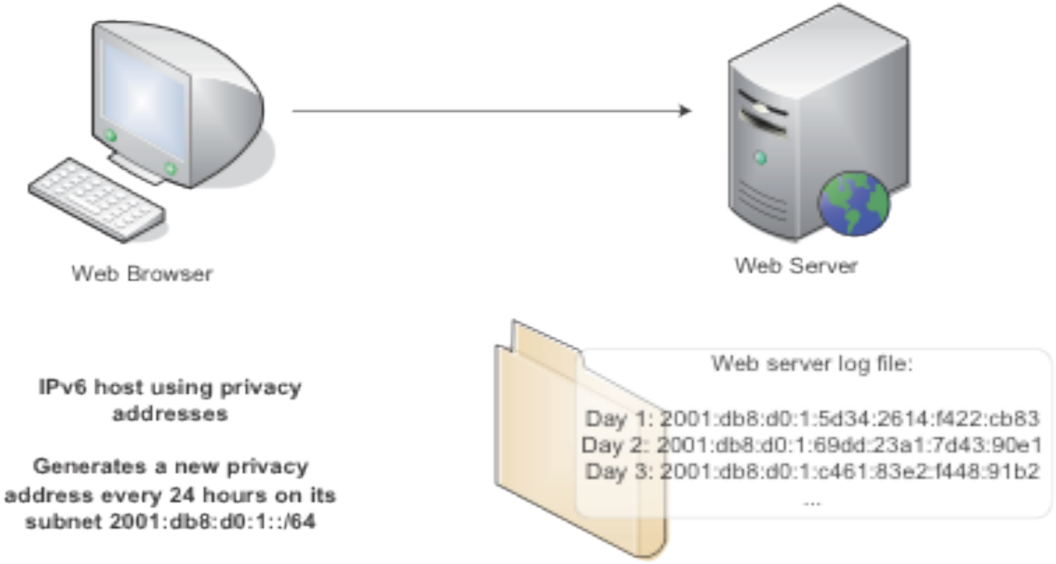
- So the IETF defined IPv6 privacy addressing (RFC 4941)
  - Randomly generate the 64-bit host part when attaching to a network
  - And a host may **also** change its privacy address periodically even if not changing subnets; typically every 24 hours (e.g., as with MS Windows)

- Privacy addresses are good for users, but complicate network management
  - Which addresses belong to which hosts?
  - More multi-addressing!

# Privacy addresses and logging

So, how many hosts do you really have?

RFC 4941 says you can change your privacy address as little as every 10 minutes



**IPv6 host using privacy addresses**

**Generates a new privacy address every 24 hours on its subnet 2001:db8:d0:1::/64**

Web server log file:

```
Day 1: 2001:db8:d0:1:5d34:2614:f422:cb83
Day 2: 2001:db8:d0:1:69dd:23a1:7d43:90e1
Day 3: 2001:db8:d0:1:c461:83e2:f448:91b2
...
```

# Linux configuration example

## Run ifconfig

```
eth0   Link encap:Ethernet  HWaddr 00:30:48:76:53:14
        inet addr:152.78.71.152  Bcast:152.78.71.255  Mask:255.255.255.0
        inet6 addr: 2001:630:d0:f110:230:48ff:fe76:5314/64 Scope:Global
        inet6 addr: fe80::230:48ff:fe76:5314/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:795291388 errors:0 dropped:0 overruns:0 frame:0
        TX packets:710162840 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:3111500779 (2.8 GiB)  TX bytes:1177068949  (1.0 GiB)
```

# Windows configuration example

```
Interface 4: Ethernet: Local Area Connection
 uses Neighbor Discovery
 uses Router Discovery
 link-layer address: 00-00-cb-68-0b-2e
  preferred global 2001:630:d0:112:309e:3ba9:d0df:1afc, life 57m25s/27m25s (temporary)
  deprecated global 2001:630:d0:112:cc4e:835c:7e1b:e482, life 57m25s/0s (temporary)
  deprecated global 2001:630:d0:112:f4c5:398e:b5f3:bf58, life 57m25s/0s (temporary)
  deprecated global 2001:630:d0:112:88bd:46d0:b997:6dc4, life 57m25s/0s (temporary)
  deprecated global 2001:630:d0:112:e07c:fe6b:a58a:1608, life 57m25s/0s (temporary)
  deprecated global 2001:630:d0:112:b4dc:cfc5:c6a7:3724, life 57m25s/0s (temporary)
  deprecated global 2001:630:d0:112:1ca9:c9b:849e:7869, life 57m25s/0s (temporary)
  preferred global 2001:630:d0:112:200:cbff:fe68:b2e, life 57m25s/27m25s (public)
  preferred link-local fe80::200:cbff:fe68:b2e, life infinite
```

Temporary addresses are IPv6 Privacy Addresses

These change over time – default of a new Privacy Address every 24 hours on Windows

A host also has a standard SLAAC-based global IPv6 address, may be DNS-registered if running services

Privacy addresses are only used for initiating connections from a host

# New: stable, per-prefix Interface IDs

- RFC 7217 has recently defined an alternative to MAC-based SLAAC addresses
- Still uses the RA for address generation, but no longer appends an EUI-64
- Instead, RFC 7217 generates Interface IDs that are stable for any given visited subnet (i.e., per /64 network prefix used)
  - Uses a hashing method on the prefix to build the Interface ID
  - So you get the same last 64 bits in your address whenever you attach to a subnet using the same prefix, without exposing your MAC address

- May be used independently of IPv6 Privacy Addressing, i.e. typically you would:
  - Use classic SLAAC, with or without Privacy addresses
  - Or use RFC7217-based SLAAC, with or without Privacy addresses
- Windows 10 seems to be using RFC7217; other OSes likely to follow

# Using DHCP for IPv6

- IPv6 has two variants of DHCPv6

- Full stateful DHCPv6 (RFC 3315; currently undergoing a refresh)
  - Includes IPv6 address lease support, as per DHCP for IPv4, i.e., the DHCPv6 server **maintains state** on the IPv6 addresses leased to hosts
  - Supported in common platforms, including the popular ISC DHCP
  - The only exception, unfortunately, is Android – see https://code.google.com/p/android/issues/detail?id=32621

- Stateless DHCPv6 (RFC 3736)
  - For use with SLAAC
  - Used for additional configuration info only, like NTP server, or search domain

# DHCPv6 considerations

- DHCPv6 uses new DHCP Unique Identifiers (DUIDs)
  - DUIDs aren't known a priori like Ethernet/MAC addresses
  - May be a concern if you want to link IP addresses to MAC addresses by DHCP
  - But there are some large-ish DHCPv6 deployments out there, e.g. at CERN, for whom this was not a concern
- In practice, in an enterprise / campus deployment, clients will speak to a DHCP server via a DHCP relay running on a router
  - Thus the IETF has introduced RFC 6939 to allow MAC addresses to be included as a DHCPv6 option, and forwarded by DHCPv6 relays
  - Support demonstrated in Ubuntu, Cisco IOS and ISC DHCP
  - Other platforms following

# RAs are still required if you use DHCPv6

- Why?

