

# IPv6

Computer Networks  
University of Applied Sciences Augsburg  
Guest lecture

Remotely from The Hague, 16 December 2020  
Iljitsch van Beijnum

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

# Today's topics

- About me
- Origins of IPv6
- Address configuration
- Issues with choice
- How do we get there

# Who is Iljitsch van Beijnum?

- Ended up in the Dutch Internet Service Provider business in the 1990s
- Started my own one-person business in 2000
- Wrote a book about BGP in 2002 and one about IPv6 in 2005
- Contributed to IETF, 7 RFCs
- Wrote for Ars Technica, a big tech website, for about 10 years
- Worked for the Dutch government for a few years

# Origins of IPv6

# 25 years ago...

Network Working Group  
Request for Comments: 1883  
Category: Standards Track

S. Deering, Xerox PARC  
R. Hinden, Ipsilon Networks  
December 1995

## **Internet Protocol, Version 6 (IPv6) Specification**

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

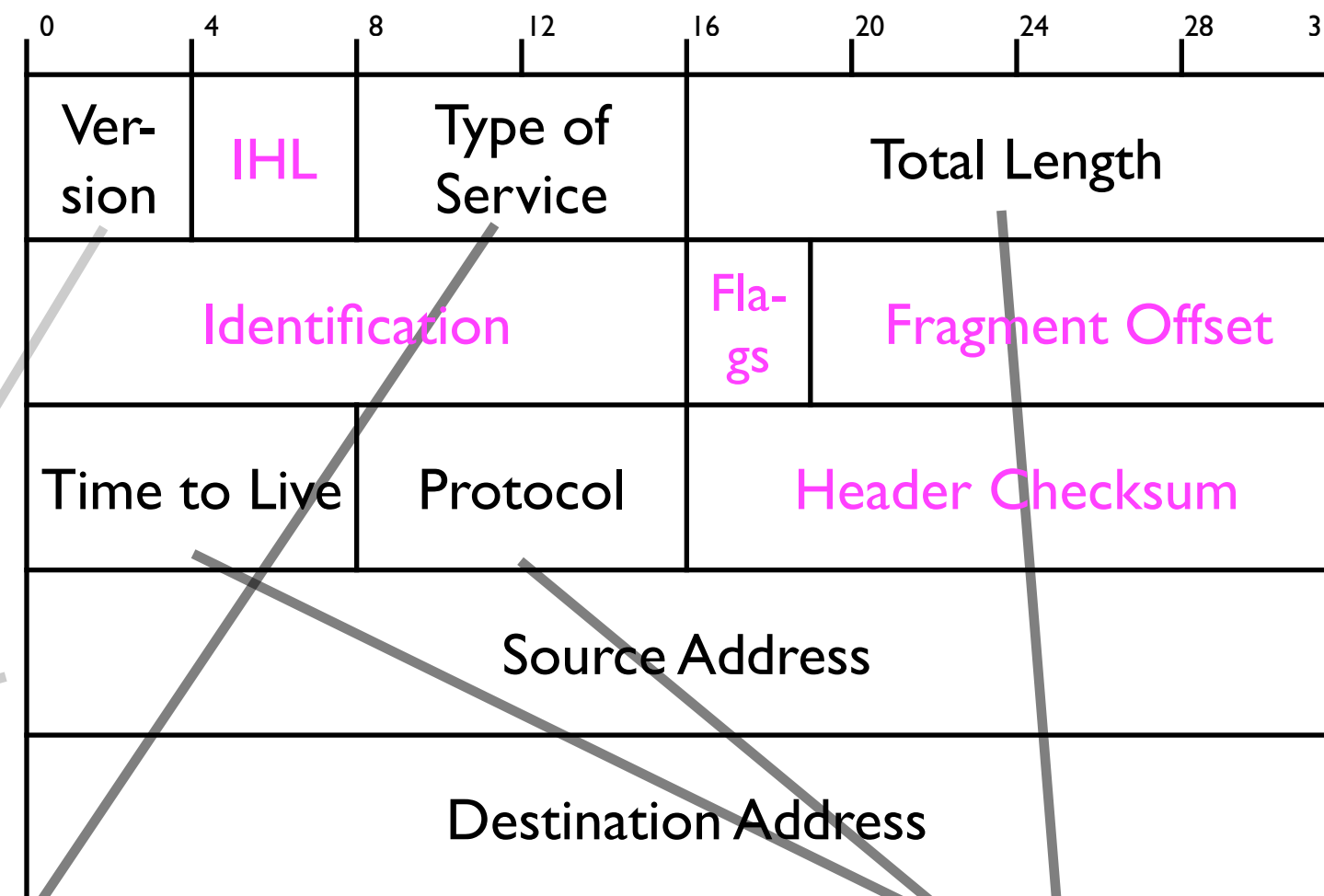
### Abstract

This document specifies version 6 of the Internet Protocol (IPv6), also sometimes referred to as IP Next Generation or IPng.

