IPv6

Geoff Huston
APNIC

Why?

Because we ran out of IP addresses

Again.

We've been here before ...

The original ARPAnet design from 1969 used the NCP protocol, which used 8 bit addresses in the NCP packet header

- Maximum network span of 256 nodes
- Enough?

- Well yes, because at the time computers were the size of entire rooms, cost many millions of dollars and there were only a few thousand in the entire world.
- At the time the entire concept of shrinking the computer to something you could hold in one hand and trying to connect billions of them together was just too far into the future to worry about

Transition V1.0

- Turns out that 8 bits of addresses was not enough for the next generation of mini-computers
- ARPAnet undertook a transition from NCP to a new protocol: TCP/IP
  - Expansion from 8-bit to 32-bit addresses
  - Flag Day: 1 January 1983
  - Shutdown and reboot every node into the new protocol!

Vint Cerf, APRICOT, Feb 2011

“This time, for sure!” *

* Actually Vint Cerf did not say that!

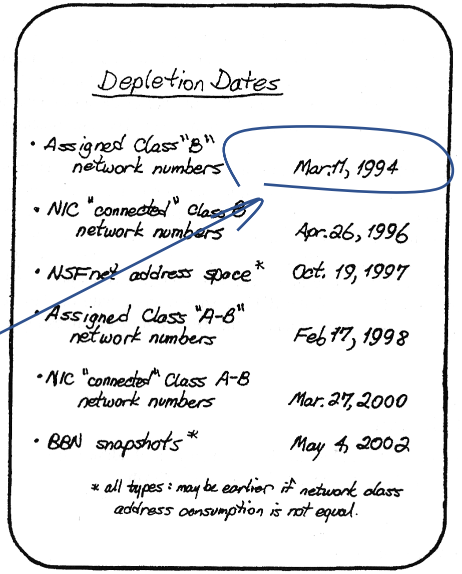
* Actually Vint Cerf didn’t say this!

IP Version 4

- 32-bit address field
  - That’s 4,294,967,296 addresses
- In 1983 that looked like a HUGE number of computers

Digital Pressures

- Through the 1980's computers changed from large expensive pieces of ironware to desktop consumer products
    - And the size of the computer market changed from thousands to tens of millions
- And the prospect was smaller, cheaper and more
    - So by 1990 that 4 billion address space was looking pretty small
    - And it was not going to last much longer!



Frank Solensky – IETF 1990
Presentation on IPv4 Address Depletion Predictions

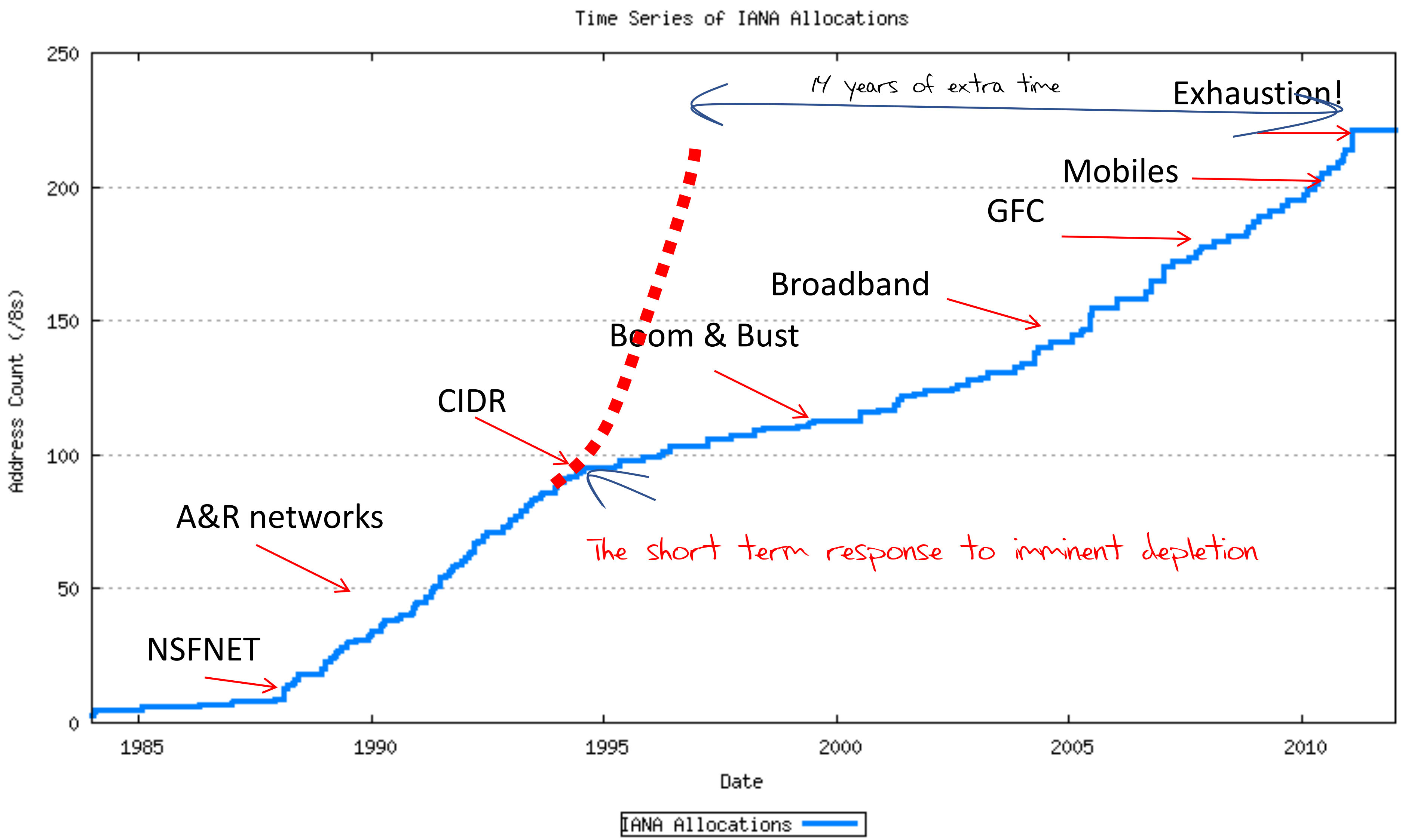
IPocalypse?