- The RA is the only way a host can learn its default gateway
  - There is no DHCPv6 Default Gateway option
  - DHCPv6 also has no option for on-link prefix(es)

- Therefore all IPv6 networks must use RAs
  - And consider their security implications
  - For example, hosts can send rogue RAs, “accidentally” or maliciously

- Note that rogue RAs can also be an issue on “IPv4 only” networks
  - More on this from Eric later…

# Addressing on point to point links

- » There's been discussion over the prefix length to use for point-to-point
  - › /64, /126 or /127?
- » Some concerns with using /64
  - › Address space 'wasted'
  - › Possible 'ping pong' attacks (packet to an unused address bounces between routers)
  - › Possible ND cache exhaustion attacks
- » IETF now recommends /127 for point-to-point links
  - › See RFC 6164
  - › Can still allocate a /64 if you want to though

# IPv6 routing

- New versions of familiar protocols have been defined
  - Multiprotocol BGP (RFC 2545)
  - IS-IS (RFC 5308)
  - OSPFv3 (RFC 2740)
  - RIPng (RFC 2080)

- Most campuses/enterprises probably run OSPFv2 or IS-IS
  - Can run OSPFv2 alongside OSPFv3
  - Note: Various platforms don't support multi-AF OSPFv3 yet, so using OSPFv3 for both protocols perhaps premature
  - Do **request feature parity in procurements** though!
  - Opportunity to migrate to IS-IS if not using it

# Activity: A quick live demonstration

## A glance at IPv6 in action

# A quick IPv6 hands-on

- A quick example…
- Using a remote ssh login at Southampton

# Differences to IPv4 – a summary

# IPv6 key differences

| | IPv4 | IPv6 |
|---|---|---|
| Address length | 32 bits | 128 bits |
| Prefix length | Varies, typically /24 | Always /64 in host subnets |
| Address configuration | DHCPv4 | Stateless Autoconfiguration<br>DHCPv6 |
| Addresses used | Private **or** Global | Link-local **and** Global |
| Address resolution | ARP | Neighbour Solicitation / Advertisement |
| Host Path MTU Discovery | Optional | Required |
| Fragmentation | By hosts or routers | Only by hosts |
| Private addressing | RFC 1918 | Unique Local Addresses (ULAs)<br>(not designed for use with NAT) |

# IPv4/IPv6 integration

# Scenarios

- There are many scenarios where IPv4/IPv6 integration tools and solutions are required, e.g.:
  - A user on a dual-stack host (laptop) on an IPv4-only ISP (e.g. a wireless hotspot) wants to access remote IPv6 services
  - Connecting IPv6 networks which only have IPv4 connectivity between them
  - An IPv6-only system needs to talk to a 'legacy' IPv4-only system
    - A realistic scenario on newly deployed access networks
    - A common scenario on mobile phone networks

# General approaches

- » Tunnels / encapsulation
  - › Sending IPv6 packets over IPv4-only infrastructure
- » Translation
  - › Used between IPv6-only and IPv4-only nodes
  - › Can be done at application, transport or IP layer
  - › (NAT64/DNS64/464XLAT not covered today – but can discuss if wanted…)
- » Dual-stack
  - › Instead, choose to run both protocols
  - › Can talk IPv4 to IPv4-only networks and IPv6 to IPv6-only ones
  - › Application chooses which to use, e.g., based on sorting DNS query responses

# IPv6 tunnels over IPv4

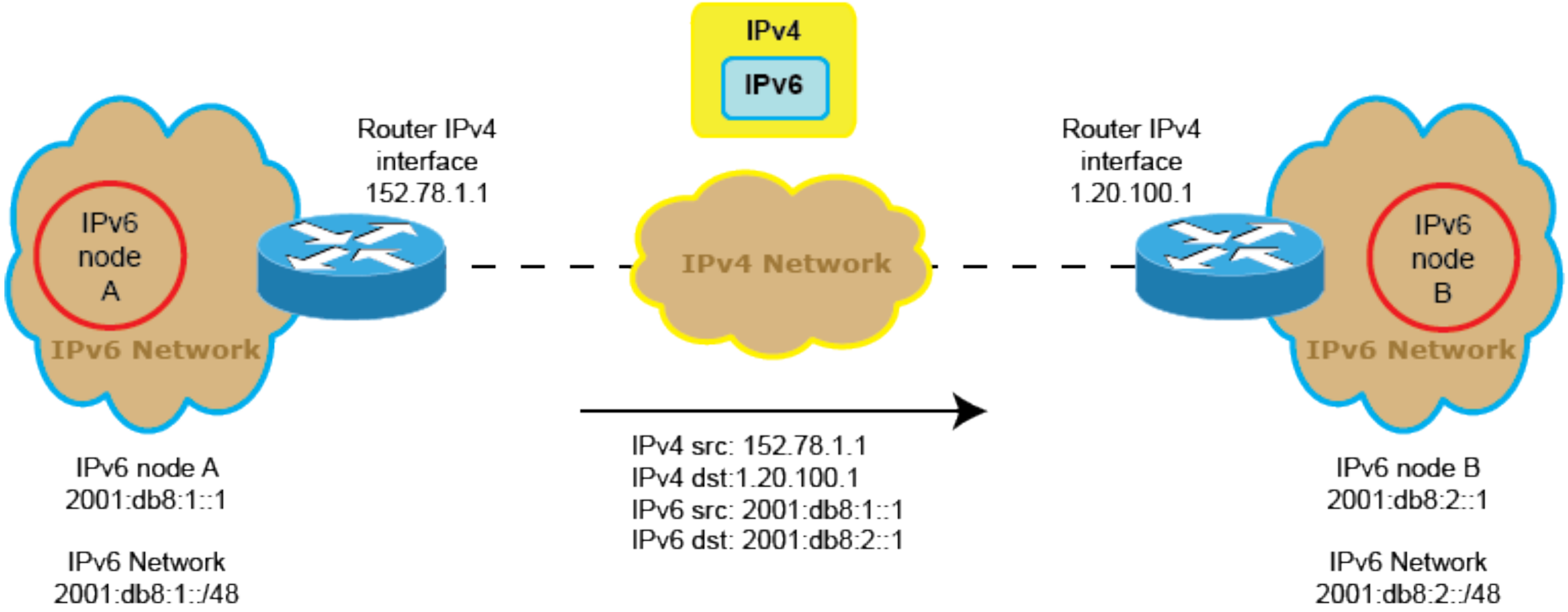
- » IPv6 packets encapsulated as the payload of IPv4 packets

- » Typical usage
  - › Connect a user with a dual-stack device on an IPv4-only ISP to IPv6 services (still common)
  - › Connect IPv6 networks over an IPv4 path (increasingly rare)

- » Thus tunnels can be
  - › Host-to-router
  - › Router-to-router

- » May be set up manually or automatically

# Tunnel addressing example


IPv4
IPv6
Router IPv4 interface 152.78.1.1
Router IPv4 interface 1.20.100.1
IPv6 node A
IPv6 Network
IPv4 Network
IPv6 node B
IPv6 Network
IPv6 node A 2001:db8:1::1
IPv6 Network 2001:db8:1::/48
IPv4 src: 152.78.1.1
IPv4 dst:1.20.100.1
IPv6 src: 2001:db8:1::1
IPv6 dst: 2001:db8:2::1
IPv6 node B 2001:db8:2::1
IPv6 Network 2001:db8:2::/48

Jisc

# Manually configured tunnels

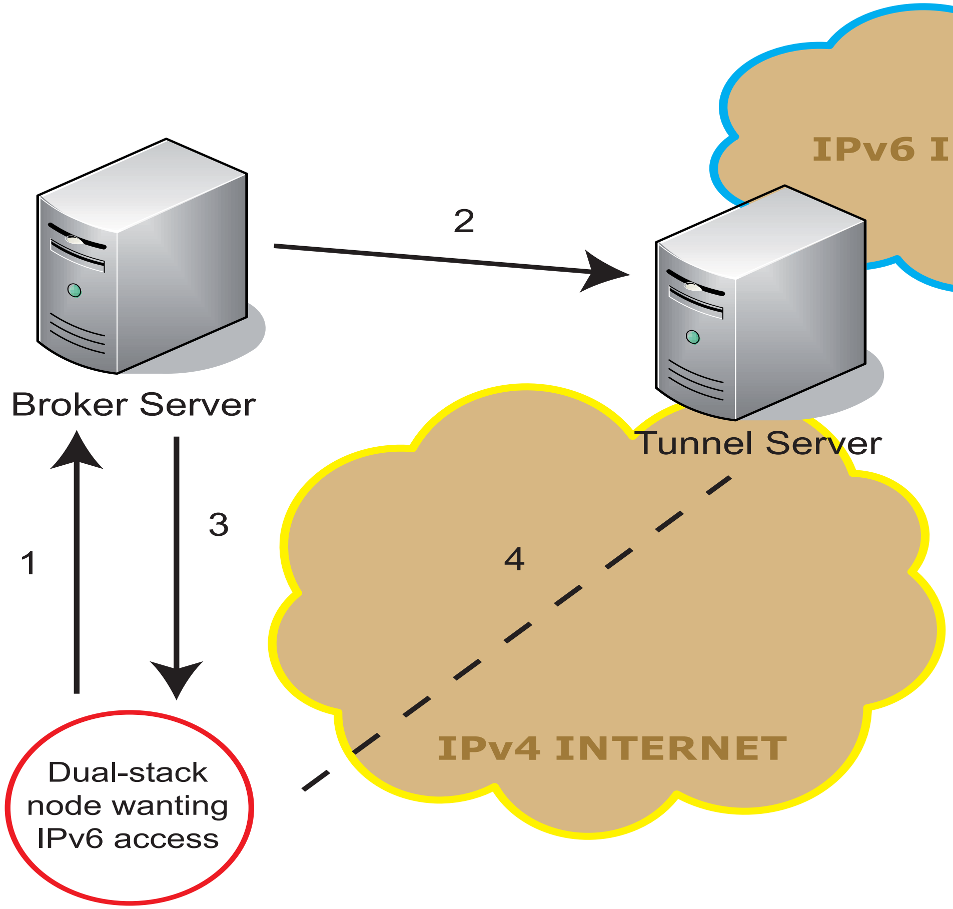
- Easy to set up and configure
- Good management potential
  - An ISP configures the tunnels, so controls deployment, and is fully aware of customer demand
  - Used historically by on Janet to connect sites running pilots; tunnel from dual-stack site router to Janet tunnel server
  - Jisc now prefers you use the native IPv6 they deliver to your door
  - I'd assume most GridPP sites have native IPv6 connectivity to Janet
- Your users may be interested in IPv6 access from home or other IPv4-only networks
  - This is a scenario served well by tunnel brokers