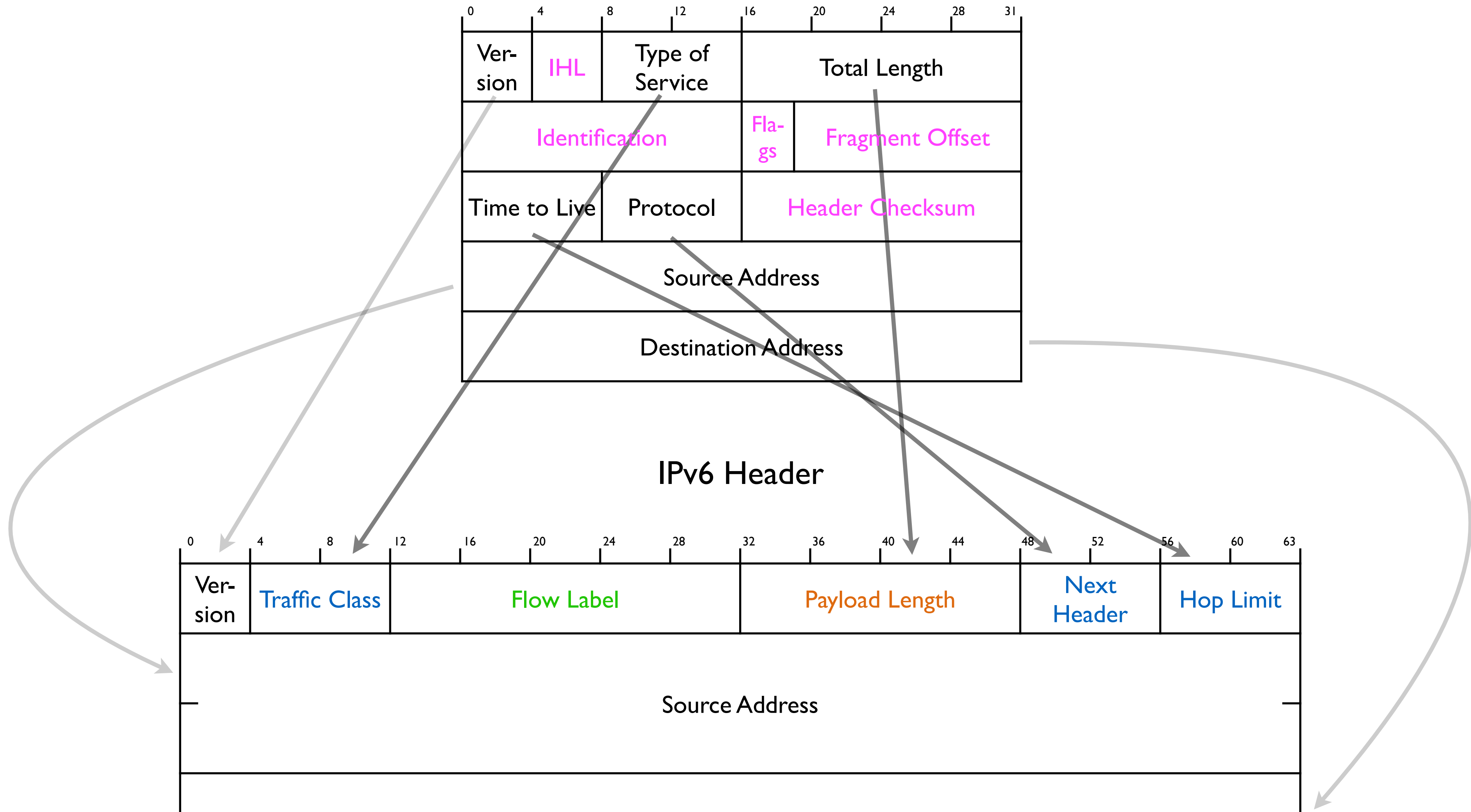
