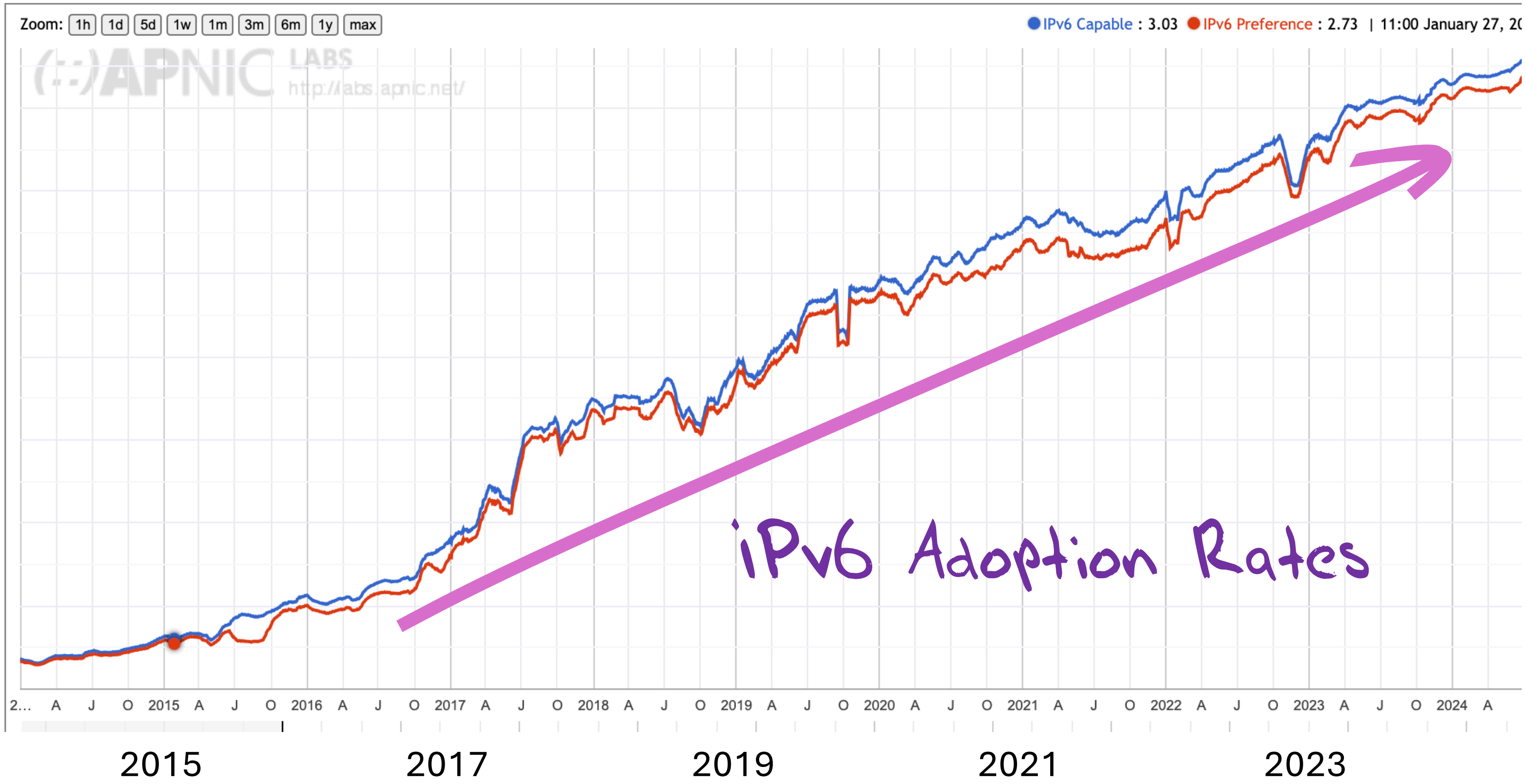
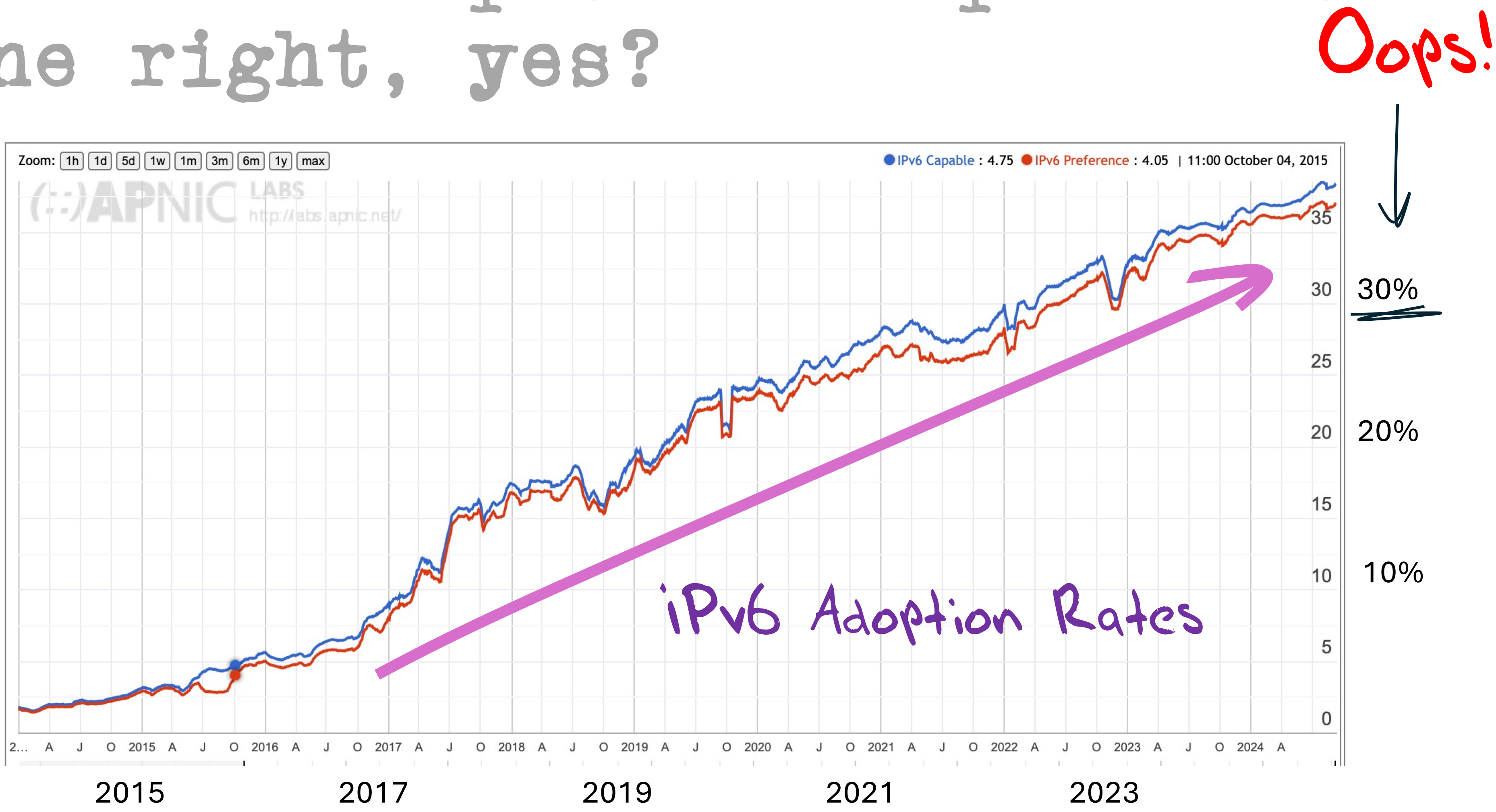
IPv6 Transition: Why is this taking SO LONG!