# Tunnel brokers

- Tunnel brokers have proven quite popular over time
  - In Europe the best example is www.tunnelbroker.net
  - Reportedly well over 100,000 users at its peak
  - Good way to get IPv6 experience at home, if your ISP lacks IPv6 support
  - Not for use to connect campus sites ☺

- General mode of operation:
  - User/client registers with broker service, e.g. via a web page
  - Tunnel requested by user from their IPv4 address
  - Broker tunnel server sets up its end of the tunnel
  - User/client configures client end of tunnel, e.g. by executing a script

# Tunnel broker architecture



1: Client requests tunnel and (in some cases) authenticates

2: Broker sends remote endpoint confirguration to tunnel server

3: Broker returns configuration to client system to create tunnel

4: Tunnel established

# Dual-stack IPv4/IPv6

» Already hinted at

- › Run both protocols on hosts and routers
  - – IPv6 support is now strong on all mainstream platforms
- › Let applications/services decide which to use
- › Aim to allow IPv4-only or IPv6-only nodes to function fully
- › A stepping stone to IPv6-only operation

» Implies

- › All network/host/application elements support IPv6
- › IPv6-capable security components are available
- › IPv4 must not be adversely affected – requires IPv6 functions to be implemented in hardware as per IPv4
- › But this is all very possible today

# Choosing the protocol

- In a dual-stack network the choice of protocol used is application specific
  - Applications use DNS to resolve names to IP addresses
  - DNS may return IPv4 (A) and/or IPv6 (AAAA) responses
  - Application may sort these as it wishes, and favour IPv4 or IPv6
- If IPv6 is preferred, you must be confident about the performance / robustness of your IPv6 connectivity
  - Users will notice connection issues, evidenced as timeouts before falling back to use IPv4
  - This was a concern a few years ago, but not so valid today – witness all DNS root servers are now available via IPv6
  - Browsers typically implement "Happy Eyeballs" (RFC 6555) to mitigate poor performance

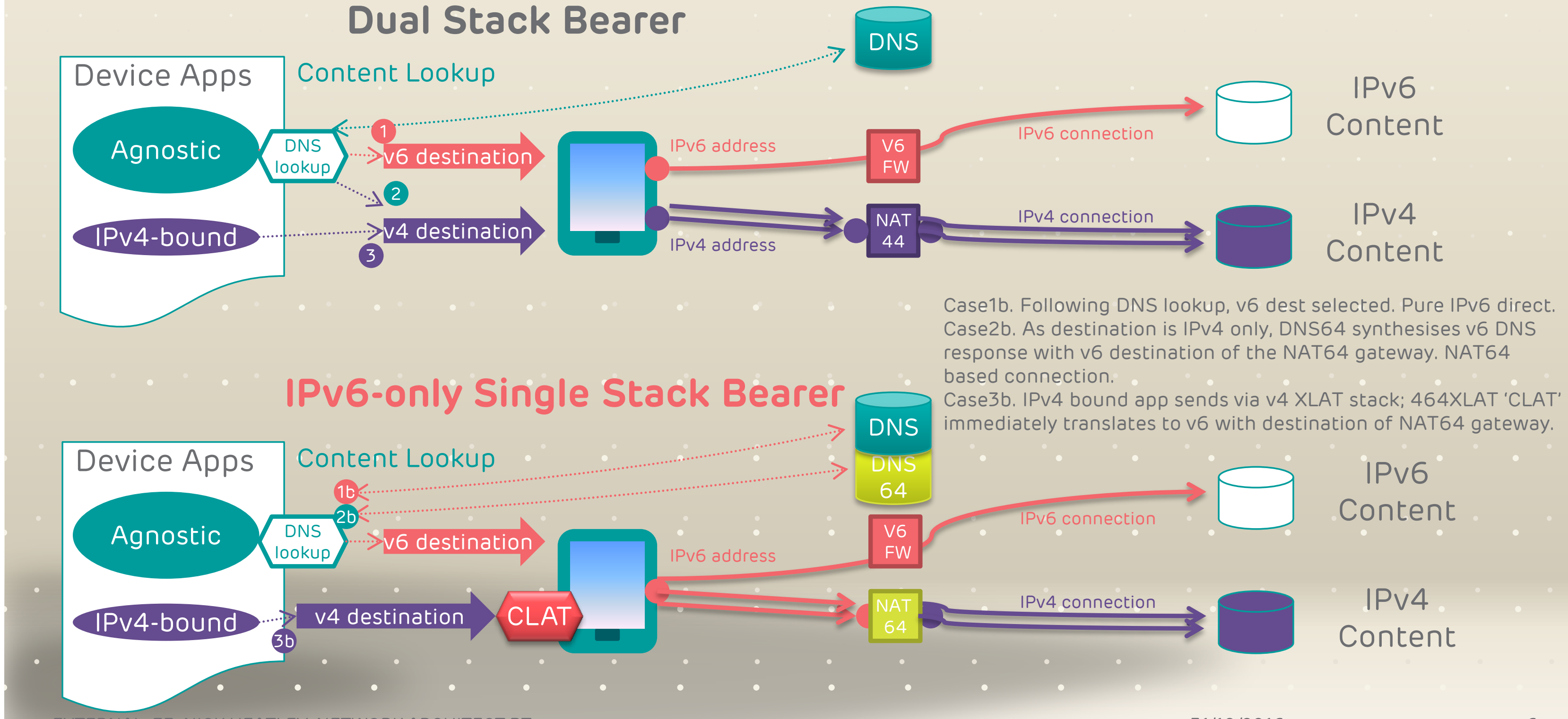
# Aside: DNS root servers

```
a.root-servers.net. IN     A             198.41.0.4
a.root-servers.net. IN     AAAA          2001:503:ba3e::2:30
b.root-servers.net. IN     A             192.228.79.201
b.root-servers.net. IN     AAAA          2001:500:84::b
c.root-servers.net. IN     A             192.33.4.12
c.root-servers.net. IN     AAAA          2001:500:2::c
d.root-servers.net. IN     A             199.7.91.13
d.root-servers.net. IN     AAAA          2001:500:2d::d
e.root-servers.net. IN     A             192.203.230.10
e.root-servers.net. IN     AAAA          2001:500:a8::e
f.root-servers.net. IN     A             192.5.5.241
f.root-servers.net. IN     AAAA          2001:500:2f::f
g.root-servers.net. IN     A             192.112.36.4
g.root-servers.net. IN     AAAA          2001:500:12::d0d
h.root-servers.net. IN     A             128.63.2.53
h.root-servers.net. IN     AAAA          2001:500:1::803f:235
i.root-servers.net. IN     A             192.36.148.17
i.root-servers.net. IN     AAAA          2001:7fe::53
j.root-servers.net. IN     A             192.58.128.30
j.root-servers.net. IN     AAAA          2001:503:c27::2:30
k.root-servers.net. IN     A             193.0.14.129
k.root-servers.net. IN     AAAA          2001:7fd::1
l.root-servers.net. IN     A             199.7.83.42
l.root-servers.net. IN     AAAA          2001:500:3::42
m.root-servers.net. IN     A             202.12.27.33
m.root-servers.net. IN     AAAA          2001:dc3::35
```

