The state of IPv6 (and IPv4)

Amsterdam, 26 february 2014

Iljitsch van Beijnum

http://www.bgpexpert.com/presentations/

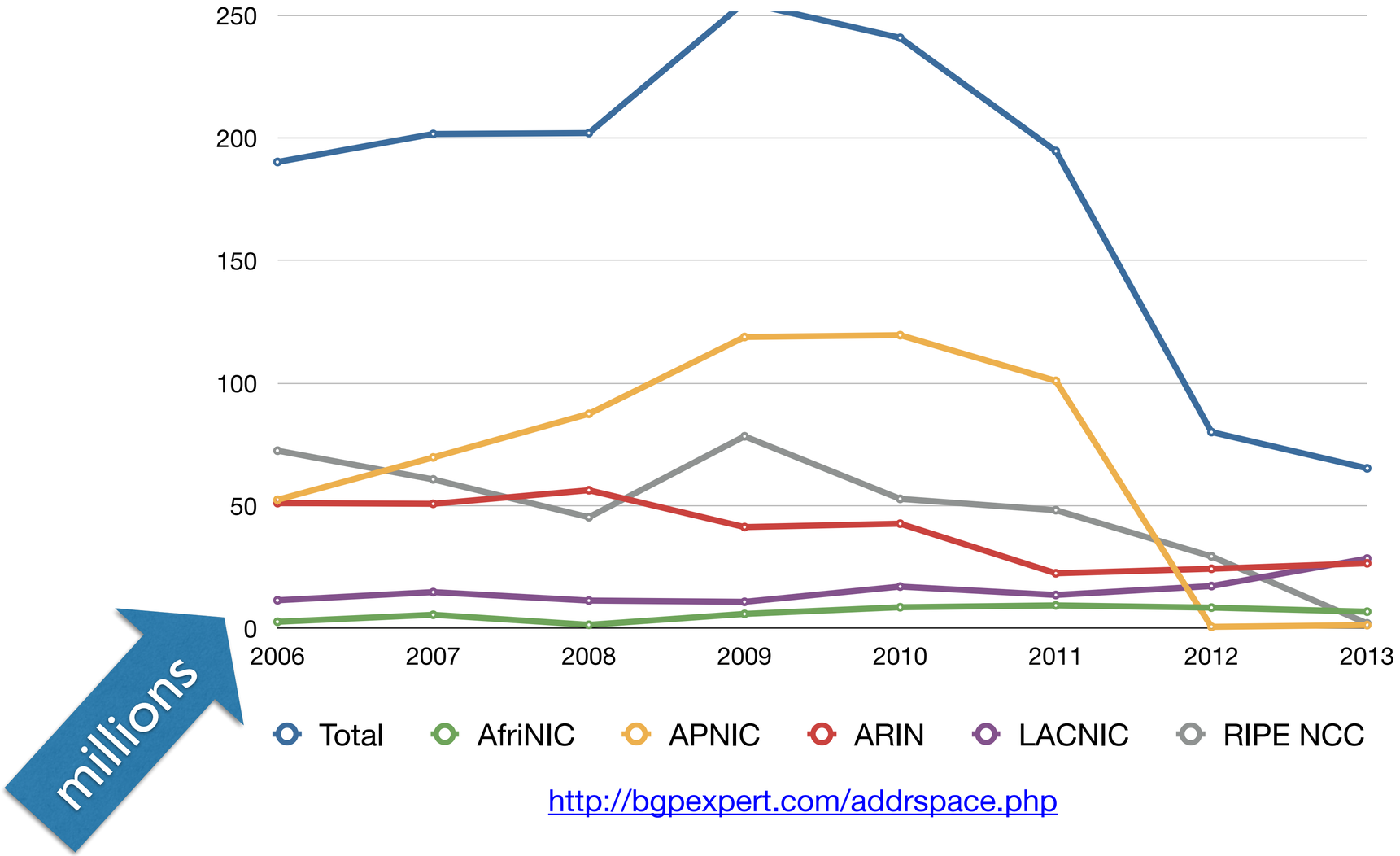
Today's topics

- IPv4 is running out
- Address configuration
- Issues with choices
- How do we get there
- The economics
- Packet sizes

Status IPv4

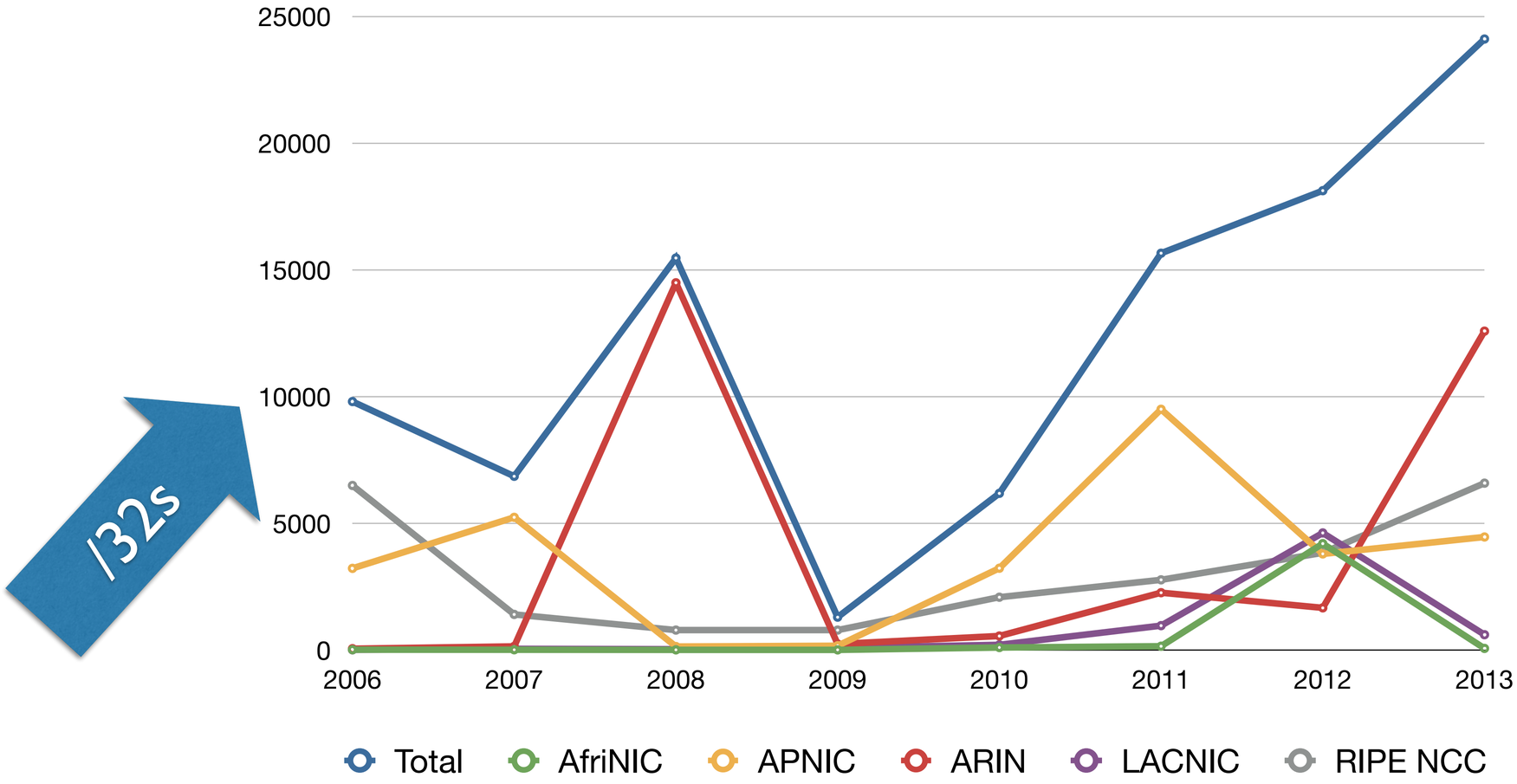

Sep 2012: final /8
Feb 2011
AfriNIC
APNIC
ARIN
LACNIC
RIPE NCC
Apr 2011: final /8