So we hit the wall - right?

Panic!

- This was a brutal wakeup call
- We had hardly started with the Internet and its demise was just 4 years away!
- So we rapidly worked on short term responses to push this exhaustion date out
- To give us more time to work on the longer term solutions
- So:
  - We dropped the classful address plan
  - We introduced NATs to allow address sharing
  - We worked on a new protocol

The short-term band aid worked!



That longer term plan

- Was to develop a new IP protocol
- And then transition the Internet over to use this new protocol

What are we transitioning to?

IPv6!

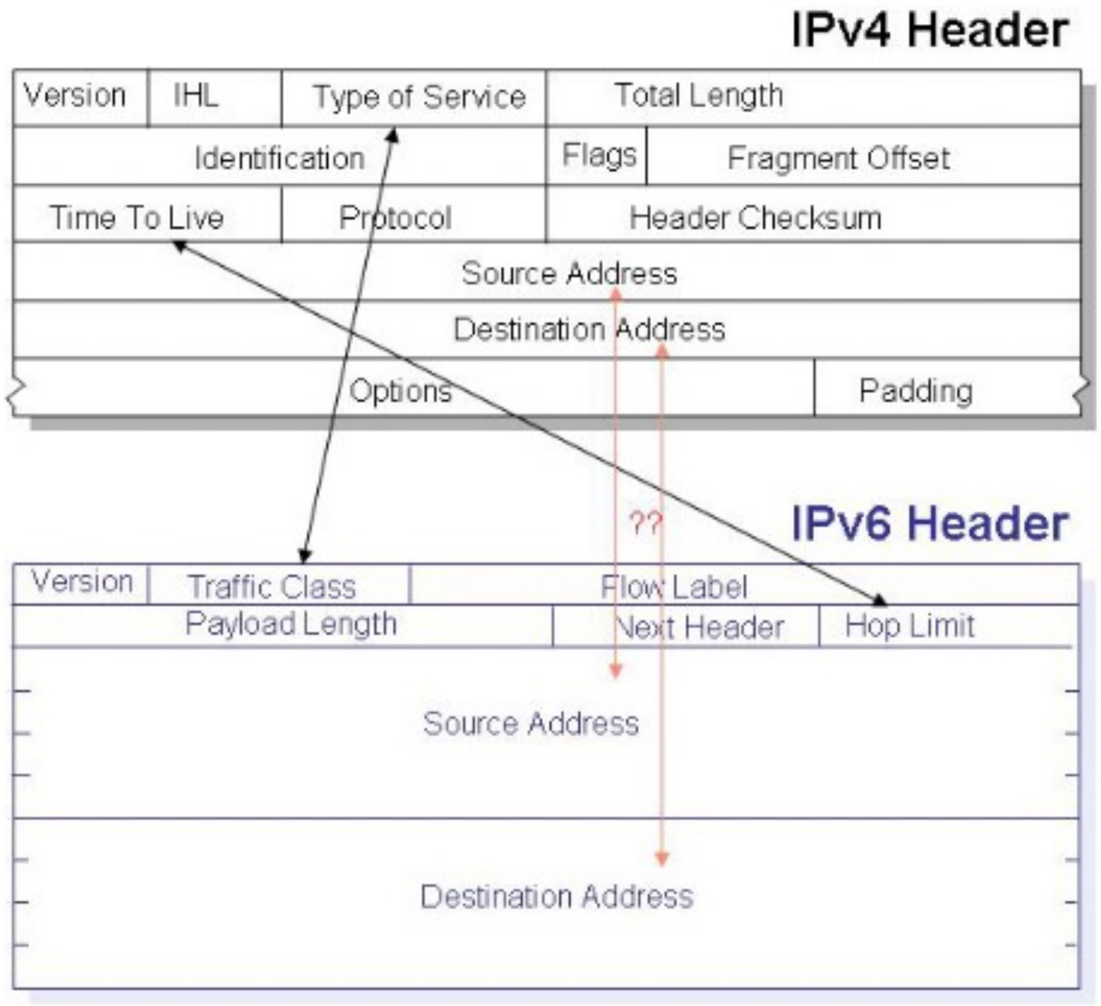
IPv6 is…

- IPv4 with larger address fields
- Not much else changed
  - It’s still an address-based stateless datagram forwarding protocol
  - It still has decoupled forwarding, routing, naming, transport
  - The interfaces to the underlying media protocols are largely unchanged
  - The APIs to interface to the upper layers are largely unchanged
- So transition to IPv6 should be easy

Aside: "Not much else changed"

Aside: "Not much else changed"

What actually changed with IPv6:



Aside: "Not much else changed"

What changed with IPv6:

- 128 bit address fields
- Fixed host/network boundary
- Replace Broadcast and ARP with Multicast and SLAAC
- Removed on-the-fly fragmentation with source fragmentation
- No NATS!
- Flow Label
- Make Fragmentation Controls an optional Extension Header
- Change “Options” to “Extension Headers”
- Multi-Addressing
- Scoped Addresses

128 bit addresses

Network ID	Interface ID
64 bits	64 bits

- We experimented with a constant interface ID for a while, and realised that this was an unacceptable privacy leak if the value was held constant
- So clients use a temporary random interface identifier for the low order 64 bits
- Which means that the constant length of an IPv6 address is back to 64 bits

The Flow ID

- Huh?
- It probably sounded like a reasonable thing at the time
- But no one remembers what its good for!
- So that’s 20 waste bits in the header!

Extension Headers

- Don’t work on today’s silicon!
- High speed switching gear that wants to preserve TCP and UDP packet flows need to see the TCP and UDP headers at a fixed offset in the packet header
- Extension headers are a chained header structure where each header points to the next header, with the transport header at the end of the chain
- Unravelling that chain in very high speed silicon is a challenging task
- Easier to drop all IPv6 packets with extension headers!

Fragmentation Control

- Packet Fragmentation was a big feature of the IPv4 design
  - It allowed new media types to the integrated into the Internet environment seamlessly
  - But it had a performance cost
- The IPv6 design disallowed fragmentation on the fly
  - If a packet is too big for the next hop you need to send a signal to the source and discard the too-big packets
  - This is slow and unreliable
  - The signalling is also uinreliable
  - So the best response is to avoid packet fragmentation completely
  - Which means that packets in IPv6 tend be 1,280 octets
  - Which has a performance cost

No NATs!

- Really?
- We’ve grown addicted to NATs and IPv6 NATs are already a common feature of IPv6

Aside: "Not much else changed"

What's giving us grief with IPv6:

- 128 bit address fields
- Fixed host/network boundary
- Replace Broadcast and ARP with Multicast and SLAAC
- Removed on-the-fly fragmentation with source fragmentation
- No NATS!
- Flow Label
- Make Fragmentation Controls an optional Extension Header
- Change "Options" to "Extension Headers"
- Multi-Addressing
- Scoped Addresses

So Transition should be easy?

Right?

So Transition should be easy?

Well, no, not really!

Why is this taking so long?

IPv6 was a minimal change to IPv4

- So early adopters found little in the way of an early adoption benefit

IPv6 was not backward compatible with IPv4

- This is a key “feature” of protocols that use fixed length addresses in their packet headers – you can’t jam all 128 bits into a 32 bit field.
- This meant that IPv6 hosts could not directly communicate with IPv4 hosts
- IPv6 adopters still had to support IPv4
- This meant that EVERYONE has to transition to dual stack BEFORE we can drop IPv4

Transition, the second time around

That last point is important – let me repeat it for you:

- A “Flag Day” switchover is impossible
- Piecemeal replacement won’t work either, as IPv6 is not backward compatible with IPv4
- So, we need to run both protocols in tandem “for a while”
- But bear in mind that one protocol has already run out of addresses
- And network growth continues at record levels