Geoff Huston AM
Chief Scientist, APNIC

What's the problem? Up and to the right, yes?



What's the problem? Up and to the right, yes?

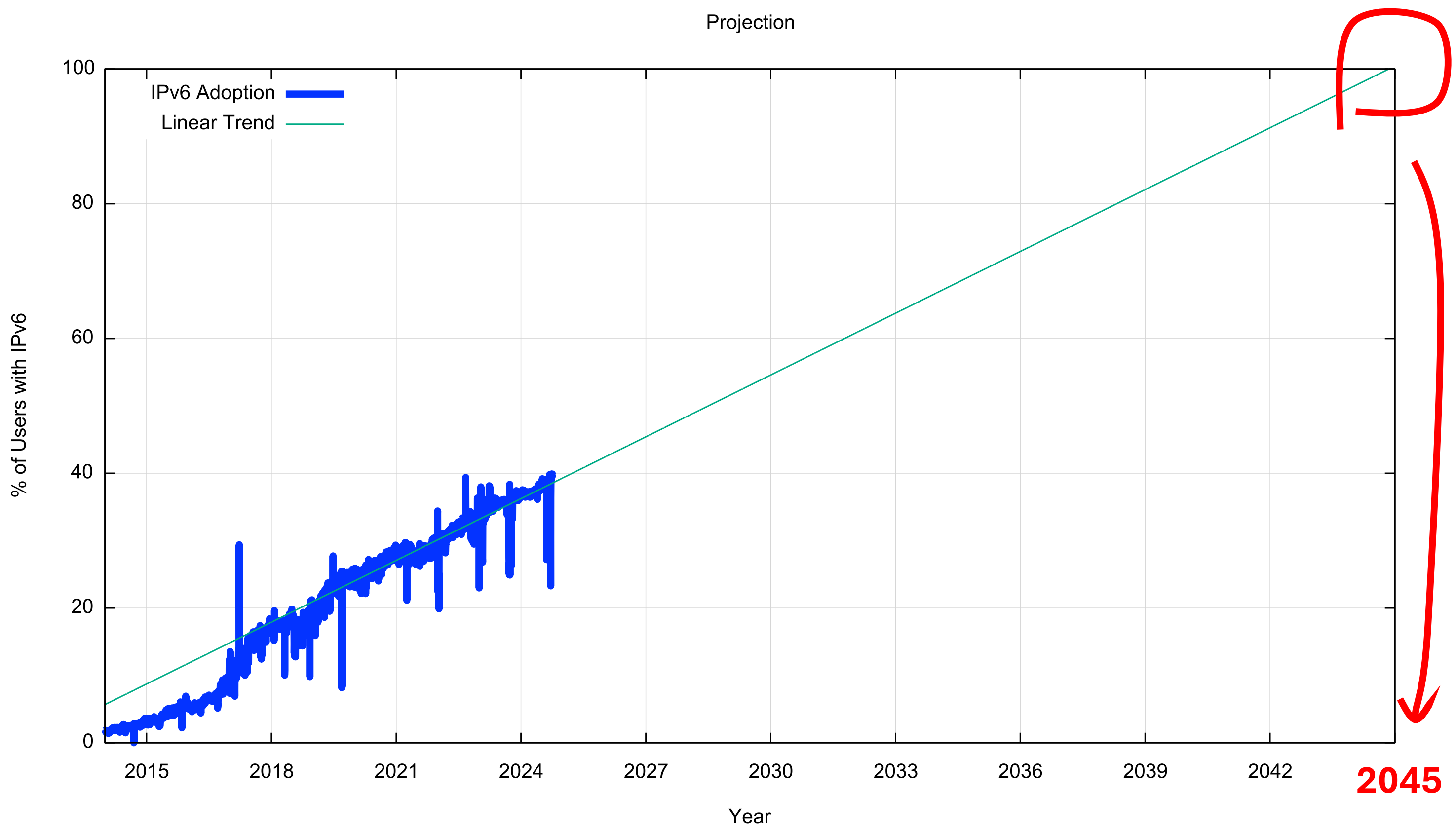


It's been around a decade…

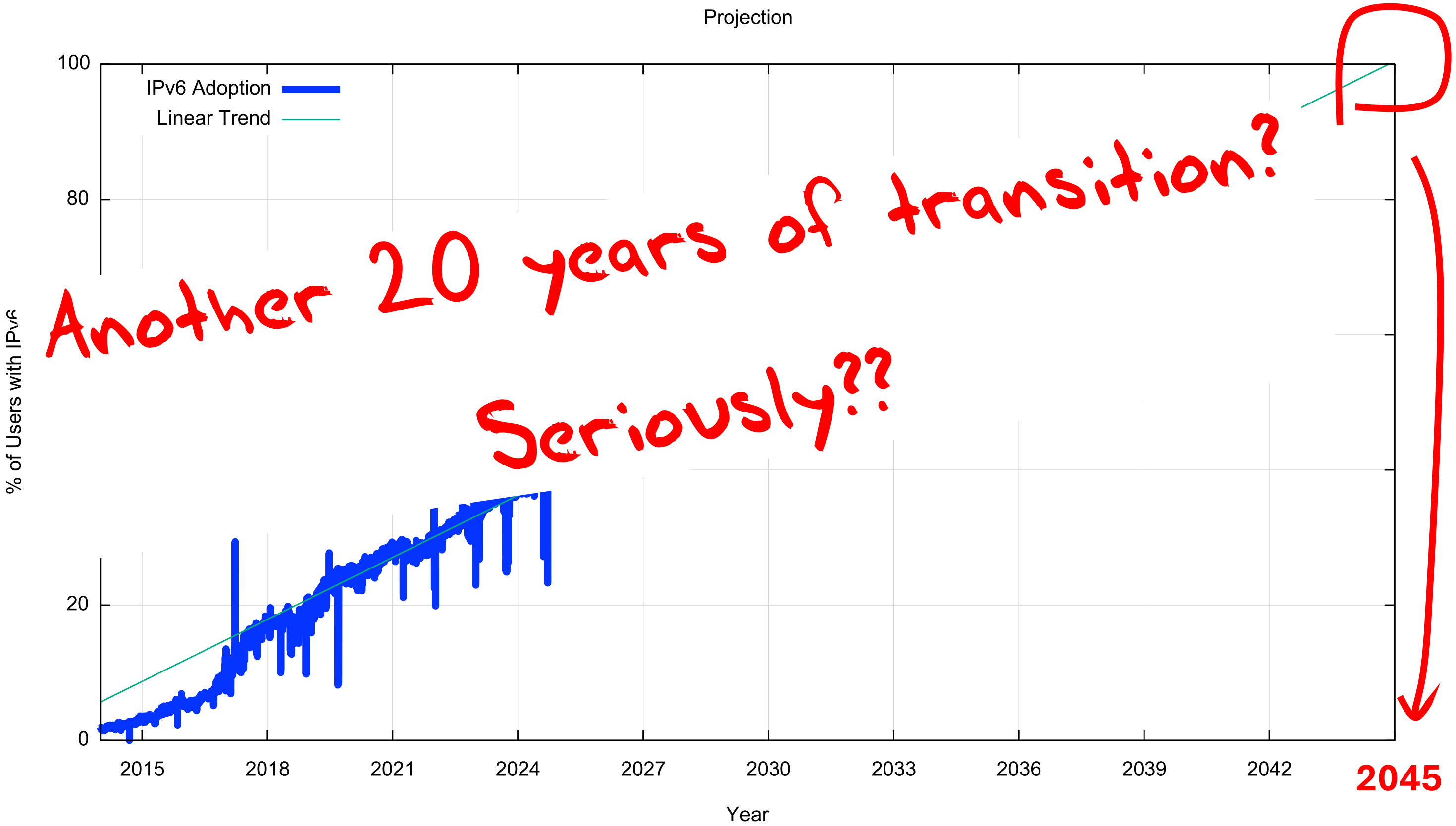
- Since the RIRs handed out their last substantial IPv4 address blocks