IPv4 addresses per year



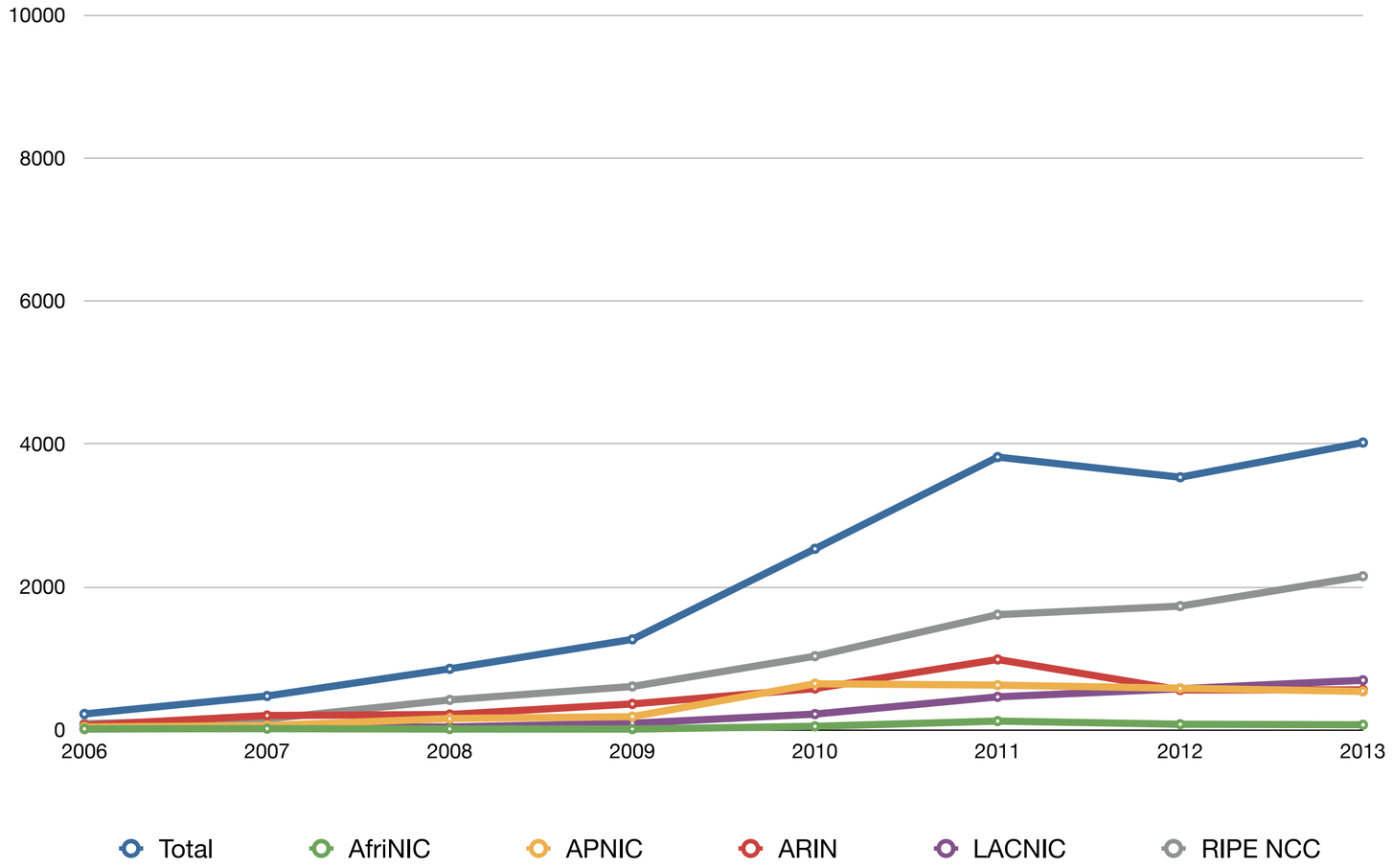
http://bgpexpert.com/addrspace.php

IPv6 addresses/yr



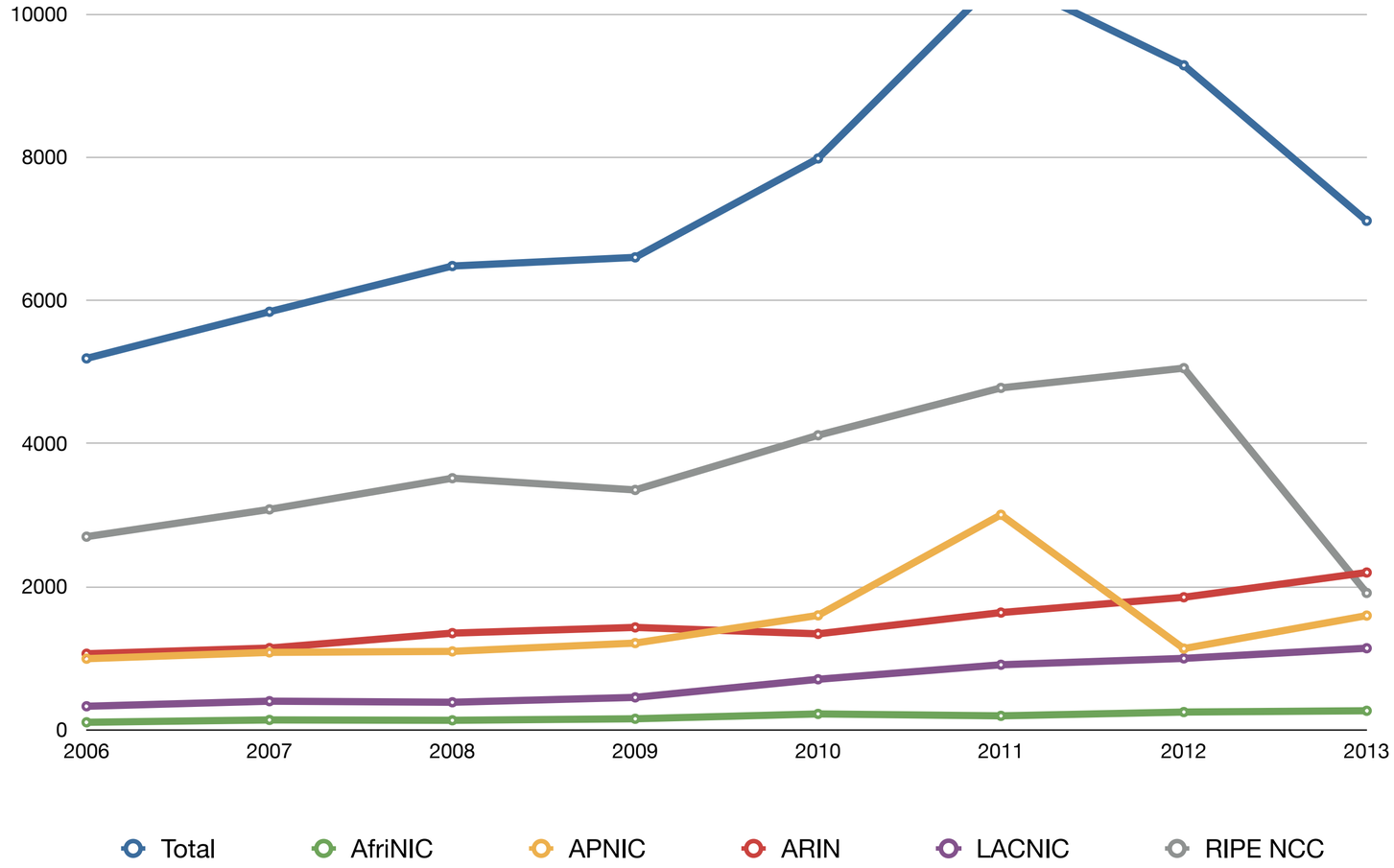
http://www.bgpexpert.com/addrspace-ipv6.php

IPv6 address blocks/yr



http://www.bgpexpert.com/addrspace-ipv6.php

IPv4 address blocks/yr



http://www.bgpexpert.com/addrspace-ipv4.php

Trading!

Microsoft | Cloud Services

We're all in.

- IPv4 address trading (buying/selling) is fairly common now
- Especially in North America
  - even though ARIN still has IPv4!
- Going rate: $/€ 5 - 10 per address
- Prominent buyers: Amazon, Microsoft
  - what do they have in common?

Where do you get
your address?

1980s: multiprotocol!

	Addr bits	Network	Host	Configuration
IPX	80	32	48	broadcast + MAC
AppleTalk	24	16	8	broadcast + random
CLNP	max 160	variable*	variable*	broadcast + MAC
IP < 1993	32	8 | 16 | 24	24 | 16 | 8	manual
IP > 1993	32	variable	variable	DHCP

IPv6

- Includes *all* address configuration methods discussed so far:
  - manual configuration
  - router broadcast + MAC address
  - router broadcast + random number
  - (router broadcast + crypto hash)
  - DHCPv6

Stateless autoconfig

- Routers send out "router advertisements"
- RAs contain one or more /64 prefixes
- Hosts add 64 bits derived from MAC address, random number or crypto hash
- Perform duplicate address detection (DAD) just in case
- Keep address until timer expires

Router advertisements

- RAs are *multicast*, not broadcast
  - so only IPv6 hosts "see" them
- Routers send RAs periodically
- Or immediately after receiving a router solicitation
  - router solicitations are sent by hosts to the all-routers multicast address

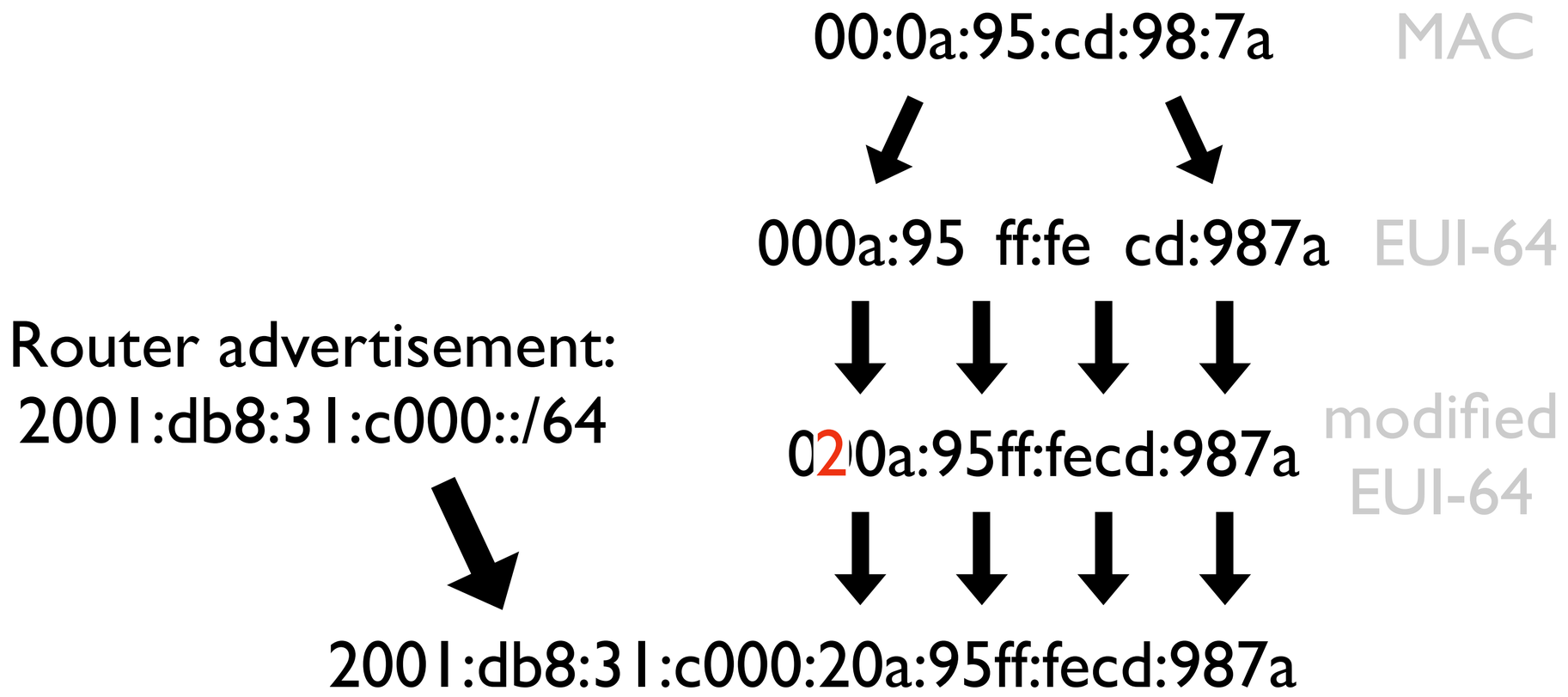
Prefix option flags

- L: on-link flag: this prefix should be considered locally reachable
- A: autonomous address-configuration flag: create an address using this prefix (if /64)
- L=1, A=1: normal stateless autoconfig
- L=0, A=1: autoconfig but *not* on-link
- L=1, A=0: no autoconfig, but on-link
- L=0, A=0: ?