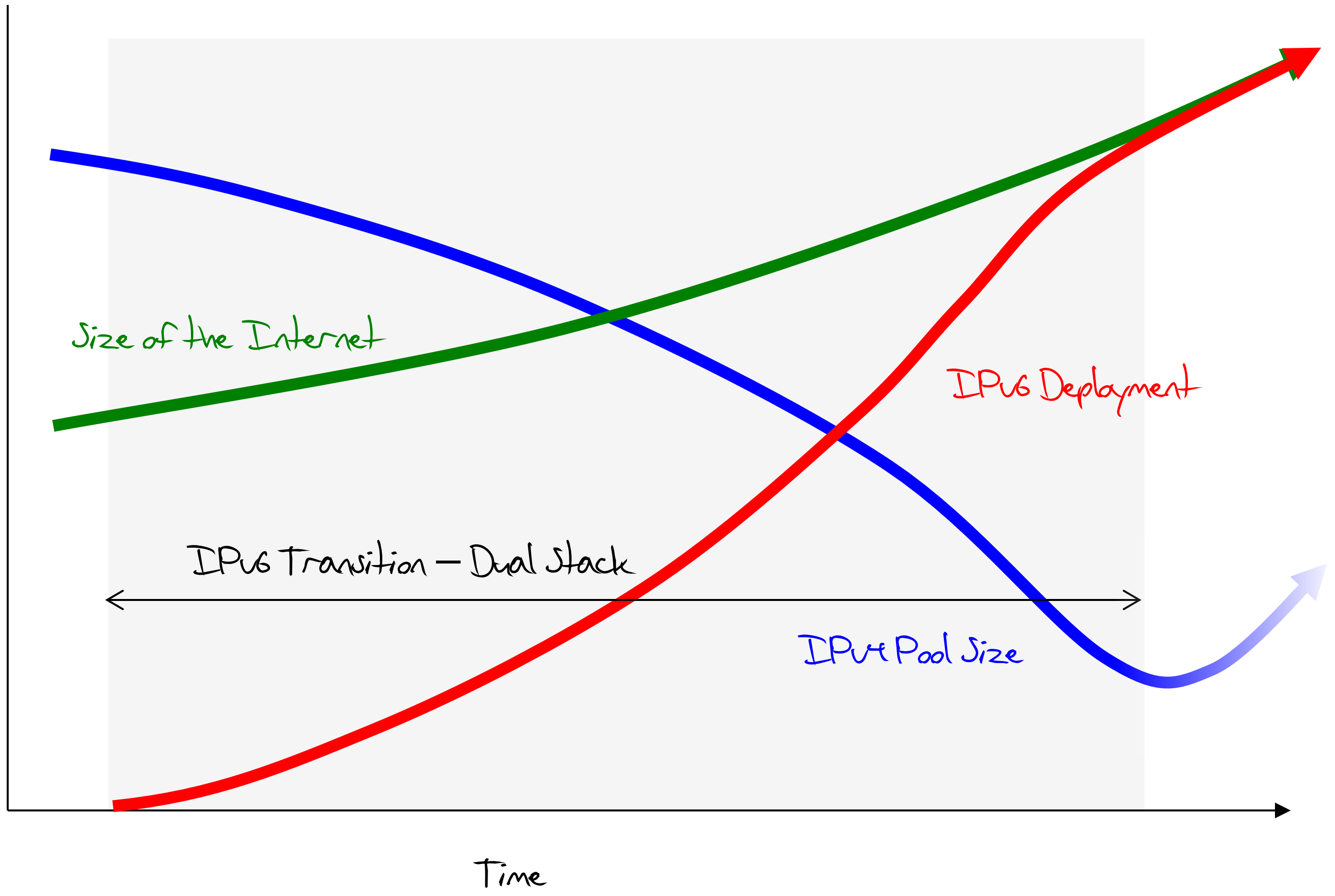
Transition, the second time around

We need to :

- deploy IPv6 in parallel with IPv4
- deploy ever more stringent IPv4 address conservation measures within the network
- allow the network to expand at an ever-increasing rate

All at the same time!