- We've been in a state of IPv4 "address exhaustion" for more than a decade
- And yet the global uptake rate of IPv6 is a little over one third of the Internet's user base

- **This is completely unexpected!**

Projecting this Forward

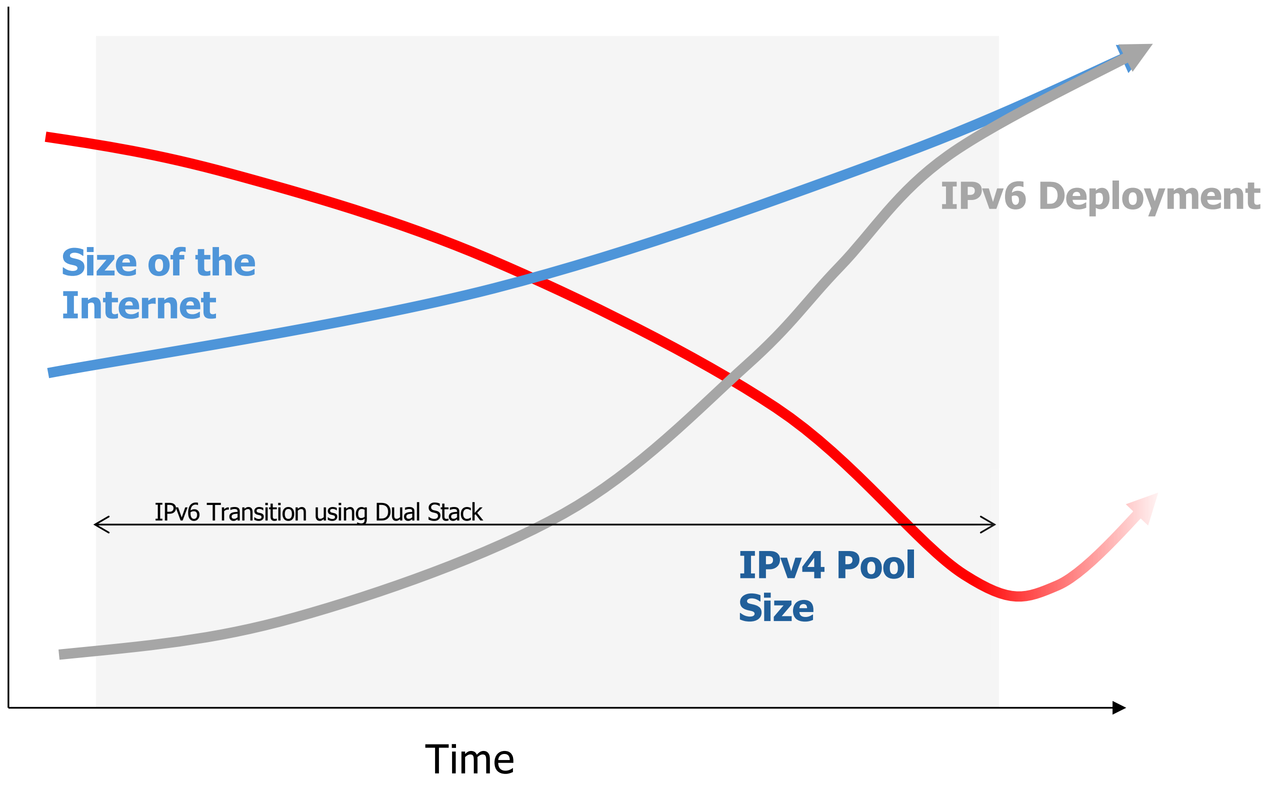


Projecting this Forward

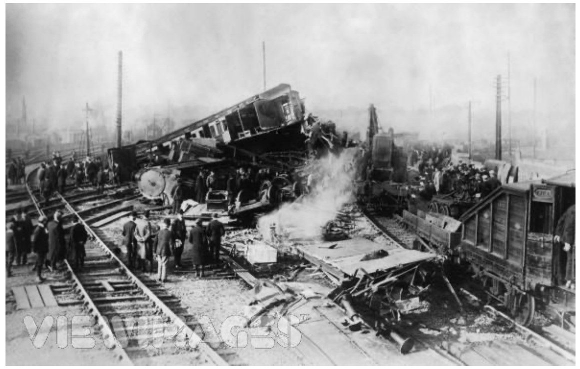


What's gone wrong here?