# Translation approaches

- Required if you want to access IPv4-only content from an IPv6-only device
  - e.g., in an IPv6-only access network
- Solutions are NAT64 / DNS64 / 464XLAT
  - Where DNS is used, a DNS64 resolver 'tricks' a client into believing it's sending to an IPv6 destination, by translating the IPv4 destination into an IPv6 address
  - Without DNS (IPv4 literals), clients can do translation through 464XLAT
- These are widely used by mobile operators
  - i.e., people selling real services that depend on it
- For EE example in UK, see a UKNOF talk by Nick Heatley:
  - https://www.youtube.com/watch?v=lKyuQ8mb_GE
  - https://indico.uknof.org.uk/event/38/contribution/8/material/slides/1.pdf

# ASSURING CONNECTION TO APPS & SERVICES:
# IPV6-ONLY WITH 464XLAT (RFC6877)



Case1b. Following DNS lookup, v6 dest selected. Pure IPv6 direct.
Case2b. As destination is IPv4 only, DNS64 synthesises v6 DNS response with v6 destination of the NAT64 gateway. NAT64 based connection.
Case3b. IPv4 bound app sends via v4 XLAT stack; 464XLAT 'CLAT' immediately translates to v6 with destination of NAT64 gateway.

EXTERNAL. EE. NICK HEATLEY, NETWORK ARCHITECT BT 31/10/2016 6

# IPv6 deployment in practice

# First - reasons to deploy IPv6?

- What are the drivers for your university to deploy IPv6?
- IPv4 address space is under pressure, but established universities and research organisations quite commonly have an old Class B /16 IPv4 address block
- They may, or may not, be running short of address space, e.g. for eduroam

- Why else might a university/college, or any other organisation, deploy IPv6?

- (In the GridPP case, your community has decided IPv6 is important, but to use it you'll presumably need support from your university or organisation to deploy it)

- Thoughts?

# Some reasons to deploy

- **To support teaching and research**
- Turn on IPv6 on public-facing servers to **simplify the ability of emerging IPv6-only access networks to communicate with you**
  - And thus avoid translation in the network (NAT64, etc)
- To **manage IPv6 as a security measure**
  - All common IP devices have IPv6 support, and usually on by default
- **Gain experience** in IPv6, to understand how to specify procurement requirements
  - Even if you don't plan to turn it on just yet
  - See http://www.ripe.net/ripe/docs/ripe-554 for example
- To allow deployment of new IPv6 applications
  - e.g. true peer-to-peer applications with IPsec (a la Xbox); **innovation at the edge**
- To improve staff / student experience
  - Bearing in mind that **your users will now increasingly have IPv6 at home**

# Janet and IPv6

- Janet has been running IPv6 dual-stack since around 2003
  - Undertook first IPv6 tests in 1996/97 on the then ‘6bone’ network

- Janet has a /32 from the RIPE NCC
  - 2001:630::/32
  - Allocates /48’s from this prefix to organisations, by default
  - Two sites have a /44 – Oxford and Cambridge, presumably due to their colleges

- Various Jisc/Janet services are IPv6-enabled
  - Jisc web site – www.jisc.ac.uk - via Cloudflare
  - The .ac.uk DNS service
  - Janet NTP servers
  - eduroam RADIUS peerings
  - …

# High-level deployment steps?

- Preparation:
  - Arranging IPv6 connectivity (to Janet)
  - Getting IPv6 address space, and forming an IPv6 address plan
  - Deciding the scope of your deployment project
    - Don’t need to do the whole site from day 1
  - Audit systems and software for IPv6 capability; what s/w might need porting?

- Deployment:
  - Enabling IPv6 on the wire; routing IPv6 on the core (not exposing IPv6 to clients initially)
  - Ensuring security policy is applied; firewalls, IDS
  - Ensuring network management and monitoring is operational
  - Configuring supporting services, including DNS
  - Finally, enabling RAs on LANs where IPv6 is required; add DNS entries

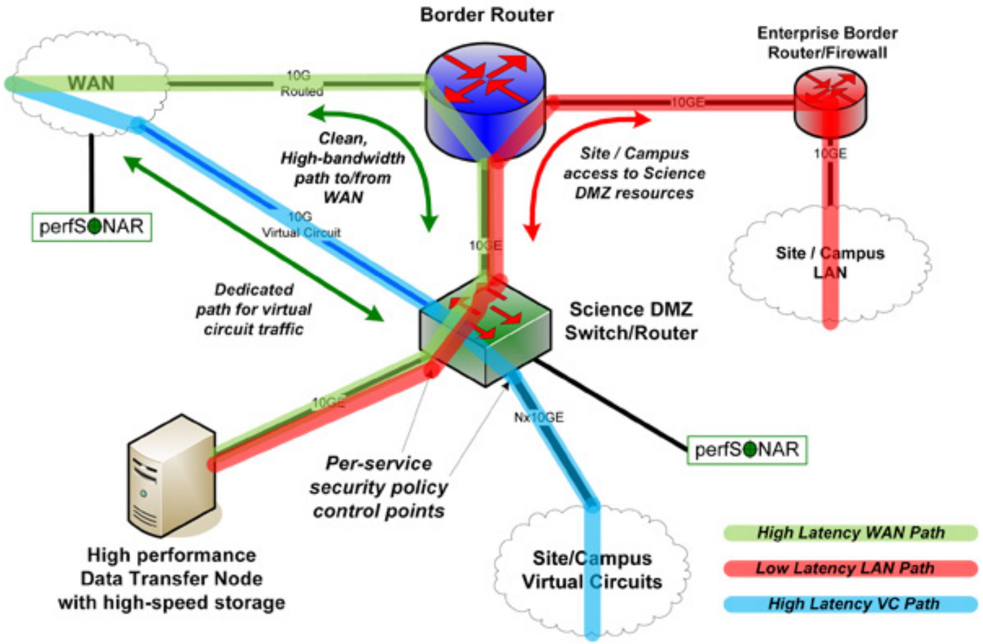
# Production deployment – projects?

- No need for a full IPv6 roll-out from day one
  - Not practical, and not required
  - UK universities to date are all deploying dual-stack; IPv6-only is a future aspiration

- Options that a number of universities have used include
  - Public-facing services (e.g., web presence, DNS servers); see RFC 6883 for advice
  - Wireless network (e.g., eduroam)
  - Computer Science / research department(s)
  - Computing Service department
  - 'Science DMZ' is another emerging use case

- In your case, you probably just want to enable your GridPP elements
  - But likely to need deployment in campus core first, whether Science DMZ used or not

# What is Science DMZ?


Border Router
Enterprise Border Router/Firewall
WAN
10G Routed
10GE
Clean, High-bandwidth path to/from WAN
Site / Campus access to Science DMZ resources
10GE
perfSONAR
10G Virtual Circuit
10GE
Site / Campus LAN
Dedicated path for virtual circuit traffic
Science DMZ Switch/Router
10GE
Nx10GE
perfSONAR
Per-service security policy control points
High performance Data Transfer Node with high-speed storage
Site/Campus Virtual Circuits
High Latency WAN Path
Low Latency LAN Path
High Latency VC Path

» It’s a design pattern published by Esnet in 2013

» Principles:

1. Optimise network for science transfers; ‘onramp’ at edge
2. Tune DTN endpoints
3. Measure with perfSONAR
4. Apply security, efficiently