What we thought was the IPv6 Transition Plan



The Plan

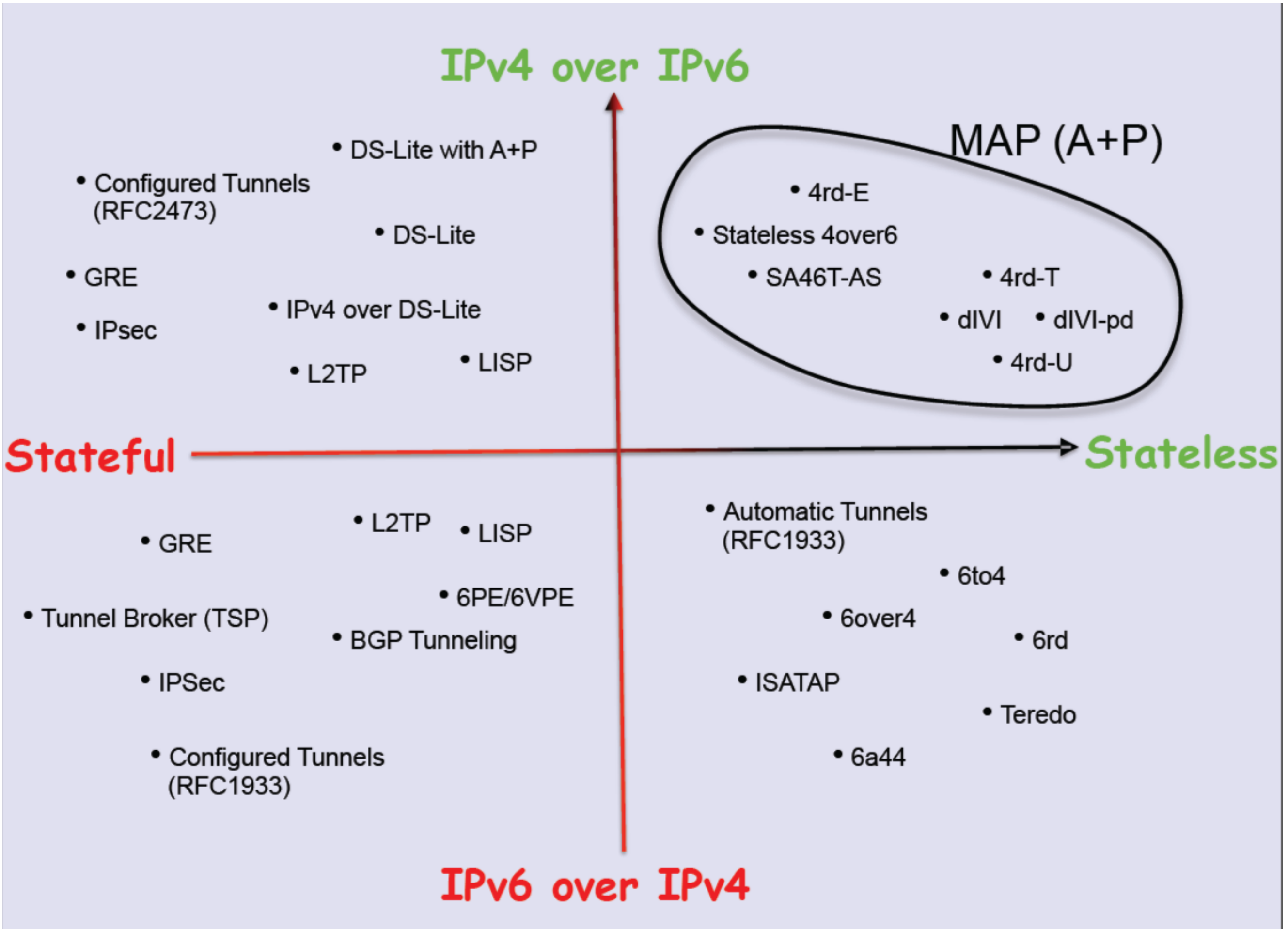
- We would build in a piecemeal fashion “islands” of dual stack networks while we still had adequate amounts of IPv4
- The IPv6 islands were meant to grow faster than the Internet itself
- Over time the dual stack “islands” would link up
- Once the entire Internet was dual stack we could drop IPv4

But…

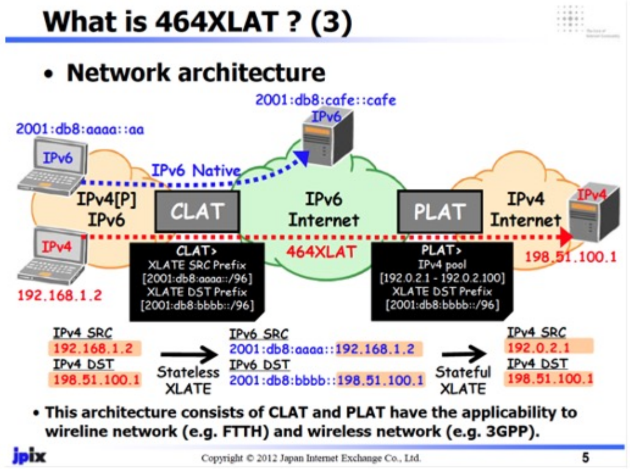
- 3G mobile networks were not designed with this transition in mind
  - A 3G network operator’s costs will double if they opt to use two protocol stacks in their network
  - So they have to tunnel IPv6 in IPv4, or tunnel IPv4 in IPv6 if they want to support IPv6
  - Which is messay
- IPv4 was running out
  - So the larger networks were using net 10 internally for their IPv4 support
  - But net 10 wasn’t big enough
  - So they needed to either internally segment their IPv4 network and reuse net 10 or run an IPv6 only internal network and tunnel IPv4 on demand wityh CGNs at the edge