On-link

- With IPv4, every address has a (sub-)netmask
  - all nodes with addresses matching the netmask are directly connected / on-link
- With IPv6, address *may or may not* have a prefix length that indicates what's on-link
  - like CNLP!
- Reach off-link addresses through a router

IPv6 address creation



Address Privacy

- Ugh, when you move around people can recognize your MAC address!
- RFC 4941 (was 3041): temporary addresses
  - use random number to generate address
  - generate new one every 24 hours

    or after disconnect/reconnect
  - default for outgoing sessions in Windows Vista/7 and MacOS 10.7

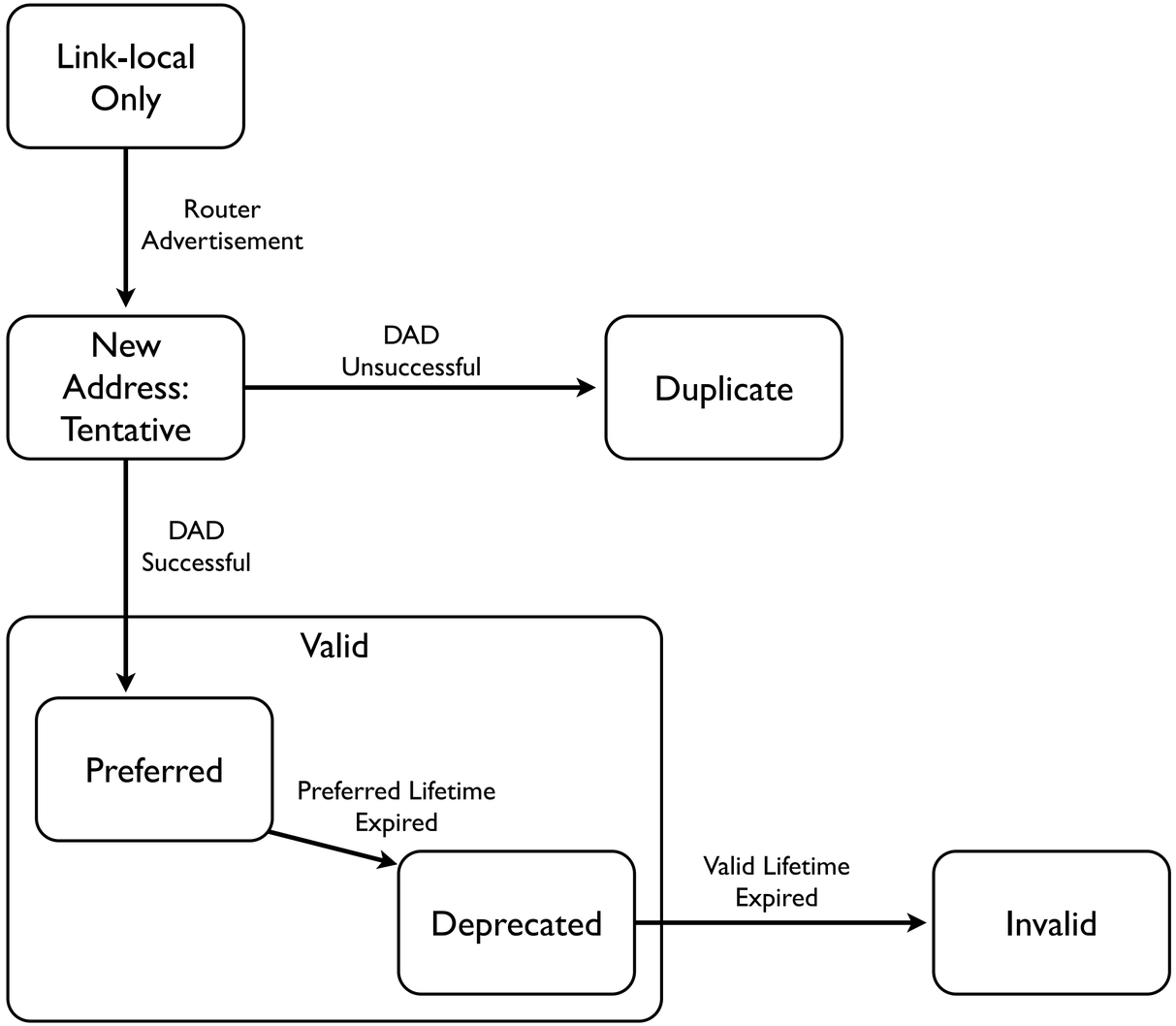
Timers

- RA timer:
  - how long router may be default gateway
- Prefix preferred lifetime:
  - how long address is "preferred"
- Prefix valid lifetime:
  - how long address can be used (at all)
- All count down unless restored by new RA

Duplicate address detection

- Before a node may use an address, see if nobody else has it
- Address is "tentative"
- Send out neighbor solicitations for tentative address
  - source address: the unspecified address ::
- If no answer, use it
- If answers, don't use it (and...?)

Lifecycle of addresses



Choice is bad

RA flags

- "Managed config" (M bit)
  - "stateful address configuration" (= DHCPv6) is used on this subnet
- "Other stateful config" (O bit)
  - other configuration information (such as DNS addresses) is available through stateful configuration mechanism

DHCPv6

- Complete reinvention of DHCP for IPv6
- Completely incompatible with DHCP
- Doesn't provide router address
- Doesn't provide subnet mask/length
- No MAC address or client identifier, but "DUID" = DHCPv6 Unique IDentifier

DHCPv6 (2)

- Two modes of operation:
  - stateful (M=1): for address configuration etc
  - stateless (O=1): for DNS configuration etc
- In addition to address configuration, also **prefix** delegation

RA flags and DHCPv6

M	O	Prfx	A	Result
0	0	-		default gw but no address
0	0	yes	0	default gw but no address
0	0	yes	1	working IPv6 but no DNS
0	1	-		default gw + DNS but no address
0	1	yes	0	default gw + DNS but no address
0	1	yes	1	working IPv6
1	0	-		address+DNS, no subnet length (may not work)
1	0	yes	0	working IPv6
1	0	yes	1	working IPv6, 2 addresses
1	1	-		address+DNS, no subnet length (may not work)
1	1	yes	0	working IPv6
1	1	yes	1	working IPv6, 2 addresses

(Dis)advantage

- As a philosopher once said: "every disadvantage has its advantage"
- So if you have both IPv4 and IPv6, and one doesn't work, you can use the other!
- But only if you can hop from the broken protocol to the working one quickly
- So: "happy eyeballs"

Happy eyeballs

- Problem: TCP doesn't know when to quit
  - Windows: 19 seconds
  - Mac: 75 seconds
  - Linux: 189 seconds
- So simple "try v6, fail, try v4" is too slow
- This was also common in the age of 6to4 tunneling... (Teredo is better/worse)

Bemused Eyeballs: Tailoring Dual Stack Applications for a CGN Environment

Happy eyeballs (2)

- Mac/Safari: try v6, try v4, measure RTTs, keep using the fastest IP version, activate the other after about an RTT of waiting
- Chrome: AAAA and A queries, use what comes back first, switch over after 300 ms
- Firefox: v4 and v6 in parallel, use first, close second unused
- Windows: ???

But how do we get there?

NCP to IP/TCP

- In the 1970s, the ARPAnet had the Network Control Protocol (NCP)
  - one protocol to rule them all
  - monolithic protocol was becoming a problem
- So IP/TCP (now known as TCP/IP or simply IP) was developed, two protocols that work together
- They took 1982 to transition

1982

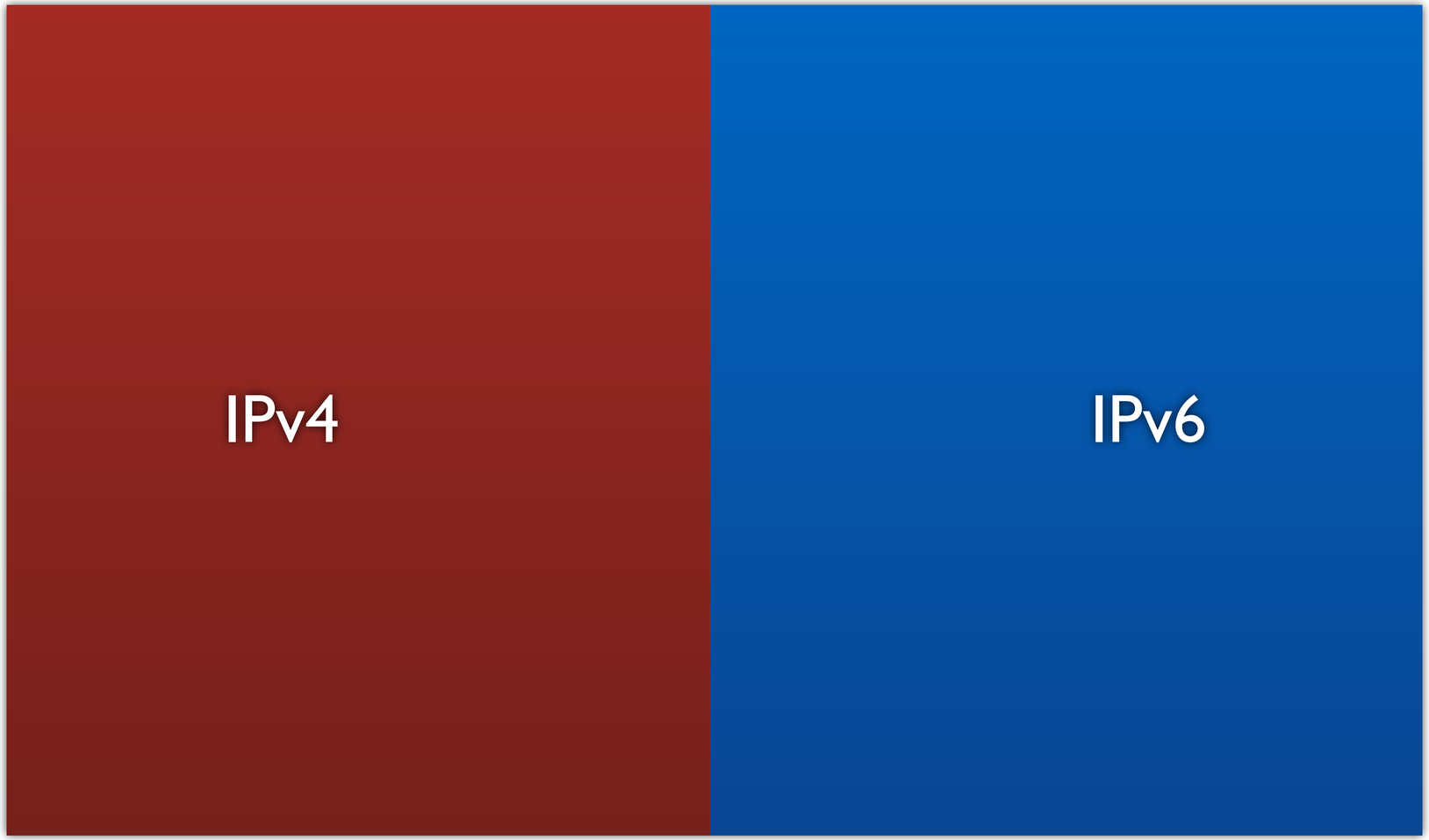
- So it took **ONE YEAR** to transition, even though:
  - there were only about 100 nodes in the network
  - really only three applications:

    FTP

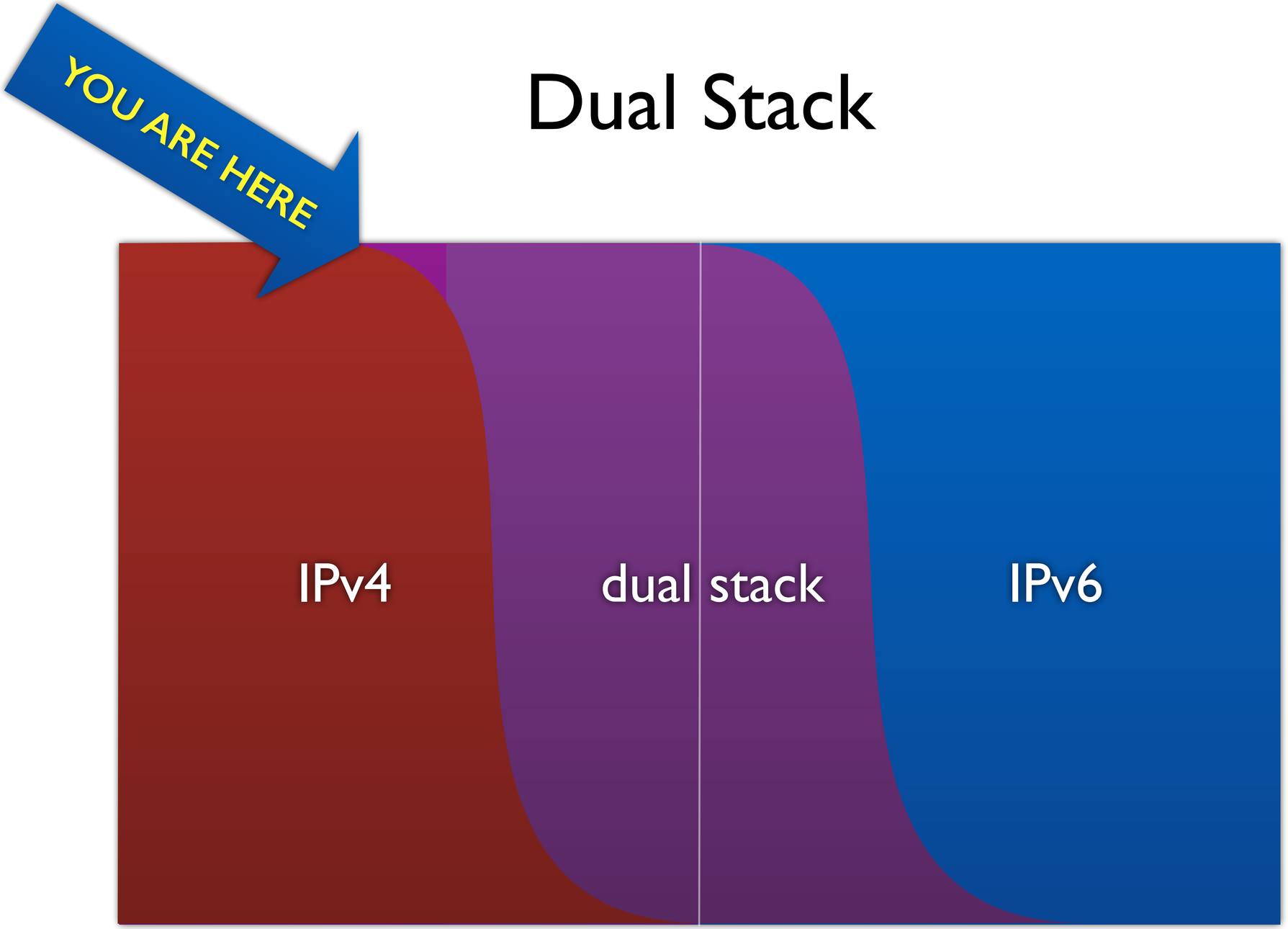
    telnet

    mail

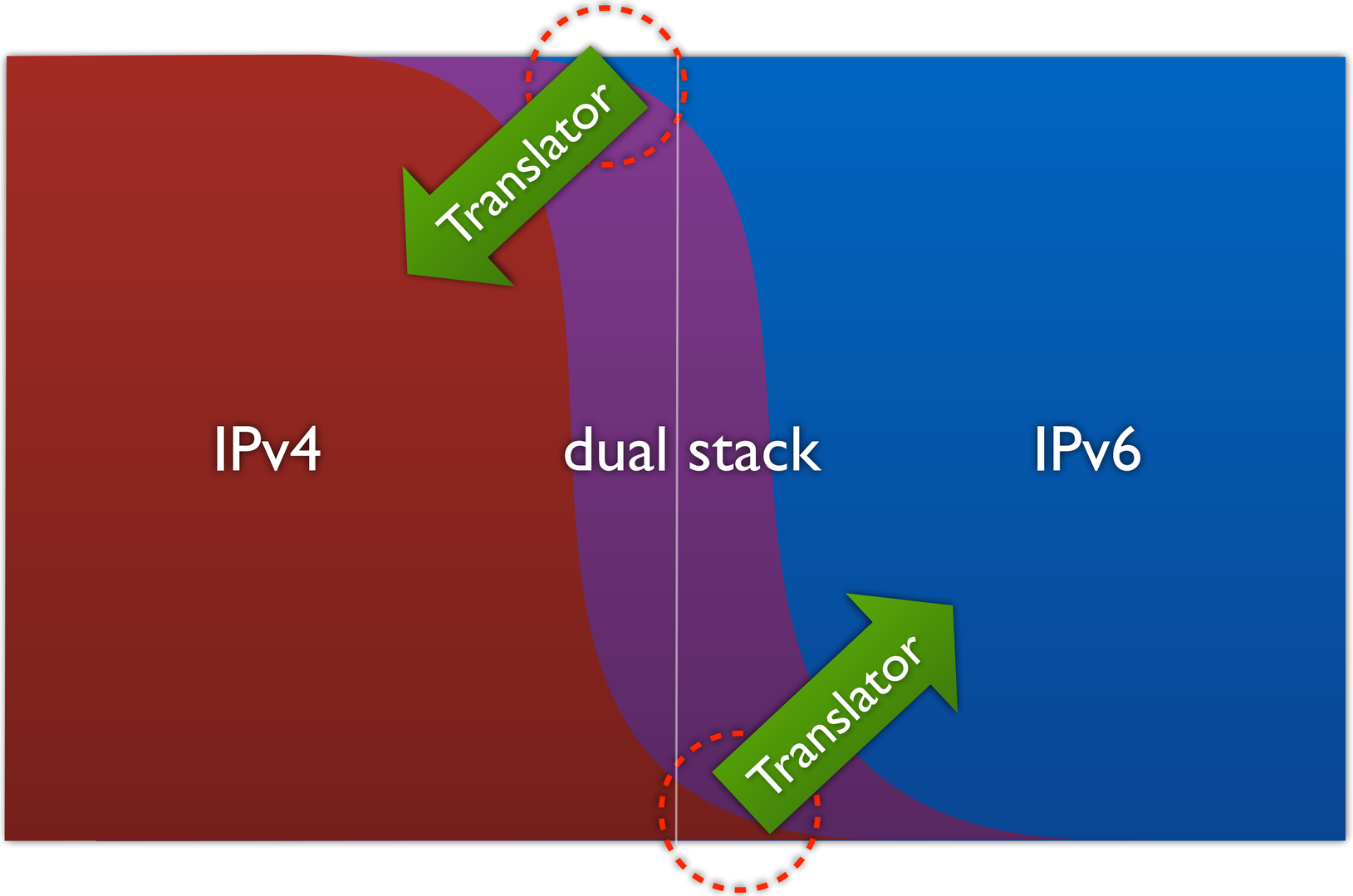
"Flag Day"



Dual Stack



Reality?