» Then add IPv6! ☺

- › Quite a nice, constrained deployment area for a campus

# Getting an IPv6 prefix

- To deploy, you'll need an IPv6 prefix for your site
- If your Janet-connected site doesn't already have a prefix, it's very easy to get a /48 from Jisc
- Send an email to Janet Service Desk
  - to service@ja.net, or directly to ipaddress@ja.net
- They will reply with a username and password for a web form – it's a fairly simple process ☺
- See https://www.jisc.ac.uk/contact

# Advice on address planning?

- A very good guide available from the RIPE-NCC:
  - https://labs.ripe.net/Members/steffann/preparing-an-ipv6-addressing-plan
- Also see a video of a recent UKNOF talk:
  - https://www.youtube.com/watch?v=lWFcIk4oMMU

- Lots of ways to be "clever"
  - e.g., embed VLAN IDs into the 16-bits of subnet space
- Can choose to plan by topology, or by administrative functions
  - May just assign a /64 IPv6 prefix to each existing IPv4 subnet
  - Should be able to route prefixes in a typical campus without aggregation
  - Might for example get a /56 allocated from your campus prefix for use by GridPP systems
- Other considerations
  - ULAs (RFC 4193) currently not widely deployed; no evidence of use within UK universities
  - No need for IPv6 PI space (or LIR status) for most Janet-connected sites

# Adding IPv6 DNS entries

- » Adding IPv6 DNS records is similar to IPv4
  - › Just add IPv6 AAAA (‘quad A’) records where you would normally add IPv4 A records, e.g.

```
$ dig -t any websites1.ecs.soton.ac.uk
websites1.ecs.soton.ac.uk. 1800      IN      AAAA    2001:630:d0:f104::80e
websites1.ecs.soton.ac.uk. 1800      IN      A       152.78.189.43
```

- » You need to arrange and configure forward and reverse DNS delegations
  - › Using the same procedure as for IPv4
  - › Reverse DNS sits under ip6.arpa, using nibble-based delegations

# Host configuration?

- Choices:
  - Manual address configuration (you may do this now for IPv4, esp. for servers?)
  - Use SLAAC
  - Use DHCPv6
- Remember you'll need to run RAs on the subnet router regardless of use of DHCPv6 or SLAAC for addresses
- In practice, in a typical campus dual-stack network you'll see:
  - IPv4 address by DHCP
  - IPv6 address by SLAAC
  - Other configuration from (IPv4) DHCP
- If you run IPv6-only (no IPv4) then you'll need at least stateless DHCPv6
  - You may want to explore RFC 8106 (DNS resolver option for RAs)
  - IPv6-only is the end-game; question is at what point it's practical to deliver
  - Ideally, don't really want to be translating high-volume science traffic

# Other deployment considerations?

- ULAs?
  - Probably no need; just use global addresses
  - Could use for systems that will never communicate externally
  - Not designed for IPv6 NAT
- Use of privacy addresses?
  - Should disable these for non-user systems; helps simplify management
- Enabling IPv6 for a service?
  - Enable IPv6 on the system; add DNS entry to 'advertise' IPv6 capability
  - Ensure all services running on that hostname support IPv6 before adding the entry
- Routing?
  - Might use static routing for a simple deployment
  - If using OSPFv2 for IPv4, you can run OSPFv3 alongside for IPv6, or use multi-AF IPv4/IPv6 support in OSPFv3

# IPv6 Deployment Status

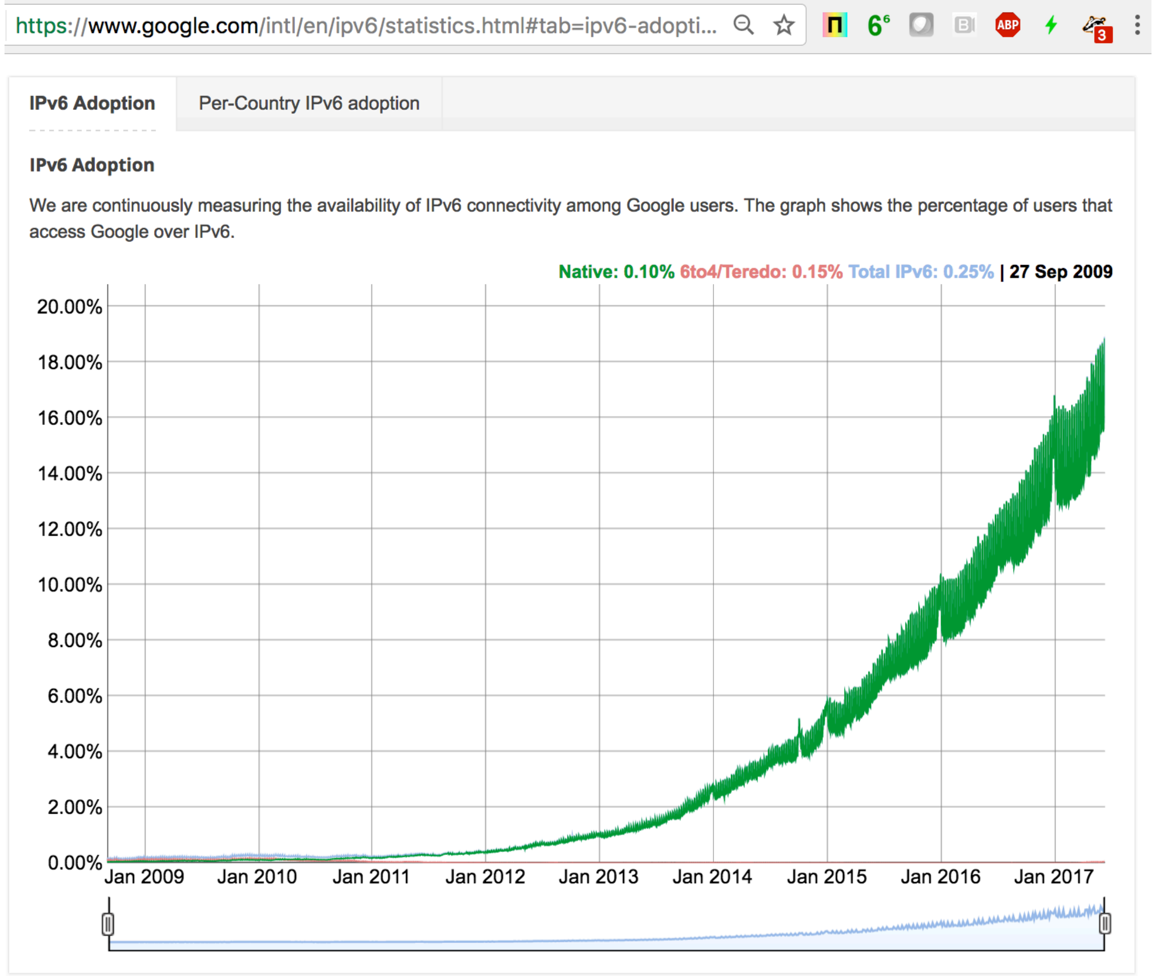
# IPv6 deployment

- Early IPv6 adoption was largely by the academic networks
- Commercial deployment was, until recently, slower
  - Larger UK ISPs now starting to move – Sky was first with over 4M users enabled, BT now ready
  - See www.ipv6.org.uk for information on UK IPv6 deployment; running security workshop on July 12th
  - UK now at around 20% deployment; Janet sites still under 5% IPv6
  - Though some high volume examples, such as approx 40Gbps IPv6 achieved by Imperial

- Significant ongoing activity by content providers
  - e.g. Google, Comcast, Facebook, Netflix, Microsoft, and Akamai / Cloudflare CDNs

- Measurement examples:
  - https://labs.ripe.net/Members/mirjam/content-ipv6-measurement-compilation (RIPE NCC)
  - http://www.worldipv6launch.org/measurements/ (ISOC World IPv6 Launch site)

# Google IPv6 statistics



https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adopti...