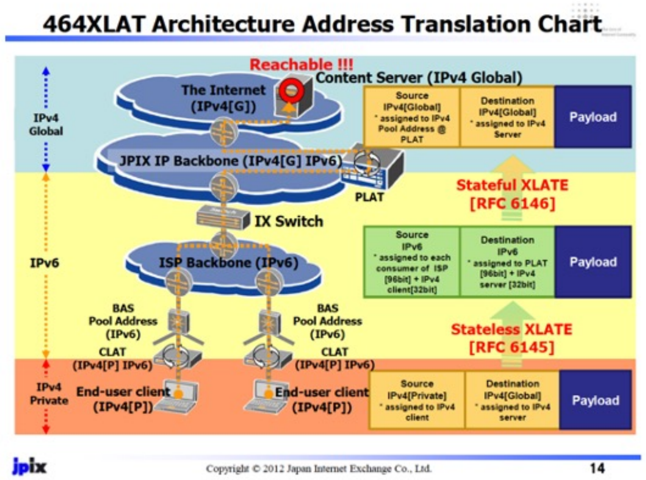
But…

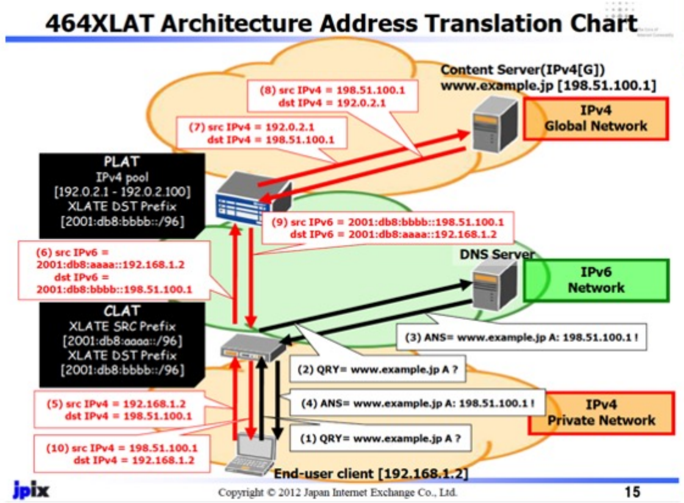
- Transition became more and more complex with multiple "solutions"