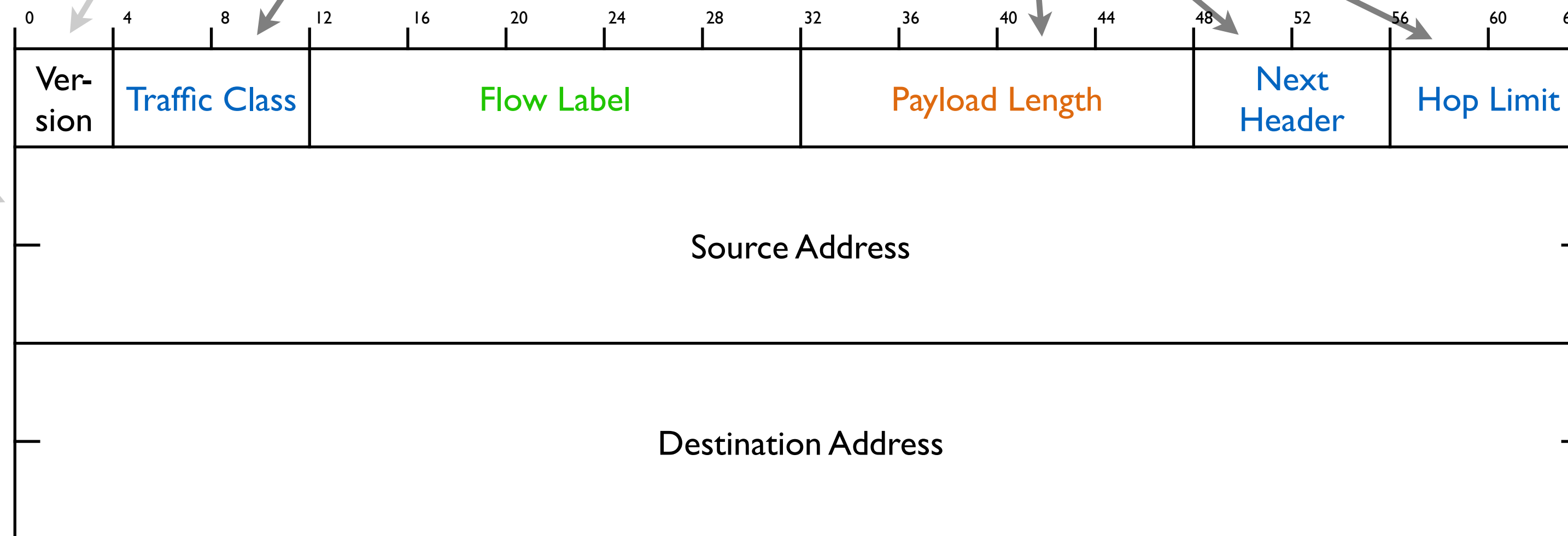
# Why IPv6?