We had this plan …

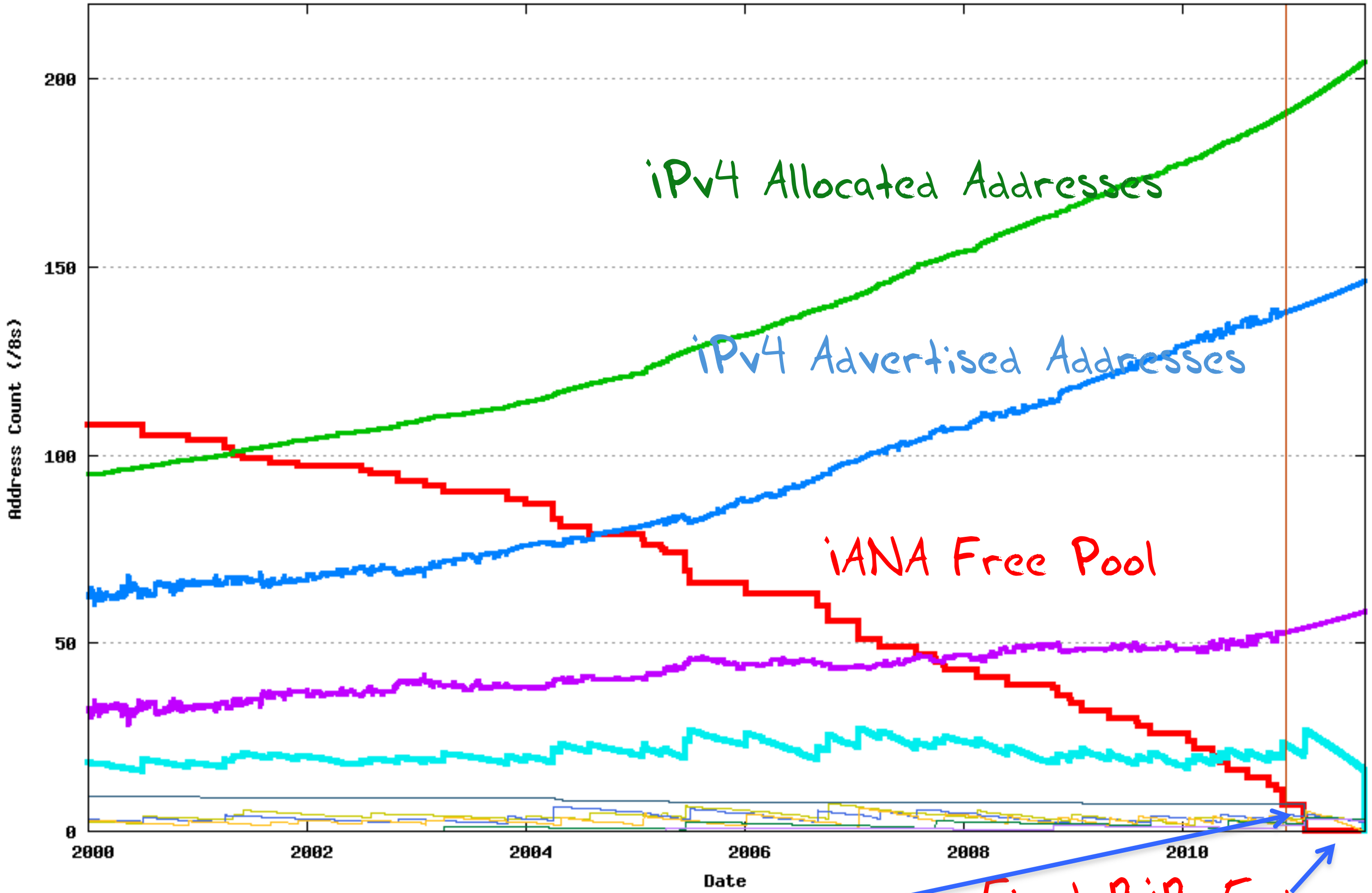


Dual Stack Transition Assumptions



- That we could drive the entire transition to IPv6 while there were still ample IPv4 addresses to sustain the entire network and its growth
- Transition would be driven by individual local decisions to deploy dual stack support
- The *entire* transition would complete *before* the IPv4 unallocated pool was exhausted!

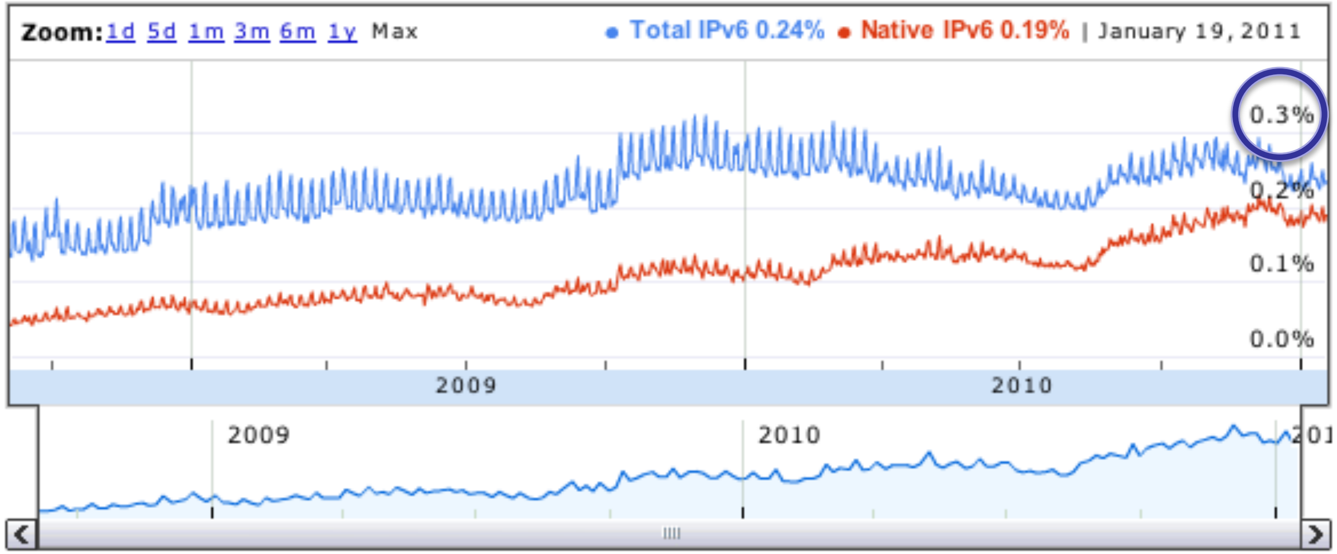
We strayed off-plan!