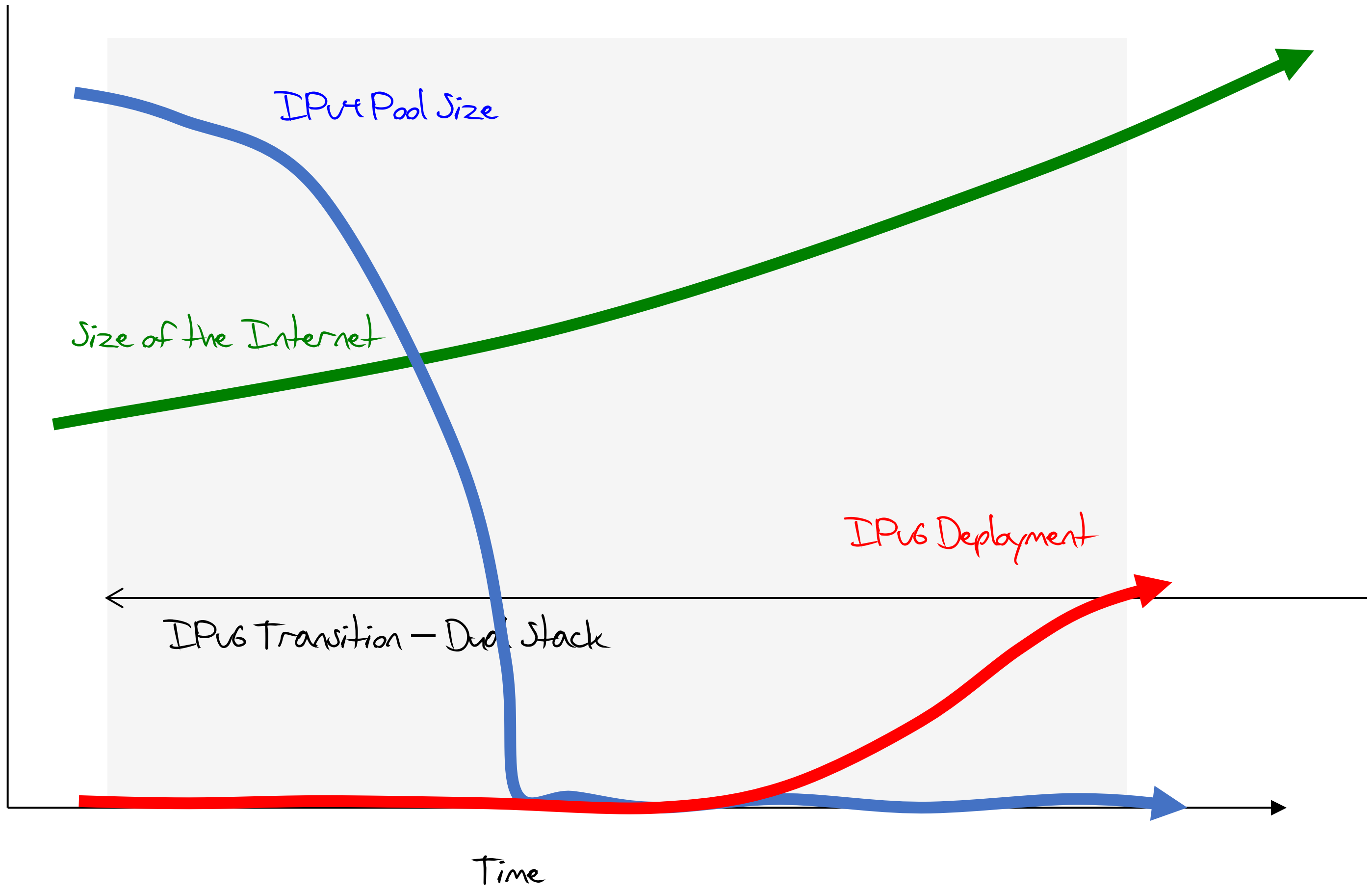
Complexity layered on complexity







The IPv6 Transition Plan as it happened!



"The Internet" does not exist

- The Internet is not a singly managed entity
- Noone is in control
- Noone is there to tell anyone else what they can (or cannot) do
- The Internet is a collection of markets (Goods and services, Transmission and switching, Technologies, …)
- We find it impossible to orchestrate collective synchronised actions across all these distinct activities
- So the status quo accumulates a huge amount of inertial momentum
- Which makes it terribly hard to change direction in an orderly manner!

The transition is more like this!


Savage Chickens
by Doug Savage
CHAINSAW JUGGLING? THAT'S EASY! I JUGGLE ANGRY FERRETS!
BUT, TO BE FAIR, THEY DON'T START OFF ANGRY
www.savagechickens.com

Or this!

Its now 2022 and we are still in this transition

We underestimated the awesome ability of NATs to squeeze out address efficiencies – the past decade of massive growth on the Internet has been largely based on various forms of NAT usage

NATs expanded the usable IPv4 address capacity by a further ~20 bits

- Each bit of increased address size doubles the total capacity of the space
- It’s taken us much longer than we thought to fill up these additional address bits