- The Internet Protocol from 1981 uses 32-bit addresses
  - 3.7 billion usable addresses
- By the early 1990s it became obvious that this wasn't enough
- So we needed more addresses = more address bits = a new packet layout = a new protocol version
- Old IP had version 4, nobody knows what happened to 1, 2 and 3, 5 was used for something else, so IPv6

# IPv4 Header



# IPv6 Header



How does your device  
get its address?



# Link-local addresses

- When you turn on IPv6, it generates a *link-local address* for each active network interface
  - a bit like IPv4 169.254.x.x
- So even if there are no IPv6 routers on the network, it's still possible to communicate over IPv6 locally
- The link-local address is kept even after one or more regular "global unicast" addresses are configured

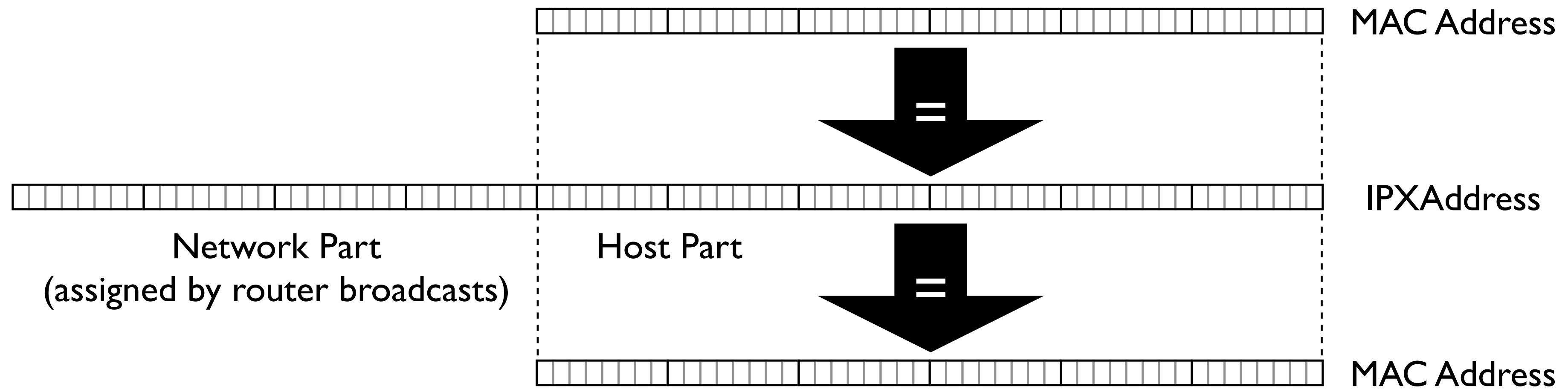
# Addresses hierarchy

- There is a *network* part and a *host* part
- All devices on the same network share the network part
  - (this way routers only have to worry about large blocks of addresses)
- Devices with the same bits in the network part are reachable over the local network
- Local networks have their own addresses (MAC addresses)