iANA Exhaustion February 2011

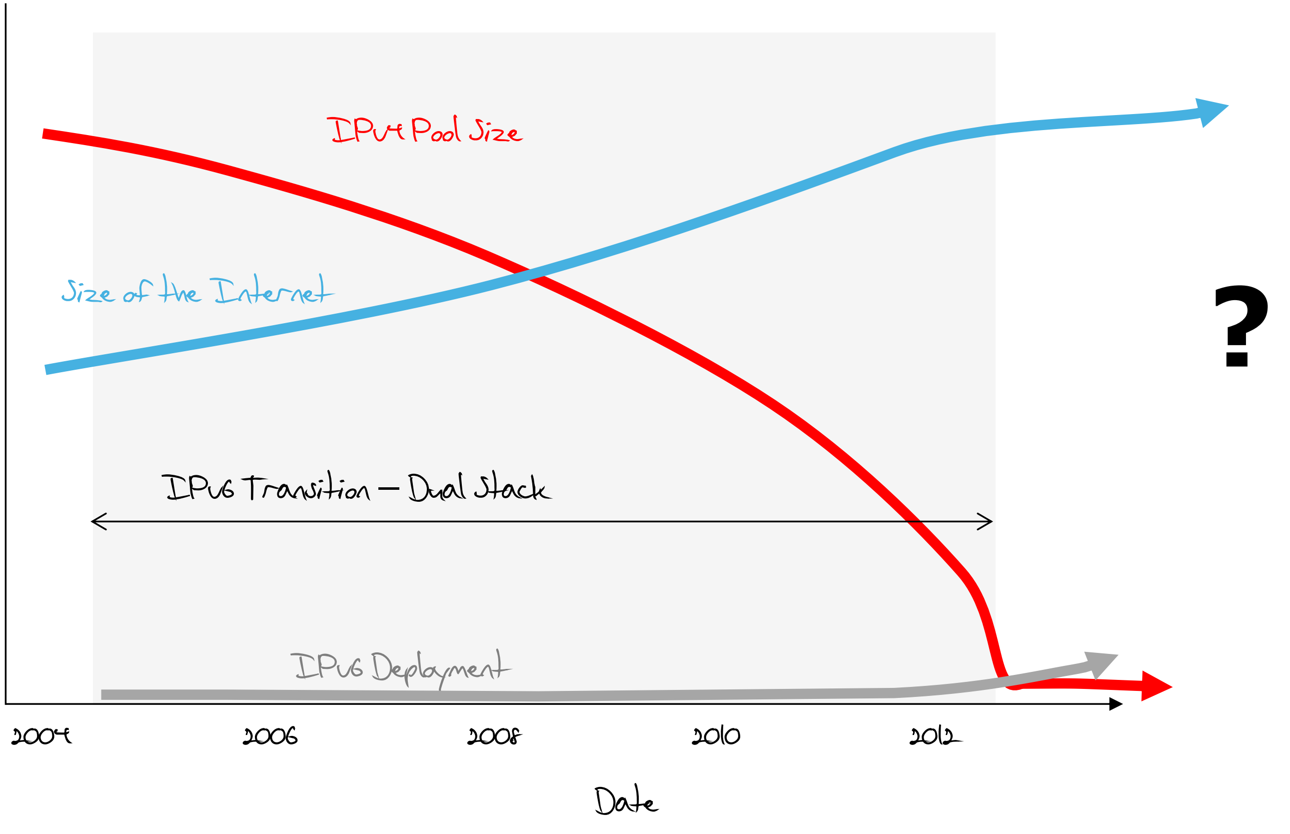
First RiR Exhaustion July 2011

Where were we with IPv6 deployment?



http://www.google.com/intl/en/ipv6/statistics/

The 2012 IPv6 Transition Plan

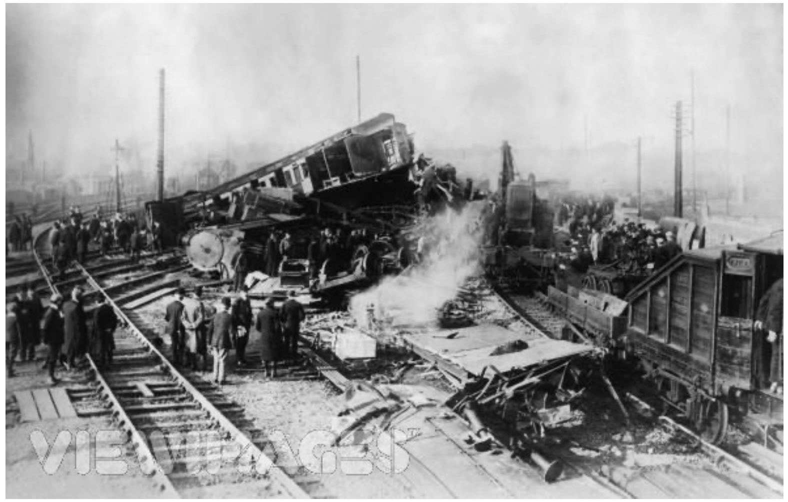


What next?

- Despite the whinging from IETF purists over the compromise of a pristine end-to-end model there really was no other option:

The answer was NATs!

NATs



- This low friction response to IPv4 address depletion had been used for more than a decade in client/server network architectures
- **Clients** initiate a service transaction and only need an external address/port binding for the duration of the transaction
- **Servers** sit in central data centres and share platform IP addresses using name-based distinguishers

Making IPv4 Last Longer with NATs


The Wreck of Maine Central Train No. 13, Oakland
(W.H. Bunting)

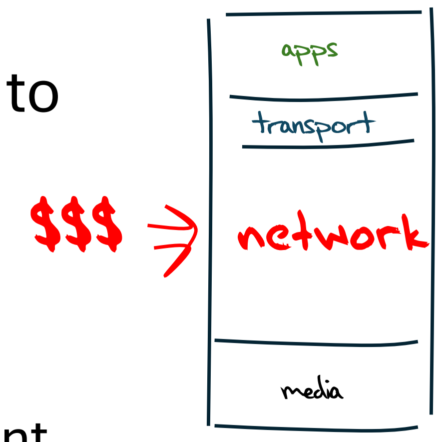
- For how long?
- For what cumulative address demand?
- For what level of fairness of access?
- At what cost?
- For whom?
- To what end?
- What if we actually achieve something different?
  - How would the Law of Unintended Consequences apply here?
  - Would this negate the entire "IPv6 is the solution" philosophy?