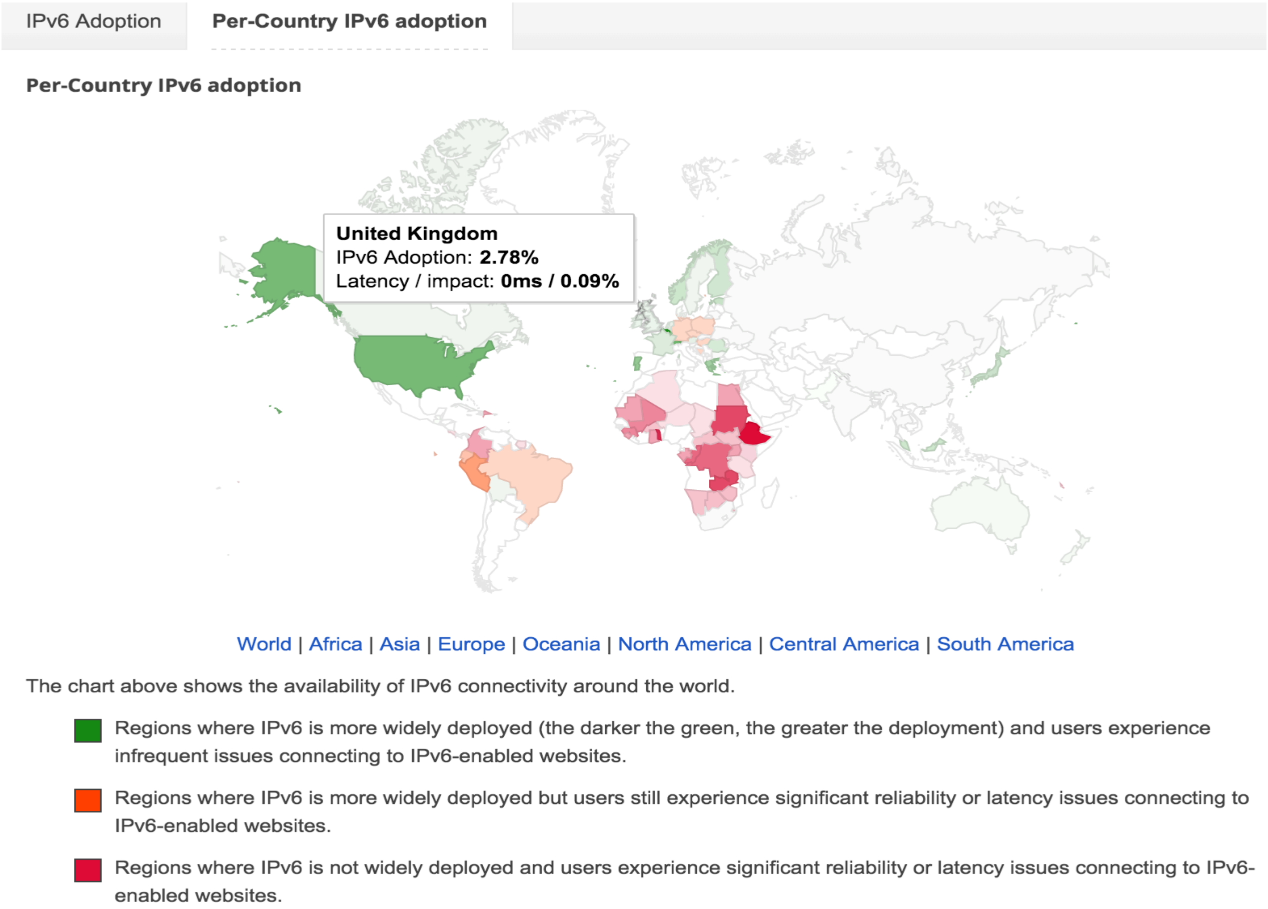
**IPv6 Adoption** Per-Country IPv6 adoption

**IPv6 Adoption**

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.

**Native: 0.10% 6to4/Teredo: 0.15% Total IPv6: 0.25% | 27 Sep 2009**

# Google stats for UK, late 2015



IPv6 Adoption | **Per-Country IPv6 adoption**

**Per-Country IPv6 adoption**

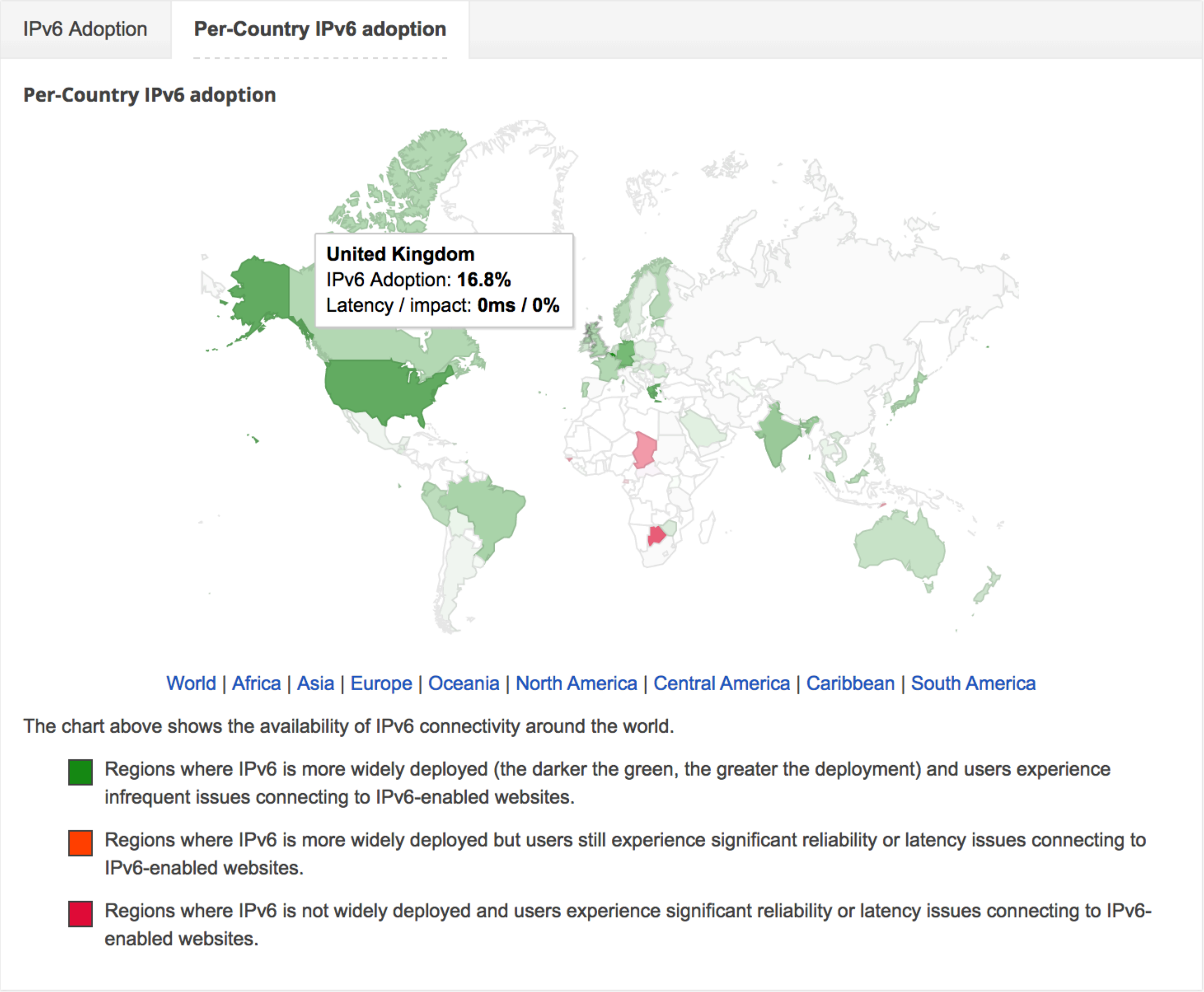
**United Kingdom**
IPv6 Adoption: **2.78%**
Latency / impact: **0ms / 0.09%**

World | Africa | Asia | Europe | Oceania | North America | Central America | South America

The chart above shows the availability of IPv6 connectivity around the world.