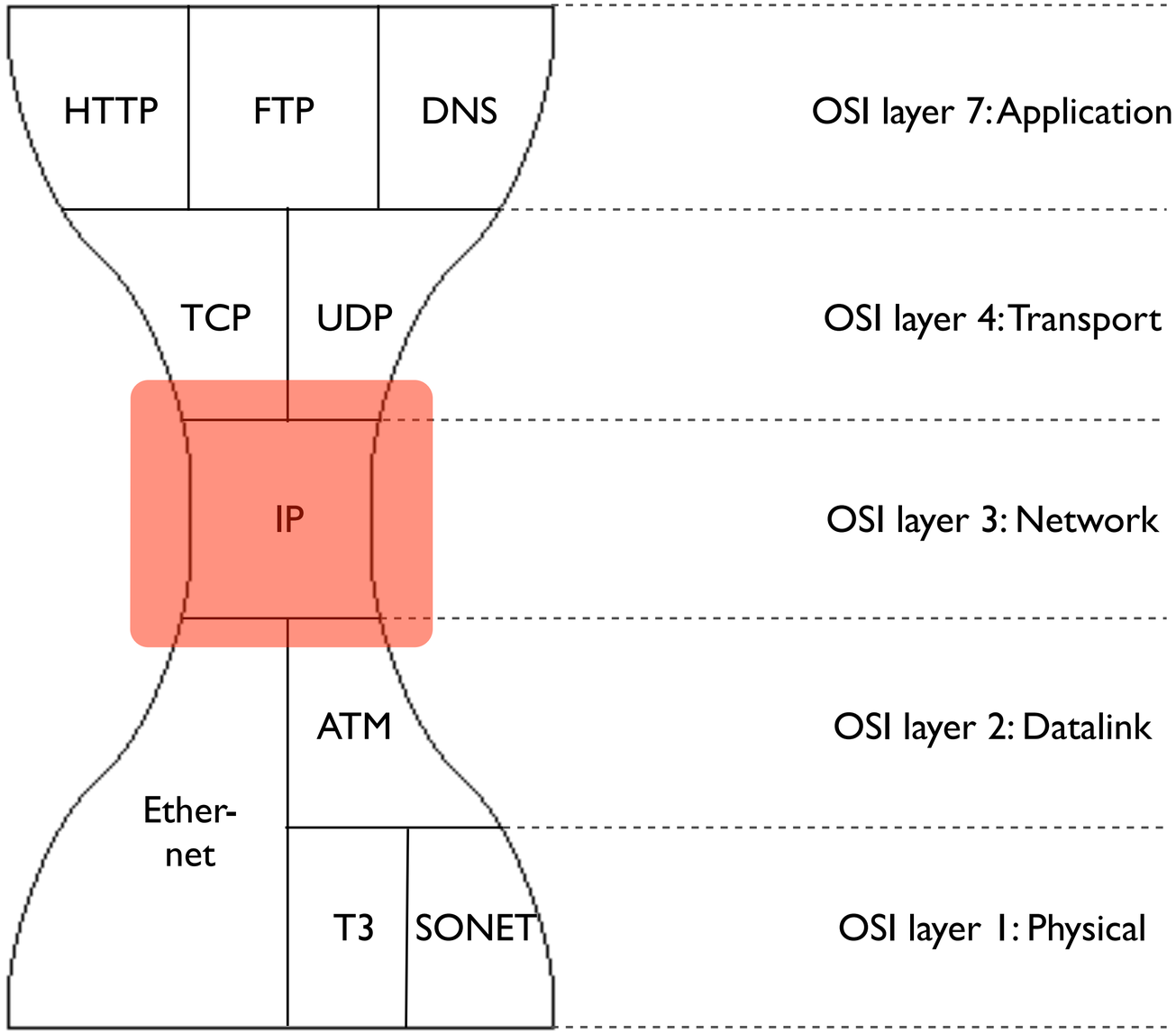


Why are layer 3 transitions so hard?

- I upgraded from 10 to 100 Mbps Ethernet to Gigabit Ethernet without trouble
- And from 11 to ~~54~~ ~~300~~ 1300 Mbps Wi-Fi
- DNS can switch from UDP to TCP on the fly
- http://twitter.com/ and https://twitter.com/ work the same

It's different

- Ethernet or Wi-Fi are only in your house
  - the rest of the network doesn't care
- Applications are between the ends
  - the rest of the network doesn't/shouldn't care
- Network layer = IP address are everywhere
  - *everything* has to care

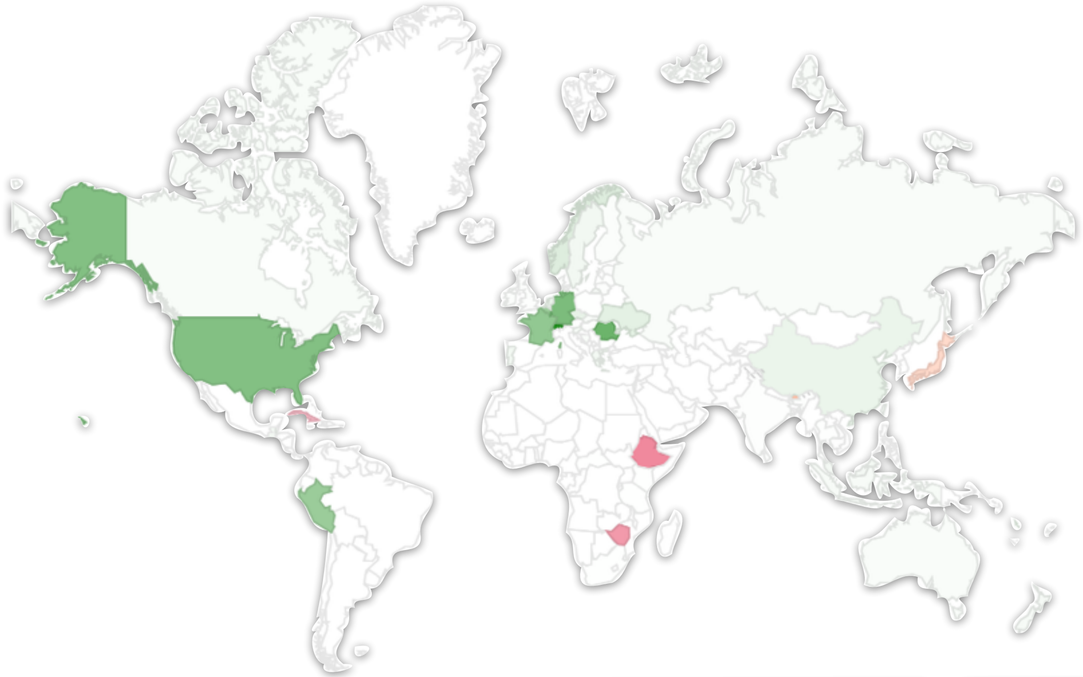

HTTP
FTP
DNS
OSI layer 7: Application
TCP
UDP
OSI layer 4: Transport
IP
OSI layer 3: Network
ATM
OSI layer 2: Datalink
Ether-
net
T3
SONET
OSI layer 1: Physical

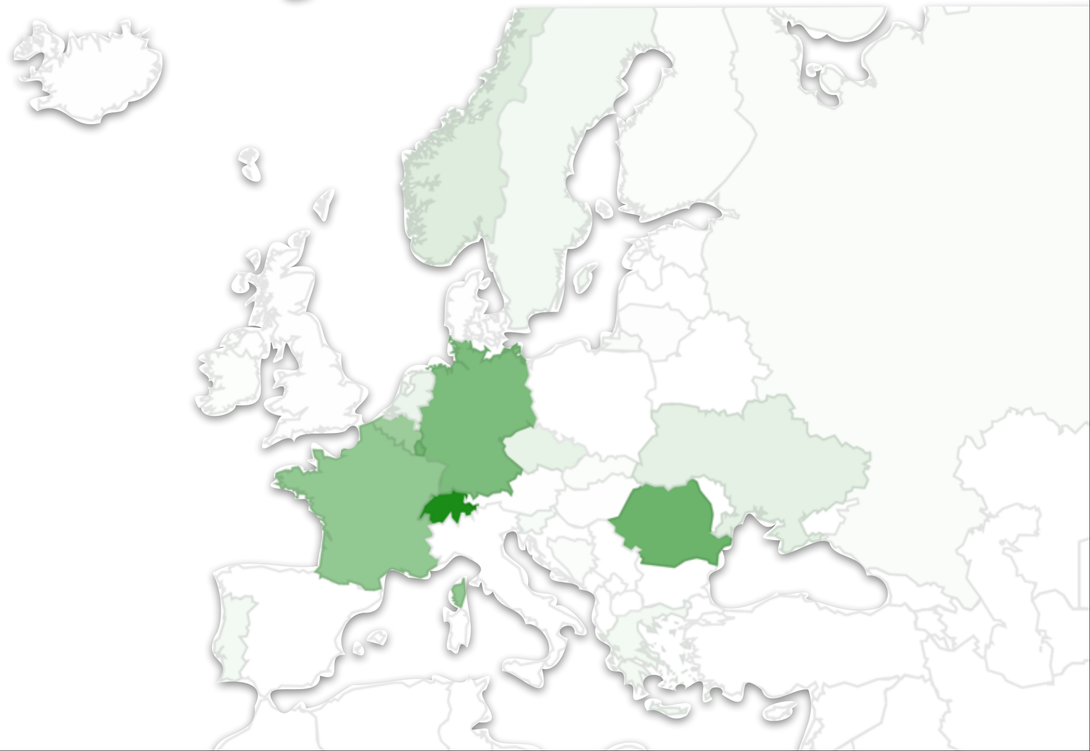
When?