Because it wasn't just an IPv4 to IPv6 transition

Follow the money…

The "Classical" Internet

- IP was a ***network*** *protocol* that provided services to attached devices
- It was the role of Network Providers to allow clients to consume content and access services
    - The costs of operating the network dominated the entire cost of the Internet
    - In networking distance dominates all cost models
    - In the Internet the role of transit providers were paramount
        - We used to spend all our time talking about peering and transit
    - ISPs were the brokers of rationing the scarce resource of distance capacity

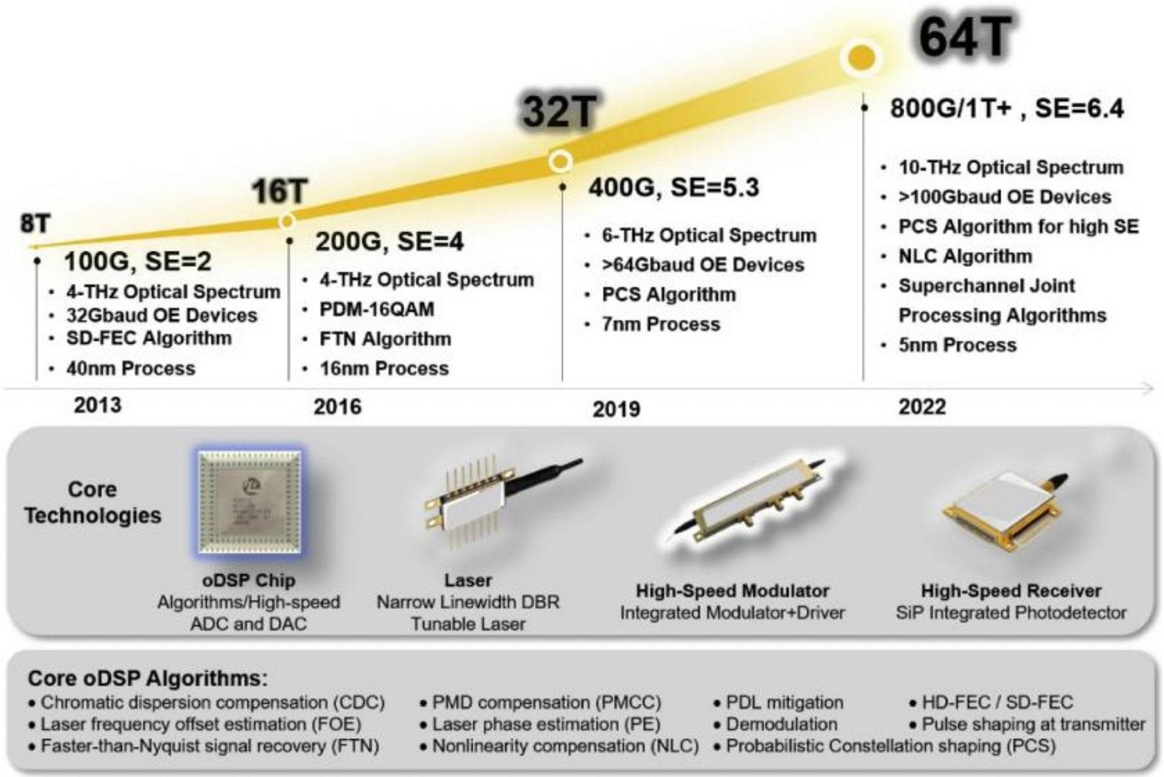


What's driving change today?

- **From scarcity to abundance**!
- For many years the demand for communications services outstripped available capacity
- We used price as distribution function to moderate demand to match available capacity
- But this is no longer the case – available capacity in the communications domain far outpaces demand

Abundant Capacity

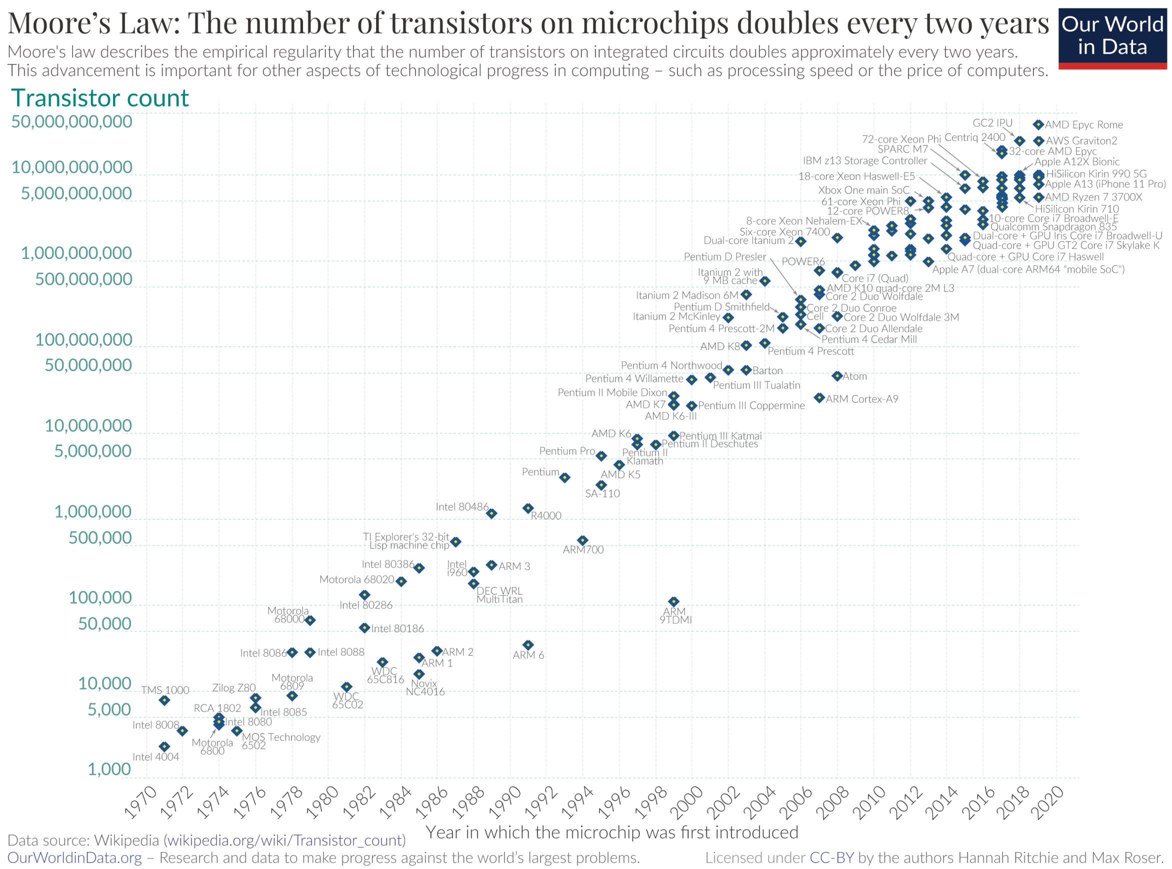
Fibre cables continue to deliver massive capacity increases within relatively constant unit cost of deployment