- There is no common sense of urgency here

So how is this transition going?

Slowly!

Let's look at progress so far

2012 – no significant deployment of IPv6

Let's look at progress so far

2012 2014 – still very little movement in the larger environment

Let's look at progress so far

2012 2014 2017 – India IPv6 deployment

Let's look at progress so far

2012 2014 2017 2019 – stirring in China

Where are we in 2022?

- 11 years after the depletion of the central IPv4 address pools we are still sitting on some 30% of the Internet user base with Dual Stack (IPv4 + IPv6) support
  - The other 3 billion uses are on IPv4 only
- How much longer will it take before we can call this transition "completed"?

Why is this taking so long?

- If we designed the IPv6 protocol 30 years ago
- And we've been running the IPv4 Internet on empty for many years
- Then why isn't the Internet an all-IPv6 Internet today?

Why is this taking so long?

IPv6 is not different enough

- IPv6 is just IPv4 with larger addresses
- It does not make packet switching any cheaper, so it does not give early adopters any advantage
- And IPv6 not backward compatible with IPv4, so early adopters still have to run IPv4 as well

Why is this taking so long?

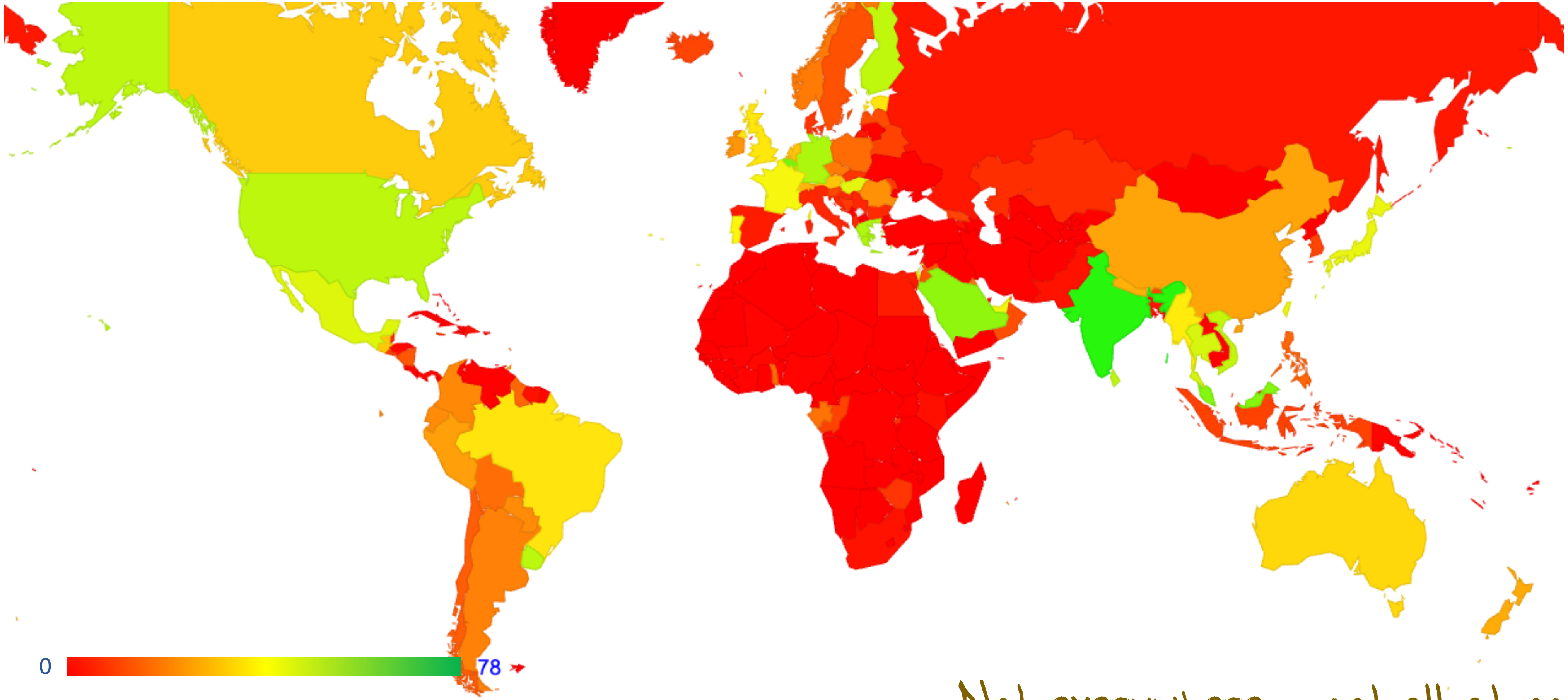
NATs are just too good!