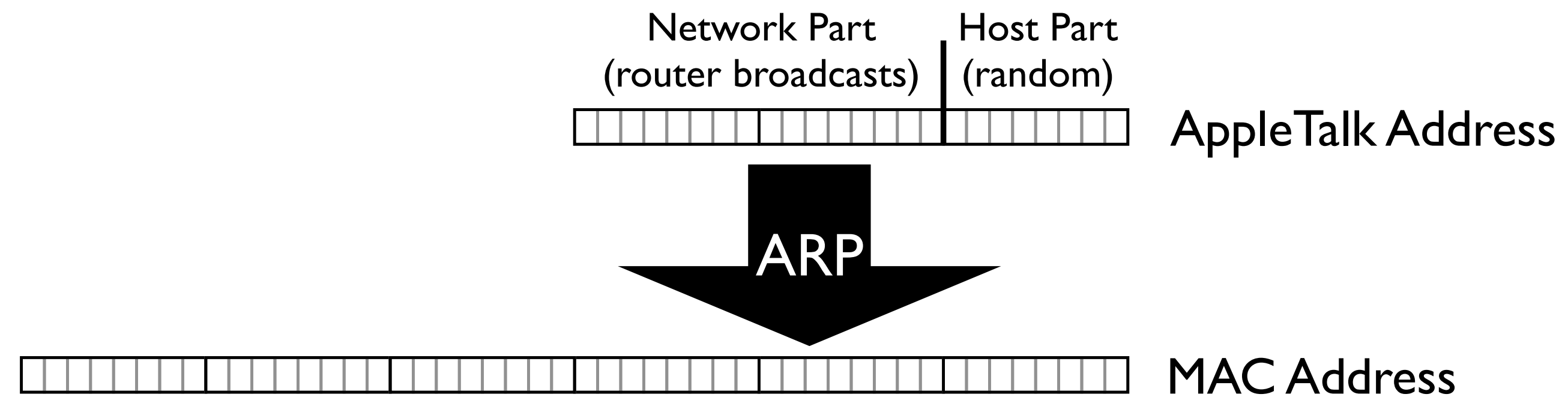
# 1980/90s: multiprotocol!