https://www.ncbi.nlm.nih.gov/

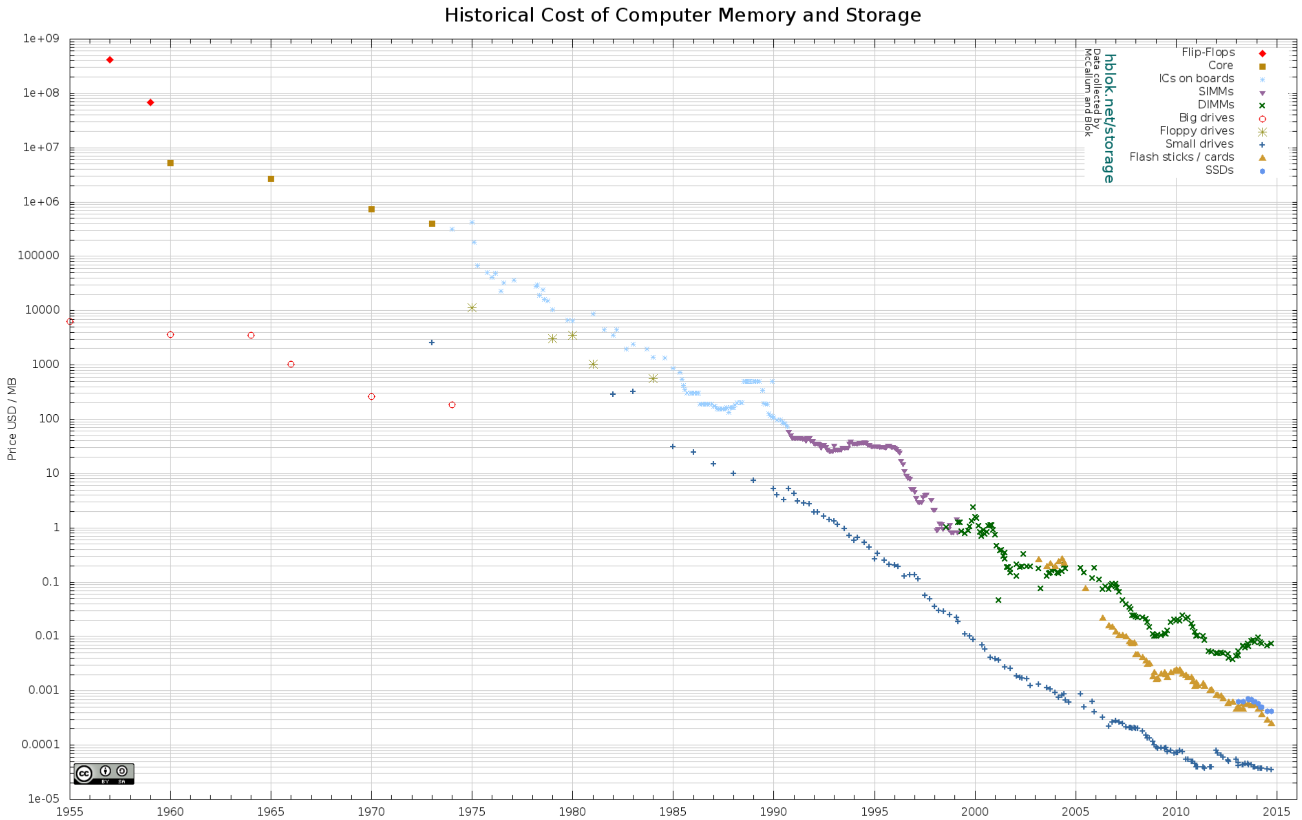
(That 2022 number is probably low – at the end of 2022 we can pull 2.2T per lambda with a 190Gbd signal rate, giving a fibre capacity of 105T)

Abundant Compute Power



By Max Roser, Hannah Ritchie - https://ourworldindata.org/uploads/2020/11/Transistor-Count-over-time.png, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=98219918

Abundant Storage



http://aiimpacts.org/wp-content/uploads/2015/07/storage_memory_prices_large-_hblok.net_.png

How can we use this abundance?

- By changing the communications provisioning model from ***on demand*** to ***just in case***
- Instead of using the network to respond to users by delivering services *on demand* we've changed the service model to provision services close to the edge just in case the user requests the service
- With this change we've been able to eliminate the factors of *distance* from the network and most network transactions occur over shorter network spans
- What does a *shorter* network enable?

Bigger

- Increasing **transmission capacity** by using photonic amplifiers, wavelength multiplexing and phase/amplitude/polarisation modulation for fibre cables
- Serving content and service transactions by distributing the load across many individual platforms through **server and content aggregation**
- The rise of high-capacity mobile edge networks and mobile platforms add massive volumes to content delivery

- To manage this massive load shift we've stopped pushing content and transactions across the network and instead **we serve from the edge**

Faster

- Reduce latency - stop pushing content and transactions across the network and instead **serve from the edge**
- The rise of CDNs serve (almost) all Internet content and services from massively scaled distributed delivery systems.

- The “Packet Miles” to deliver content to users has shrunk - that’s faster!
- The development of high frequency cellular data systems (4G/5G) has resulted in a highly capable last mile access network with Gigabit capacity
- Applications are being re-engineered to meet faster response criteria

- Compressed interactions across shorter distances using higher capacity circuitry results in a much faster Internet

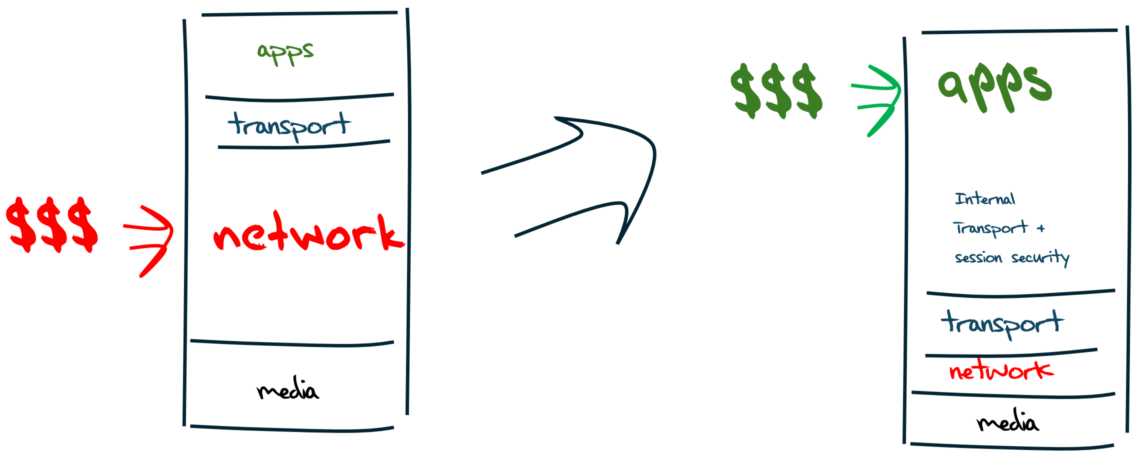
Better

- If “better” means “more trustworthy” and “more privacy” then we are making progress at last!
  - Encryption is close to ubiquitous in the world of web services
  - TLS 1.3 is moving to seal up the last open TLS porthole, the SNI field
  - QUIC is sealing up the transport controls from the networks
  - Oblivious DNS and Oblivious HTTP is moving to isolate knowledge of the querier from the name being queried
  - The content, application, and platform sectors have all taken the privacy agenda up with enthusiasm, to the extent that whether networks are trustable or not doesn’t matter any more – **all network infrastructure is uniformly treated as untrustable!**