- Some people happy to go to IPv6 now/soon
- Some people very much against it
- Most users: huh?
  - depend on vendors / service providers
- Vendors in reasonable shape
- Service providers: stick with v4 to the end

Current state

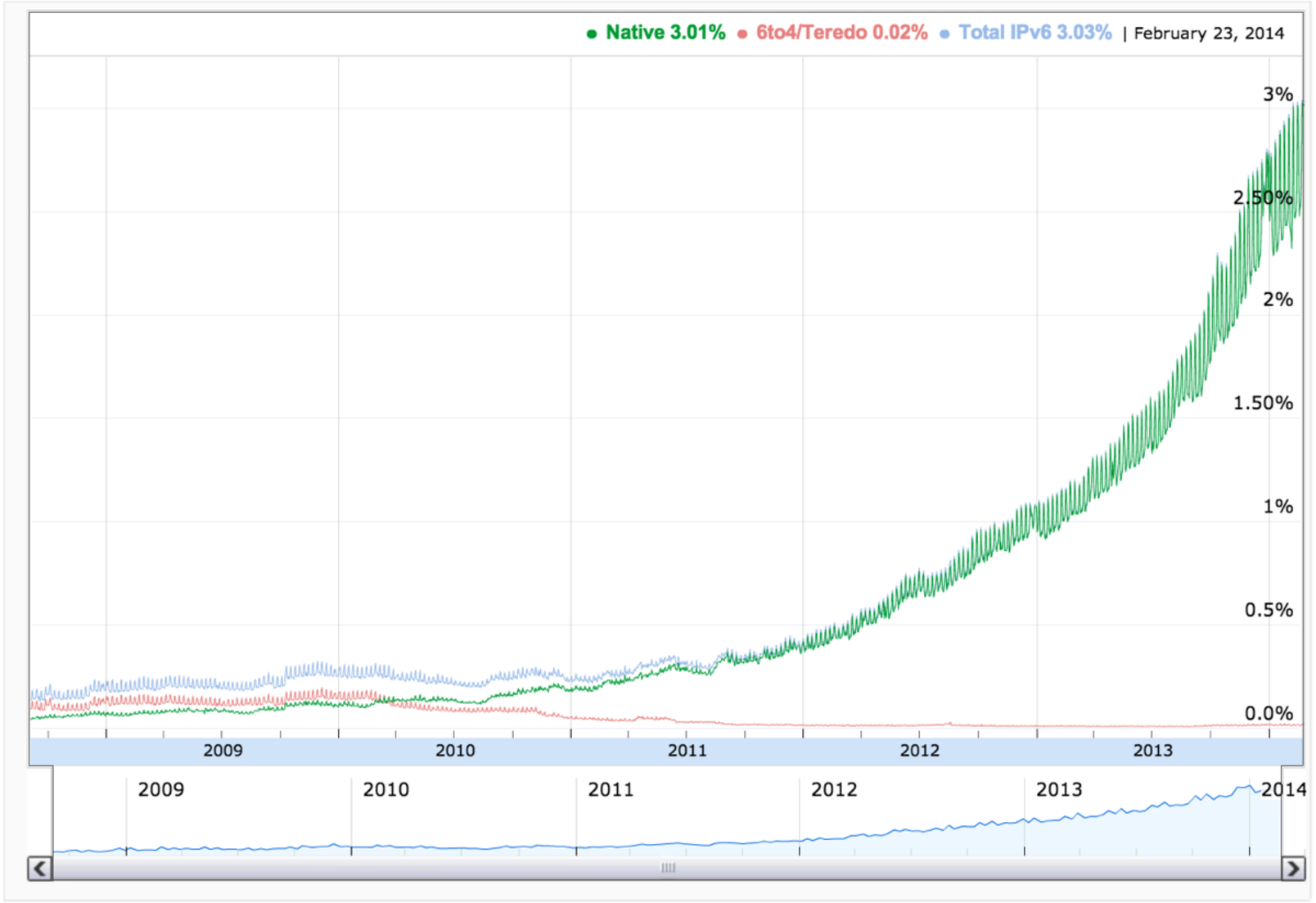
- (Well, jan 1st)
- Web: IPv6 stagnating
- End-users: IPv6 emerging
  - Google sees 3%
  - (one little country is leading the resistance...)



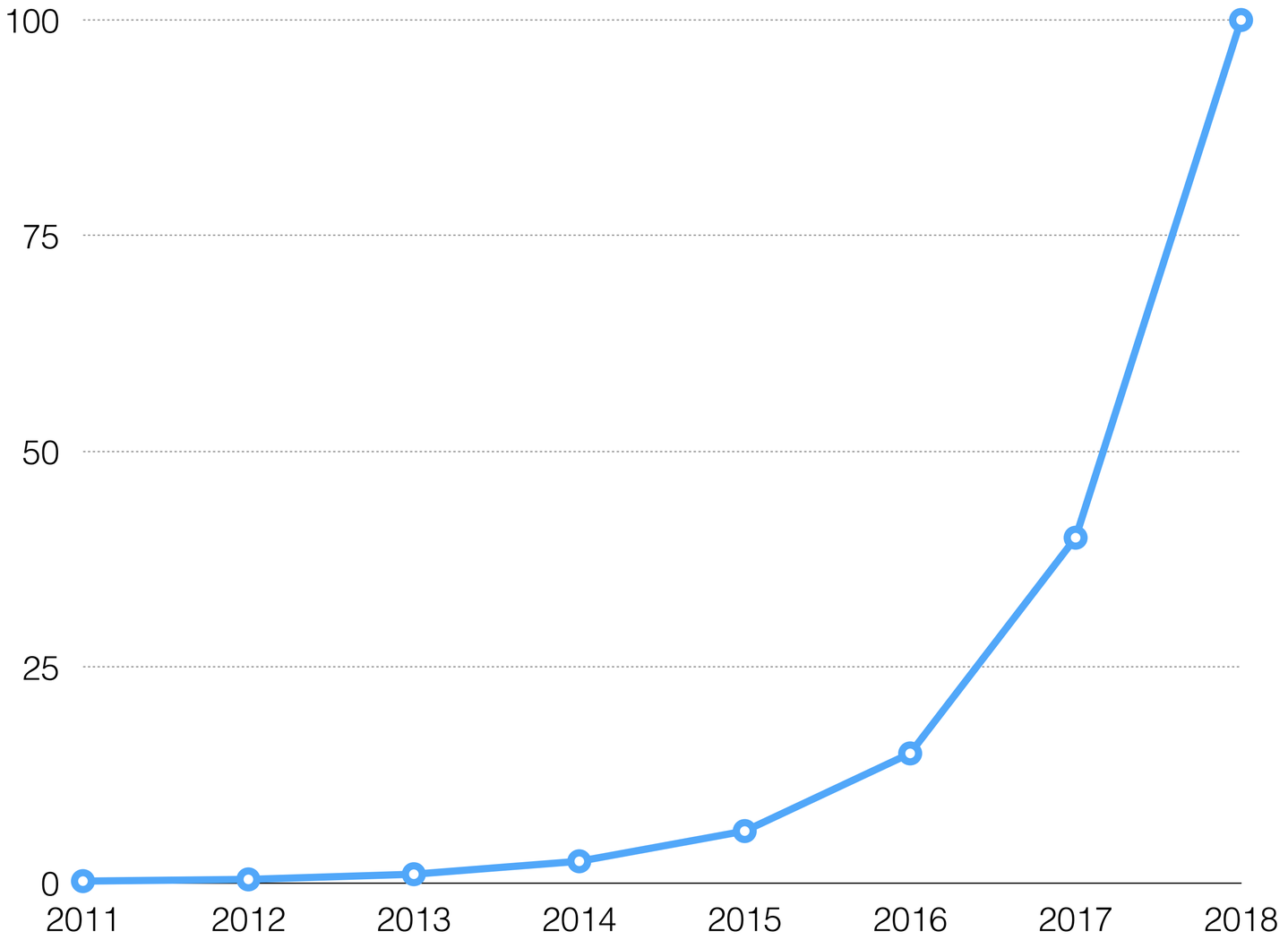


http://www.google.com/intl/en/ipv6/statistics.html

Google: 3%


Native 3.01%
6to4/Teredo 0.02%
Total IPv6 3.03%
February 23, 2014
3%
2.50%
2%
1.50%
1%
0.5%
0.0%
2009
2010
2011
2012
2013
2009
2010
2011
2012
2013
2014

So?



The End of IPv4

- Small address users: pretty much never
- Large address users: end around 2012, then:
  - Existing large users: fairly light NAT
  - New large users: very heavy NAT
- Heavy Network Address Translation / multiple NATs bad for peer-to-peer

NAT Crunch

- VoIP, BitTorrent, personal servers etc. harder and harder
- IPv6 to bypass NAT
- IPv6 will be promoted by service providers with few IPv4 addresses to be competitive
- People with adequate IPv4 will add IPv6 to talk to others behind NAT

ISP NAT

- No more new IPv4 addresses:
  - customers need to share an address
  - ISP runs NAT
- NAT from IPv4 to IPv4 to IPv4 (NAT444)
- Carrier Grade NAT (CGN)
- Large Scale NAT (LSN)

ISP NAT (2)

- (Currently) no protocols to poke holes in the NAT
  - (future: PCP?)
- Who gets port 80 or port 5060?
- Result: more applications break
- Also can't do 6to4 tunneling

NAT64

- Lets IPv6 clients talk to IPv4 servers
- Client looks up AAAA record
- DNS64 returns fake AAAA record: /96 prefix + A record
- /96 is routed to NAT64
- NAT64 translates between IPv4 and IPv6
- IPv6 traffic bypasses NAT64 translator

NAT64 vs NAT444

	NAT64	NAT444
Translated traffic	IPv4 destinations	all traffic
IPv6	supported	orthogonal and breaks most tunnels
IPv4-only applications	unsupported	supported
DNSSEC	mostly supported	supported
IPv4 literals	unsupported	supported
Network topology	(can be) simple	complex

What ratio?

- 1 IPv4 address / 10 users: not so bad!
- 1 / 100: ??
- 1 / 1000: ?
- 1 IPv4 address / 10000 users: trouble!
- (65000 TCP ports per IPv4 address)
- So still many IPv4 addresses required

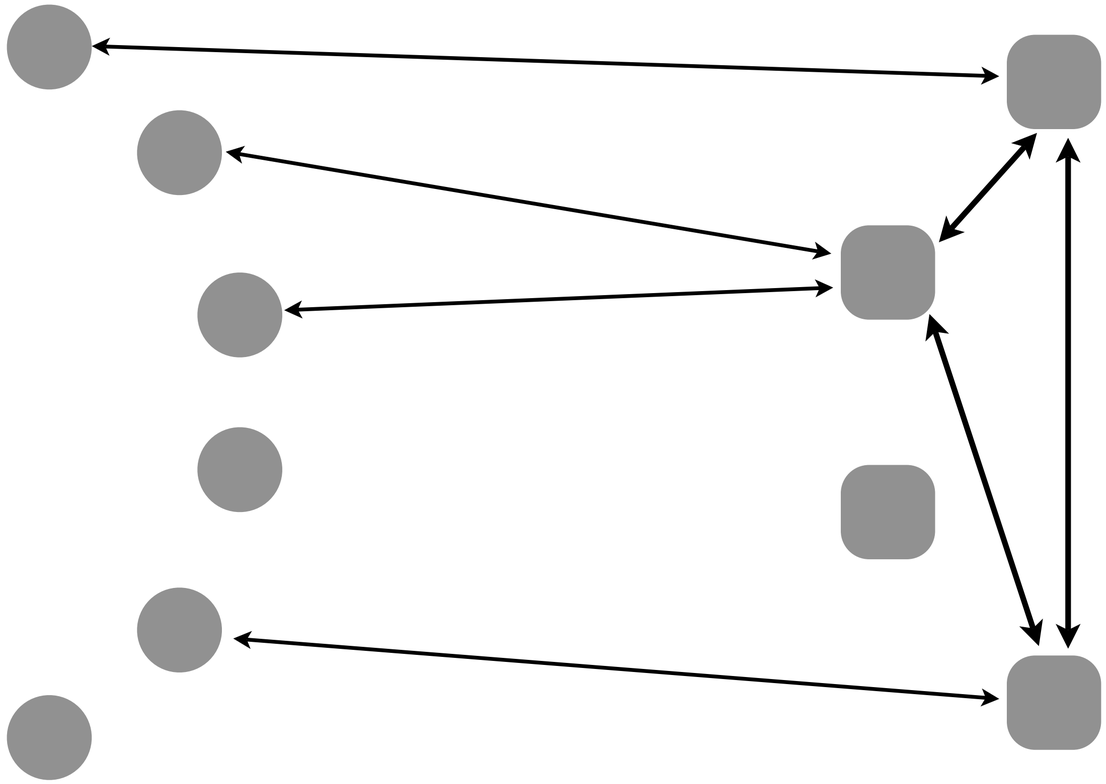
NAT46?