	Total 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

# IPX



# AppleTalk



# IP(v4)

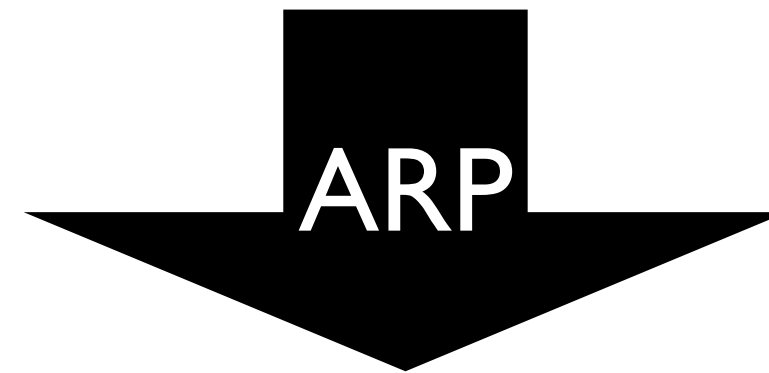
Manual Configuration or DHCP

Network Part

Host Part



IP Address



MAC Address

# Intermezzo: ARP and ND

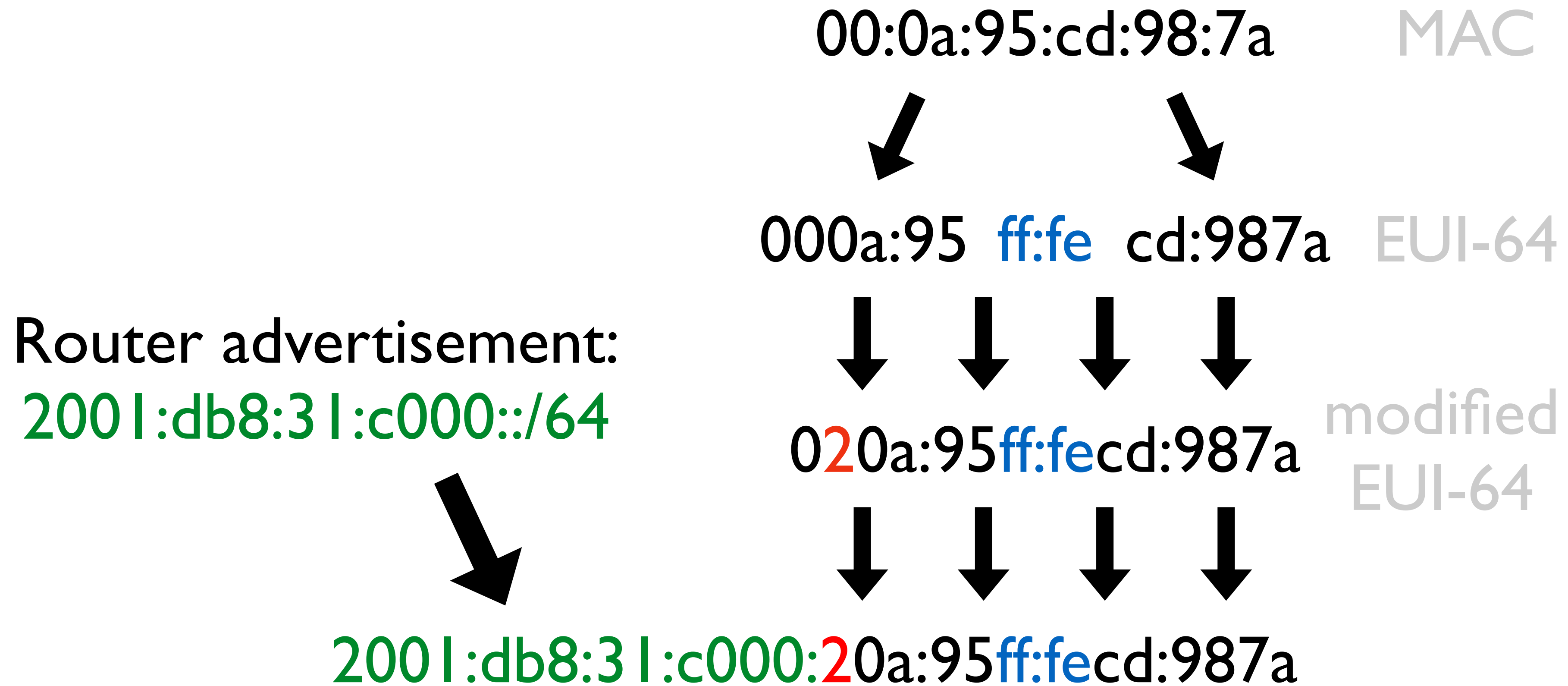
- You need a device's layer 2 (MAC) address to send packets
- You only know its layer 3 (network) address...
- Address Resolution Protocol:
  - ARP request: "Who has this IP address?"
  - ARP reply: "That's me, this is my MAC address"
- IPv6 uses Neighbor Discovery (ND) instead:
  - *neighbor solicitation* and *neighbor advertisement*

# IPv6

- Includes *all* address configuration methods discussed so far:
  - manual configuration
  - stateless autoconfiguration:
    - 64 bits from router + MAC address
    - 64 bits from router + random number
    - (64 bits from router + crypto hash)
  - DHCPv6