Cheaper

- We are living in a world of abundant comms and computing capacity
- And working in an industry when there are significant economies of scale
- And its being largely funded by capitalising a collective asset that is infeasible to capitalise individually – the advertisement market
- The result is that a former luxury service accessible to just a few has been transformed into an affordable mass-market commodity service available to all

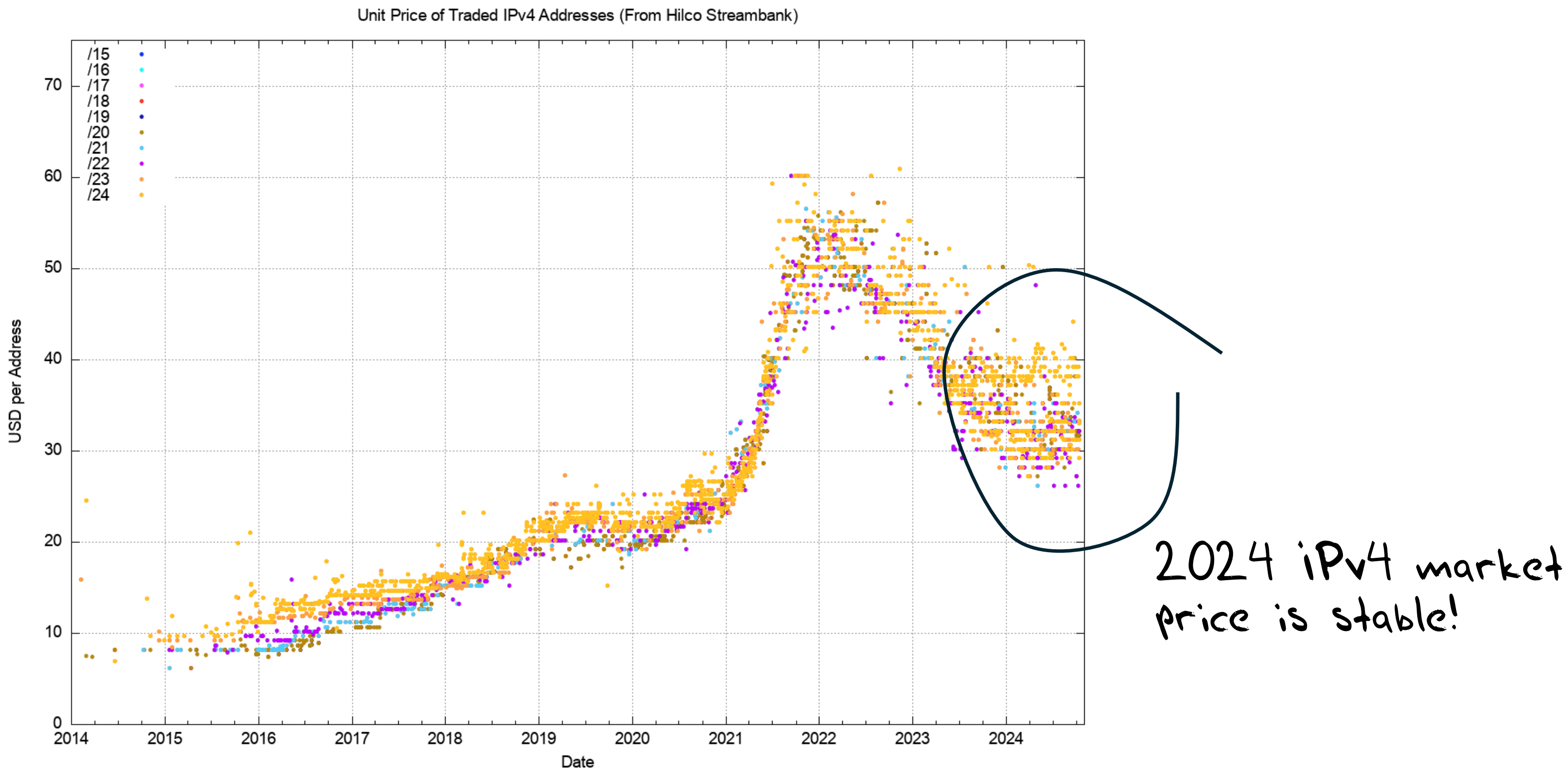
And in all this, the money moved up the stack!



So, who needs to pay?

- Networks need to make an investment to switch to a dual stack mode that includes IPv6
- But neither the user base not the content world really care
  - And they are certainly not going to pay a premium to the network operator for IPv6 support
- And in the application service world, IP addresses are not the critical resource

IPv4 Scarcity?



A Network of Names

- Today's public Internet is largely a service delivery network using CDNs to pull content and service as close to the user as possible
- The multiplexing of multiple services onto underlying service platforms is an application-level function tied largely to TLS and service selection using SNI
- The DNS is now used to perform "closest match" service platform selection, supplanting the role of routing
  - Most large CDNs run a BGP routing table with an average AS Path Length that is intended to converge to 1!

A new Internet Architecture

- We've moved from end-to-end peer networks to client/server asymmetric networks
- We've replaced single platform servers-plus-network to replicated servers-minus-network with CDNs
- Clients aren't identified with a unique public IP address – clients are inside NATs are uniquely identified only in a local context
- Individual services aren't identified with a unique public IP address – services are identified in the DNS

A new Internet Architecture

- We've moved from end-to-end peer networks [...] nt/server asymmetric networks
- We've replac[...] to replicated ser[...]
- Clie[...] a unique public IP address – clients are i[...] are uniquely identified only in a local context
- Individual services aren't identified with a unique public IP address – services are identified in the DNS

We've moved from address-based networks to name-based services

What am I saying?

- We’ve been able to take a 1980’s address-based architecture and scale it more than a billion-fold by altering the core reliance on distinguisher tokens from addresses to names
  - There was no real lasting benefit in trying to leap across to just another 1980’s address-based architecture (with only a few annoyingly stupid differences, apart from longer addresses!)

Today's Internet:

- Names Matter
- The DNS Matters

Today's Internet:

- Names Matter
- The DNS Matters

- Addresses not so much
- Address-based Routing not so much

Longer Term Trends?

Pushing EVERYTHING out of the network and over to applications

- Transmission infrastructure is becoming an abundant commodity
  - Network 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 bringing services towards consumers and using the content frameworks to replicate servers and services
- 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 our 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 public Internet"?
    - A common shared transmission fabric, a common suite of protocols and a common protocol address pool?

  or

    - A disparate collection of services that share common referential mechanisms using a common name space?

Thank You!