- 🟩 Regions where IPv6 is more widely deployed (the darker the green, the greater the deployment) and users experience infrequent issues connecting to IPv6-enabled websites.
- 🟧 Regions where IPv6 is more widely deployed but users still experience significant reliability or latency issues connecting to IPv6-enabled websites.
- 🟥 Regions where IPv6 is not widely deployed and users experience significant reliability or latency issues connecting to IPv6-enabled websites.

# Google stats for UK, Jun 2017



IPv6 Adoption | **Per-Country IPv6 adoption**

**Per-Country IPv6 adoption**

**United Kingdom**
IPv6 Adoption: **16.8%**
Latency / impact: **0ms / 0%**

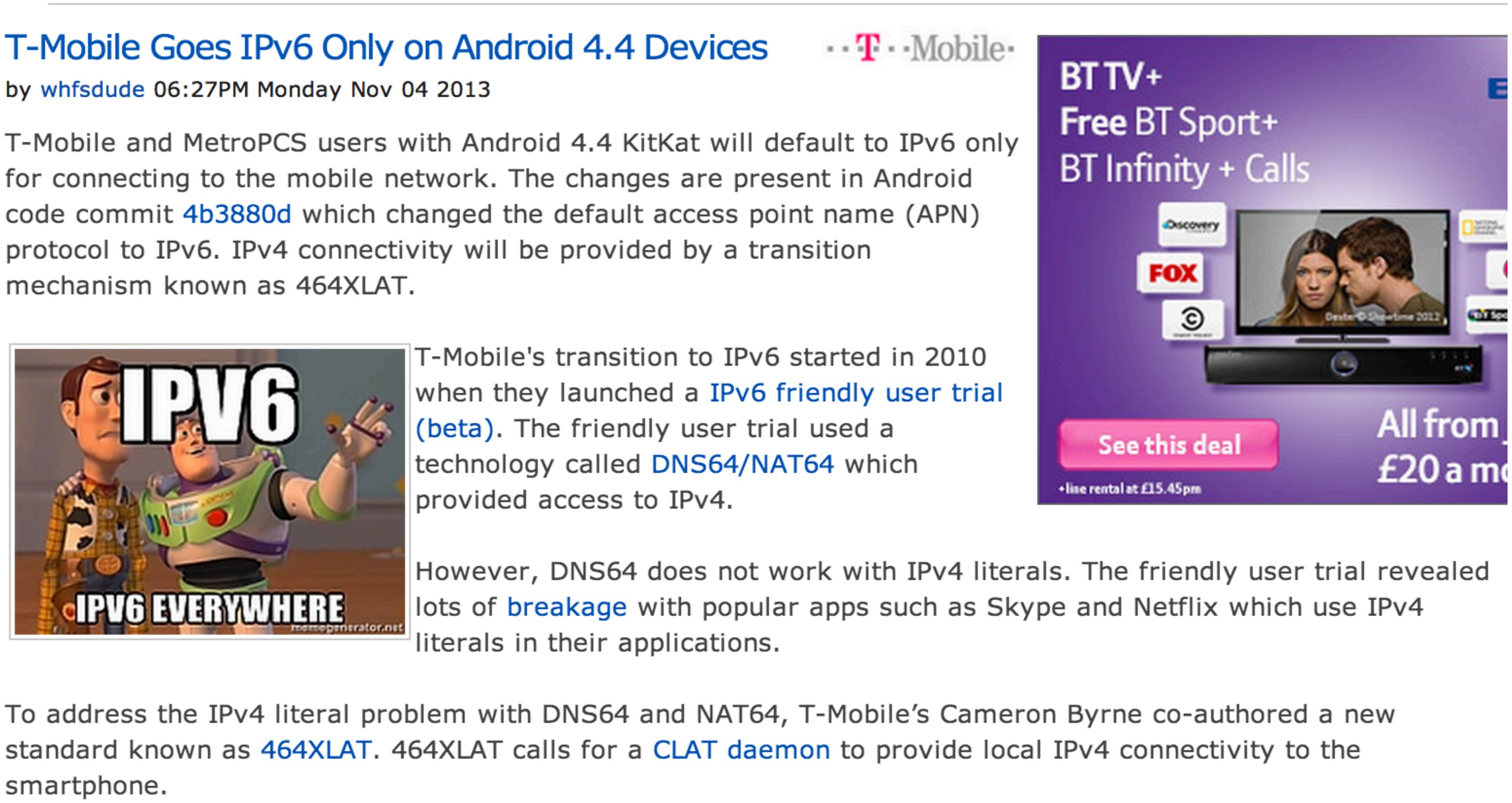
World | Africa | Asia | Europe | Oceania | North America | Central America | Caribbean | South America

The chart above shows the availability of IPv6 connectivity around the world.

- 🟩 Regions where IPv6 is more widely deployed (the darker the green, the greater the deployment) and users experience infrequent issues connecting to IPv6-enabled websites.
- 🟧 Regions where IPv6 is more widely deployed but users still experience significant reliability or latency issues connecting to IPv6-enabled websites.
- 🟥 Regions where IPv6 is not widely deployed and users experience significant reliability or latency issues connecting to IPv6-enabled websites.

# 2013: T-Mobile went IPv6-only on Android

## IPv6 WAN now uses IPv6-only, with NAT64/DNS64



### T-Mobile Goes IPv6 Only on Android 4.4 Devices

by whfsdude 06:27PM Monday Nov 04 2013

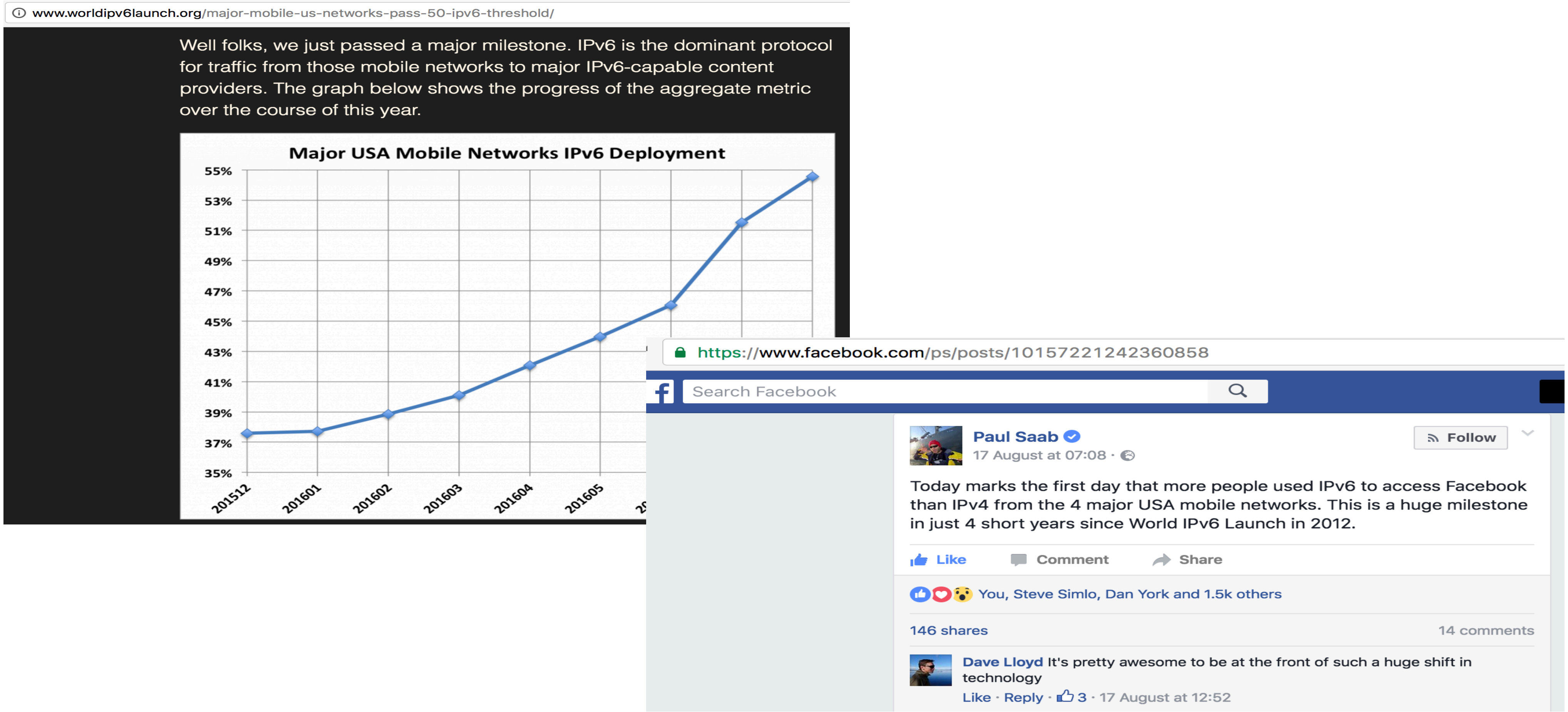
T-Mobile and MetroPCS users with Android 4.4 KitKat will default to IPv6 only for connecting to the mobile network. The changes are present in Android code commit 4b3880d which changed the default access point name (APN) protocol to IPv6. IPv4 connectivity will be provided by a transition mechanism known as 464XLAT.