# Stateless autoconfiguration



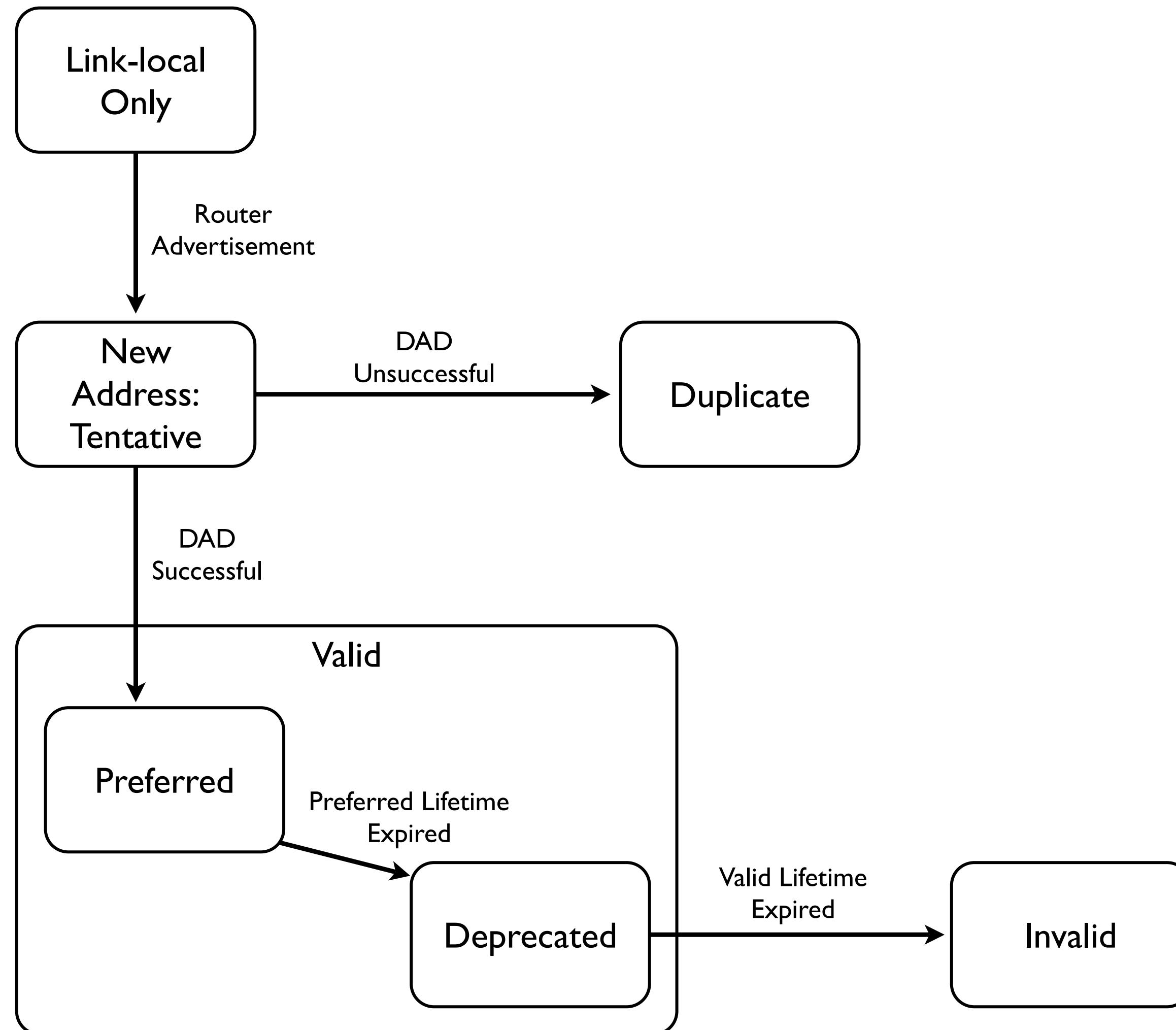
# Address Privacy

- 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 as of Windows Vista/7 and MacOS 10.7

# Duplicate address detection

- Before a device may use an address, see if anyone 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

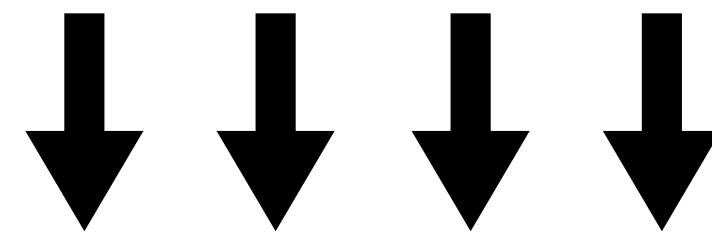
# Lifecycle of addresses



# Address Notation

IPv4:

01011100	00000101	00001100	00000110
----------	----------	----------	----------



92.5.12.6

IPv6:

0010000000000001	110110111000	0000001100010000	0000010111001010	001000001010	1001010111111110	111111011001100	1001100001111010
------------------	--------------	------------------	------------------	--------------	------------------	-----------------	------------------