- NAT64: server's 32-bit IPv4 address can be encoded in the 128-bit IPv6 address that the client sees
- NAT46 with 128-bit address in 32-bit address: not so much
- Not entirely impossible, but very hard
- IPv4-only clients will be in trouble when IPv6-only servers start appearing

Not Uniform

- Different transition scenarios per:
  - application
  - user group
- Different applications/users communicate in different ways
- No requirement that the same IP version is used for all communication

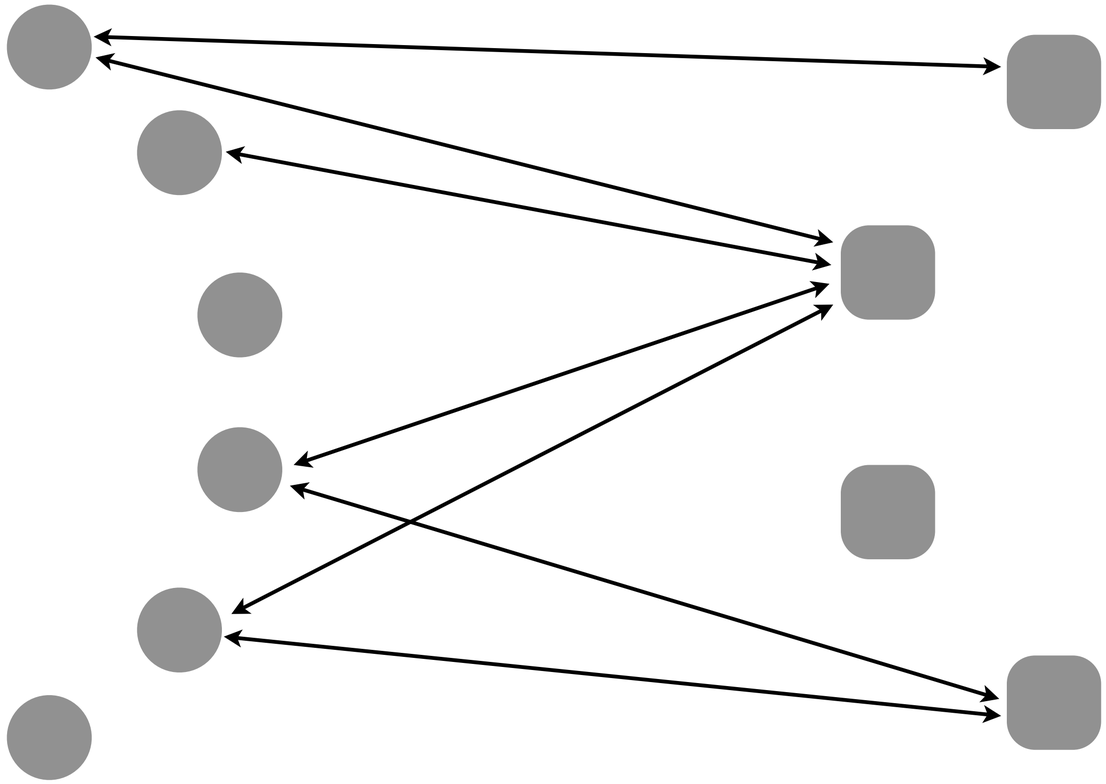
Email Model



Clients

Servers

WWW Model



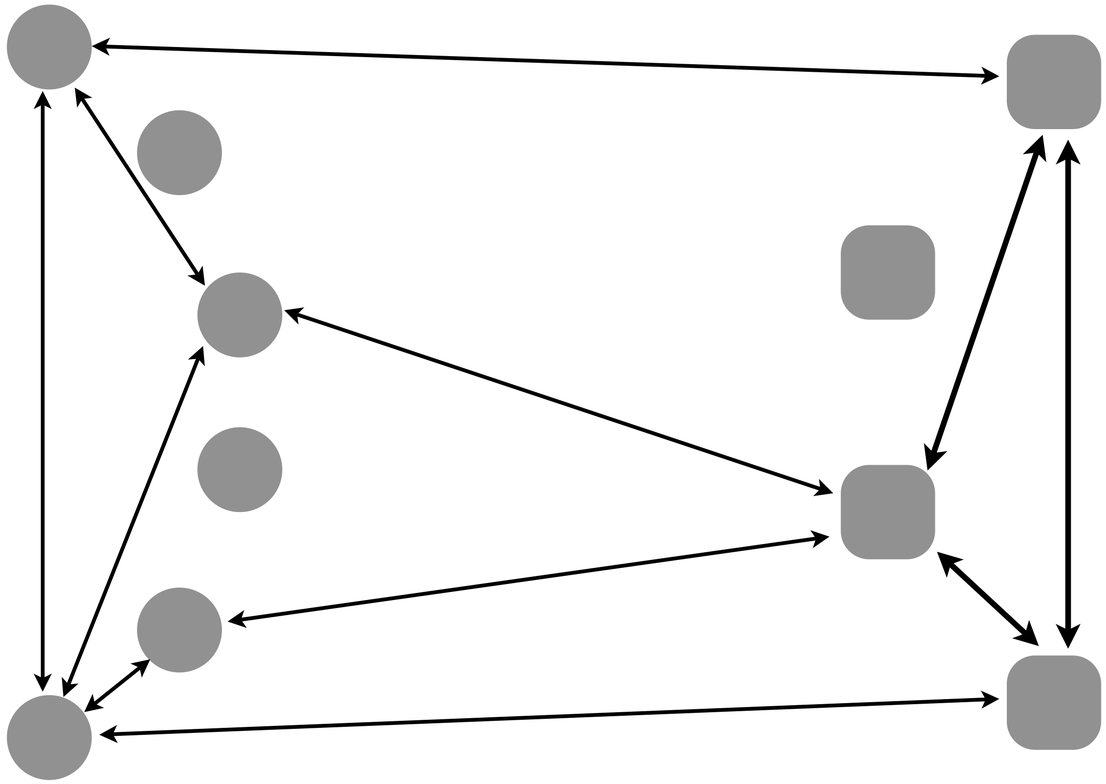
Clients

Servers

Client/Server Apps

- Email
  - clients talk to one server
  - servers communicate between them
- World Wide Web
  - clients talk to all servers
  - servers don't communicate with servers

P2P Model



Peers

Servers

Peer to Peer Apps

- P2P type BitTorrent (file distribution):
  - no server-to-server and only subset clients needs to be reachable
- P2P type VoIP (one-to-one/one-to-few):
  - potentially all servers with all servers, all clients with all clients

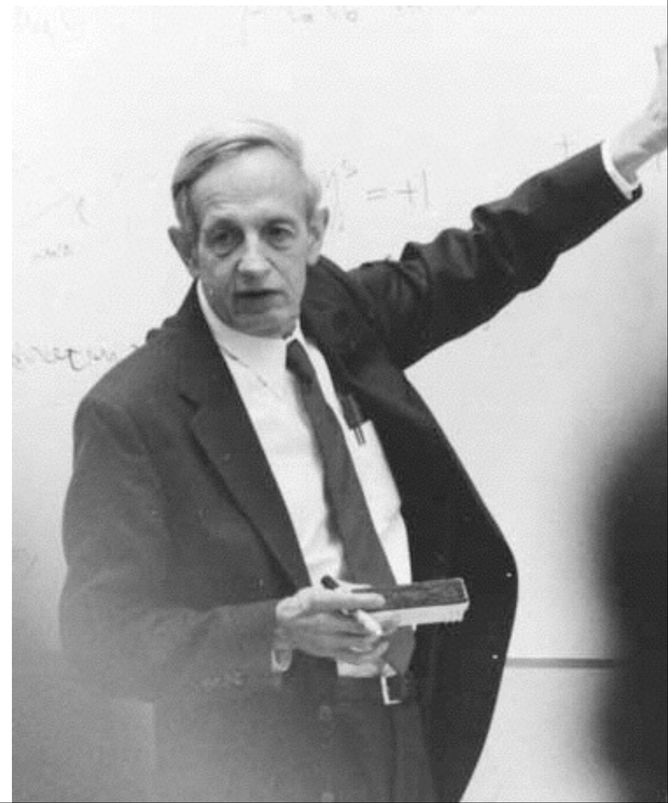
Client IPv6-only?