T-Mobile's transition to IPv6 started in 2010 when they launched a IPv6 friendly user trial (beta). The friendly user trial used a technology called DNS64/NAT64 which provided access to IPv4.

However, DNS64 does not work with IPv4 literals. The friendly user trial revealed lots of breakage with popular apps such as Skype and Netflix which use IPv4 literals in their applications.

To address the IPv4 literal problem with DNS64 and NAT64, T-Mobile’s Cameron Byrne co-authored a new standard known as 464XLAT. 464XLAT calls for a CLAT daemon to provide local IPv4 connectivity to the smartphone.

# 2016: IPv6 is dominant protocol in US mobile



www.worldipv6launch.org/major-mobile-us-networks-pass-50-ipv6-threshold/

Well folks, we just passed a major milestone. IPv6 is the dominant protocol for traffic from those mobile networks to major IPv6-capable content providers. The graph below shows the progress of the aggregate metric over the course of this year.

https://www.facebook.com/ps/posts/10157221242360858

**Paul Saab**
17 August at 07:08 ·

Today marks the first day that more people used IPv6 to access Facebook than IPv4 from the 4 major USA mobile networks. This is a huge milestone in just 4 short years since World IPv6 Launch in 2012.

You, Steve Simlo, Dan York and 1.5k others

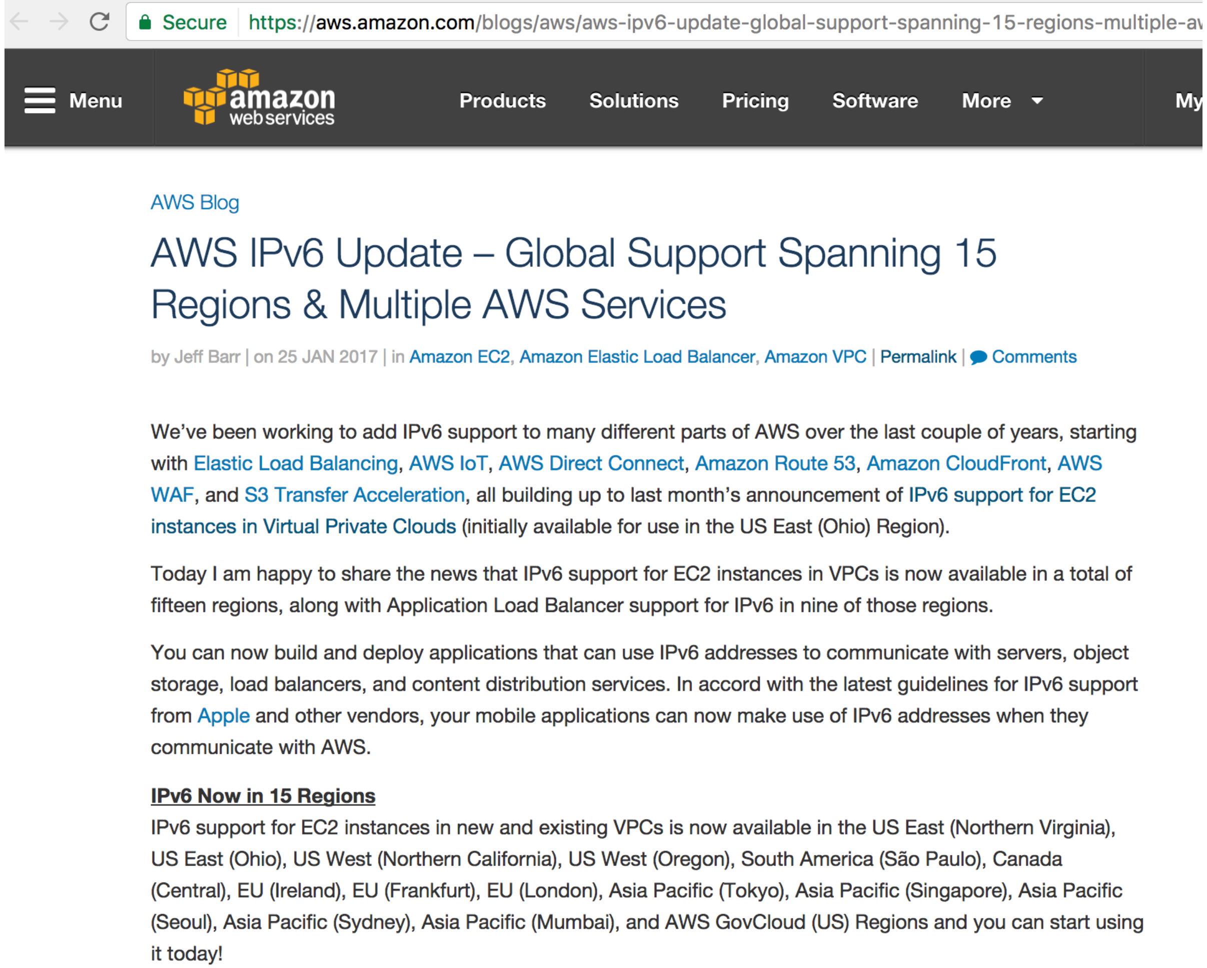
146 shares

14 comments

**Dave Lloyd** It's pretty awesome to be at the front of such a huge shift in technology

Like · Reply · 3 · 17 August at 12:52

# Cloud: significant recent progress



Secure
https://aws.amazon.com/blogs/aws/aws-ipv6-update-global-support-spanning-15-regions-multiple-av
Menu
amazon web services
Products
Solutions
Pricing
Software
More
My

AWS Blog

## AWS IPv6 Update – Global Support Spanning 15 Regions & Multiple AWS Services

by Jeff Barr | on 25 JAN 2017 | in Amazon EC2, Amazon Elastic Load Balancer, Amazon VPC | Permalink | Comments

We've been working to add IPv6 support to many different parts of AWS over the last couple of years, starting with Elastic Load Balancing, AWS IoT, AWS Direct Connect, Amazon Route 53, Amazon CloudFront, AWS WAF, and S3 Transfer Acceleration, all building up to last month's announcement of IPv6 support for EC2 instances in Virtual Private Clouds (initially available for use in the US East (Ohio) Region).

Today I am happy to share the news that IPv6 support for EC2 instances in VPCs is now available in a total of fifteen regions, along with Application Load Balancer support for IPv6 in nine of those regions.

You can now build and deploy applications that can use IPv6 addresses to communicate with servers, object storage, load balancers, and content distribution services. In accord with the latest guidelines for IPv6 support from Apple and other vendors, your mobile applications can now make use of IPv6 addresses when they communicate with AWS.

**IPv6 Now in 15 Regions**
IPv6 support for EC2 instances in new and existing VPCs is now available in the US East (Northern Virginia), US East (Ohio), US West (Northern California), US West (Oregon), South America (São Paulo), Canada (Central), EU (Ireland), EU (Frankfurt), EU (London), Asia Pacific (Tokyo), Asia Pacific (Singapore), Asia Pacific (Seoul), Asia Pacific (Sydney), Asia Pacific (Mumbai), and AWS GovCloud (US) Regions and you can start using it today!

# Questions?

Email: tim.chown@jisc.ac.uk