- NATs drive every part of today’s internet, and we’ve adapted the entire IP infrastructure and application space to work in a NAT environment
- We’ve pushed the role of service identification to the name space
- And changed the architecture of the Internet into a client/server model
- And clients don’t need permanently assigned public addresses
- So NATs have allowed us to build a network with some 40 billion connected devices – and we are confident that we can grow further

Why is this taking so long

- The transition timetable is not being set by the early adopters
  - There is not enough competitive advantage for early adopters to drive the transition
- The transition timetable is not being set by the late adopters
  - At some point the last to adopt cannot impose a hold over everyone else
- So transition is a form of “majority” decision making
  - It’s a case of “critical mass” that will determine at what point dual stack service providers will contemplate dropping IPv4 and completing their part of the transition

IPv6 Adoption today



Not everywhere – not all at once
~1 B iPv6 users out of 4B

Will we complete this transition?

- Or will it just run out of momentum and we will turn our collective attention elsewhere
- But where?

If not IPv6 then what?

Some issues to think about

What matters in such a network?

- Addressing – IPv4 / IPv6 / IPv? Absolute? Relative?
  - Is universal unique end-point addressing a 1980’s concept who’s time has come and gone?
- Naming and Name Spaces – DNS evolution?
  - Are ”names” a common attribute of the network, or an attribute of a service environment?
- Referential Frameworks?
  - In a world of densely replicated service delivery points how does a client rendezvous with the “best” service point? Does the client work it out? Or the network? Or the service?

Some issues to think about

What matters in such a network?

- If we don't try to share our transactions across a common network then what are we asking from common infrastructure?
- If anything?

- Why is a 1980's networking architecture roughly modelled on telephony still relevant today?
  - How should we represent service identity and location?
  - What is the relationship between applications, hosts and networks?
    - Mutual trust or mutual suspicion?

Longer Term Trends?

Pushing EVERYTHING out of the network and over to the edge!

- Transmission infrastructure is becoming an abundant commodity
  - Sharing technology (multiplexing) is decreasingly relevant
- We have so much network and computing that we no longer have to bring consumers to service delivery points - instead, we are shifting services towards consumers and using the network to replicate servers
- With so much computing and storage the application is becoming the service, rather than just a window to a remotely operated service

Do networks matter any more?

- We have increasingly stripped out network-centric functionality in out search for lower cost, higher speed, and better agility
- We are pushing functions out to the edge and ultimately off “the network” altogether and what is left is just dumb pipes
- What defines “the Internet”?
  - A common network, a common protocol and a common protocol address pool?

  or

  - A disparate collection of services that share common referential mechanisms?

What's important in a "flat" access network?

- Unique endpoint addressing?
  - Hardly
- Routing?
  - Nope!
- Names and references?
  - That's **all** that matters!

Where are we?

- What we call the "IPv4 Internet" has changed the semantics of "an address"
  - Its not an endpoint identity
  - It's an ephemeral session token to allow the network to distinguish the traffic between various active sessions
- Much of what we build upon these days is the name infrastructure (DNS)
  - It may not have been deliberate, but we've been building name-based networks for the past decade or so
  - And so far it has been working!

Thanks!