- Email: only own server needs to be DS
- BitTorrent: server and some clients DS
- WWW: all servers must be dual stack
- VoIP: all servers and clients dual stack
- NAT64 or proxy (incl. VoIP gateway) turns everything into email model
  - but no P2P from IPv4 to IPv6 clients

The economics

Nash equilibrium

- Advantages and costs of transition differ massively per organization, so:
  - some want to transition quickly
  - some not at all
- IPv6 only works if everyone adopts it...
- Nash equilibrium: nobody can unilaterally improve the situation



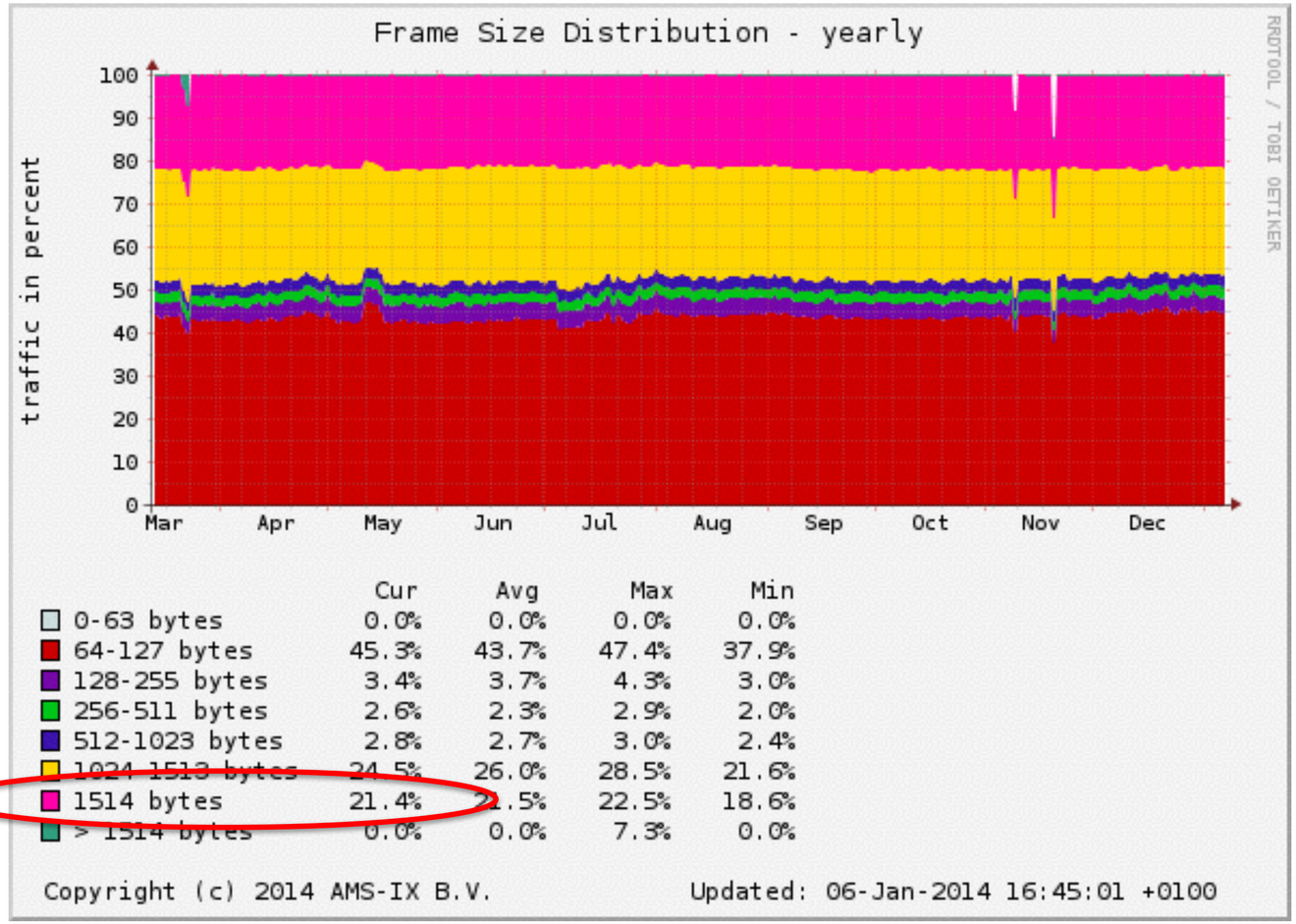
The way forward

- Patience:
  - IPv4 gets more expensive (no addresses...) and IPv6 gets cheaper
  - slowly, more organizations adopt IPv6
  - Metcalfe's law comes into play
- Even if you don't turn off IPv4 you may run IPv6-only on the go from time to time

Packet sizes

But...

- IPv6 or IPv4:
  - the packets are still way too small!



Why only 1500 bytes?

- The original Ethernet standard specifies an MTU of 1500 bytes
- MTU = Maximum Transfer Unit
  - the maximum size of an IP packet
  - (resulting Ethernet packet is 1514 / 1518 bytes)
- Or: ± 800 packets per second (PPS)

But that was 30 years ago!

~ 1980	10 Mbps	Ethernet	800 PPS
~ 1995	100 Mbps	Fast Ethernet	8000 PPS
~ 1998	1000 Mbps	Gigabit Ethernet	80000 PPS
~ 2002	10000 Mbps	10 Gigabit Ethernet	800000 PPS
~ 2010	100000 Mbps	100 Gigabit Ethernet	**8 MPPS**

Compatibility

- Fast Ethernet had to be interoperable with Ethernet = 1500 bytes
- Gigabit Ethernet had to be interoperable with Fast Ethernet = 1500 bytes
  - (even though nearly all GE hardware can handle "jumboframes")
- Same thing for 10 and100 Gigabit Ethernet

The problem

- Amount of work is about the same regardless of MTU
- So smaller packets = more CPU use
  - (or, with routers and switches: faster ASIC)
- So: lower performance and/or higher energy use!

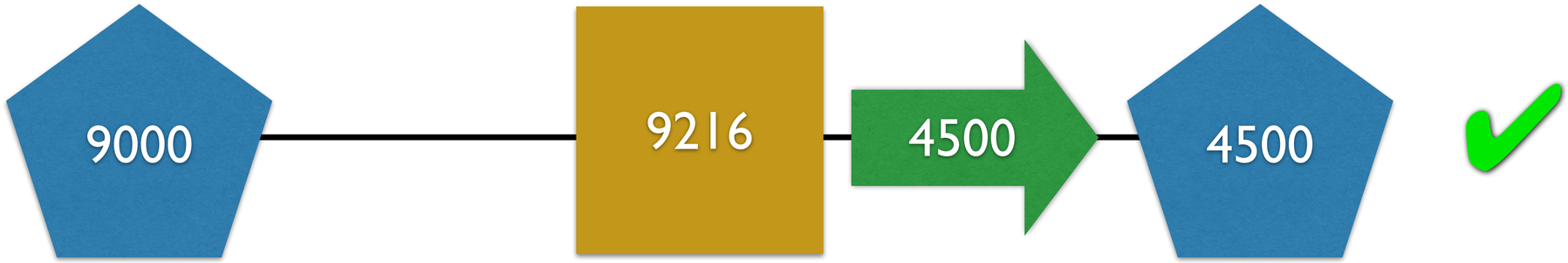
What do we do about it?

- Standardize new packet size?
  - will also be too small 10 years from now...
- Instead: flexibility!
  - everyone has their own MTU
  - tell your MTU to your neighbors
  - they will send you packets of the appropriate size

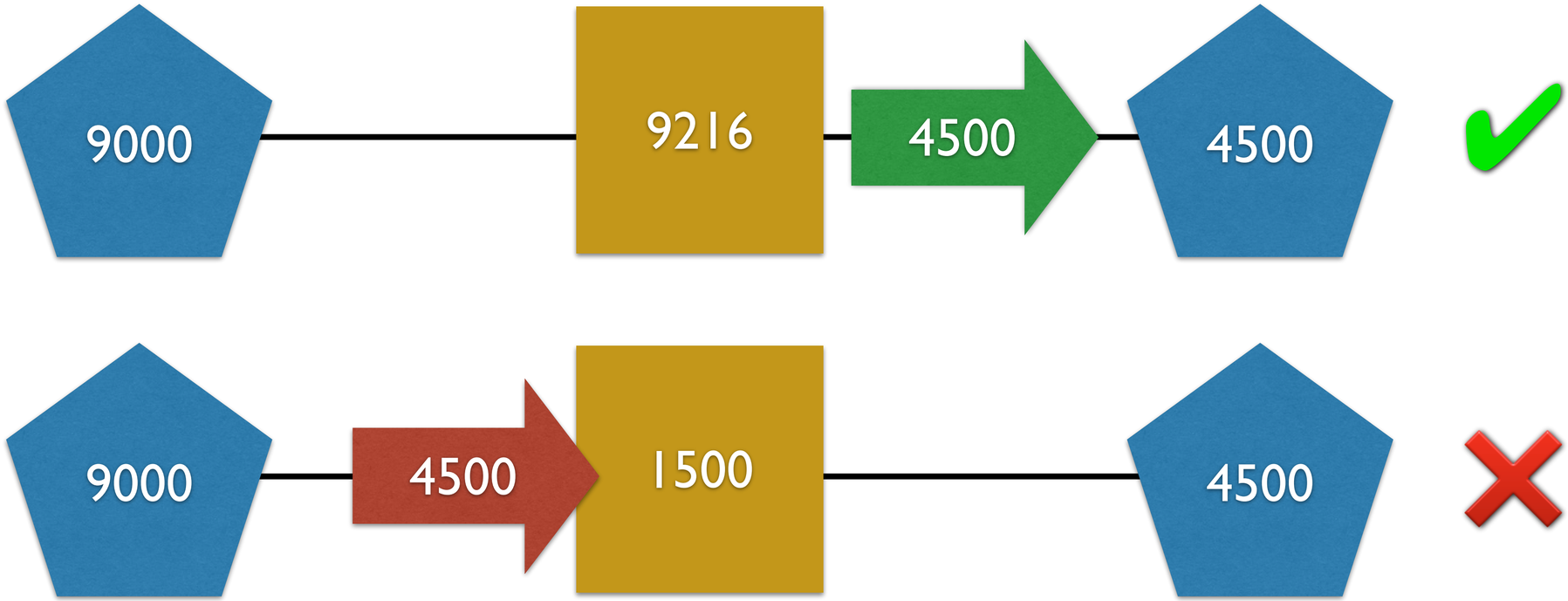
But... IEEE can't do this

- With Ethernet, every packet is self-contained and stateless
  - so you don't know anything about the receiver's capabilities
- But IP can do this:
  - first ARP or Neighbor Discovery before data is exchanged
  - so: put MTU in ARP or ND option

Complications...



Complications...



- So test packets to detect switch limitations

Questions?

http://tools.ietf.org/html/draft-van-beijnum-multi-mtu-03