2001:db8:310:5ca:20a:95ff:fe cd:987a

# Address Notation

- Zero compression for one contiguous range of zeros:
  - `2001:db8:31:0:0:0:0:1` → `2001:db8:31::1`
  - `2001:db8:0:0:ff:0:0:1` → `2001:db8::ff:0:0:1` or `2001:db8:0:0:ff::1`, *not* `2001:db8::ff::1`
- Localhost: `0:0:0:0:0:0:0:1` → `::1`
- Unspecified address: `0:0:0:0:0:0:0:0` → `::`

# Address Types

- Global unicast: for regular world-wide use
- Site local unicast: within own "site"
- Link local unicast: on a "link" (subnet), won't pass a router
- Multicast: one-to-many
- Anycast: one-to-any (shares unicast space)

# IPv6 Address Space

First Bits	Prefix	Purpose
000	::/3	Special uses
001	2000::/3	Global unicast
01 - 1111 1011	4000::/2 - fb00::/8	Reserved
1111 1100	fc00::/8	Unique site local
1111 1110 10	fe80::/10	Link local
1111 1110 11	fec0::/10	Old site local
1111 1111	ff00::/8	Multicast

bit 0 bit 127

2 0 0 1 0 d b 8 0 0 3 l d e a d b e e f l 2 3 4 5 6 7 8 9 a b c

Link-local: fe80::/10

f e 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Old site-local: fec0::/10 (deprecated)

f e c

Unique site-local (ULA): fc00::/7

f d 2 d c f 6 a 6 0 2 2

Multicast: ff00::/8

f f

Global unicast: 2000::/3

2

6to4 tunneling: 2002::/16 + IPv4 address

2 0 0 2 c 0 0 0 0 2 3 l

Teredo tunneling: 2001:0::/32 + IPv4/flags/port/IPv4

2 0 0 1 0 0 0 0 c 0 0 0 0 2 a d 8 0 0 0 3 2 l 0 3 5 2 6 3 f 9 7

MAC address

b d e f l 2 7 8 9 a b c

Interface identifier

b e e f l 2 f f e 7 8 9 a b c

IPv4-mapped: ::ffff:0:0/96 + IPv4 (not on the wire!)

0 f f f f c 0 0 0 0 2 b c

NAT64 well-known: 64:ff9b::/96 + IPv4

0 0 6 4 f f 9 b 0 c 0 0 0 0 2 b c



**Choice is bad**

# Router advertisements

- Routers periodically send an RA packet to let devices configure a default gateway
- and for address configuration, if one or more /64 prefixes are included and *autonomous address configuration flag* = 1
- *RDNSS* (recursive DNS server) option with one or more DNS addresses may be included
- Several timers govern lifetime of this information
- IPv6 doesn't have *broadcasts*, so these are *multicasts*
- So if the device doesn't run IPv6, it doesn't listen for RAs

# 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)

# DHCPv6

- Complete reinvention of DHCP for IPv6
- Completely incompatible with DHCP
- Doesn't provide router address
- Doesn't provide subnet mask / prefix length
- So DHCPv6 must work together with router advertisements

# 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
- used by ISPs to give customers a block of IPv6 address space

# RA flags and DHCPv6

M	O	Prfx	A	Result
0	0	-		default gateway, but no address
0	0	yes	0	default gateway, but no address
0	0	yes	1	working IPv6, but no DNS unless in RDNSS
0	1	-		default gateway + DNS, but no address
0	1	yes	0	default gateway + 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

# (Mostly) not a problem anymore

- All these options and complexity were a problem for some years, but mostly solved now
- Only Android doesn't support DHCPv6, which annoys people who run enterprise networks

But how do we get  
there?



# Why are layer 3 transitions so hard?

- I upgraded from 10 to 100 to 1000 to 10000 Mbps 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

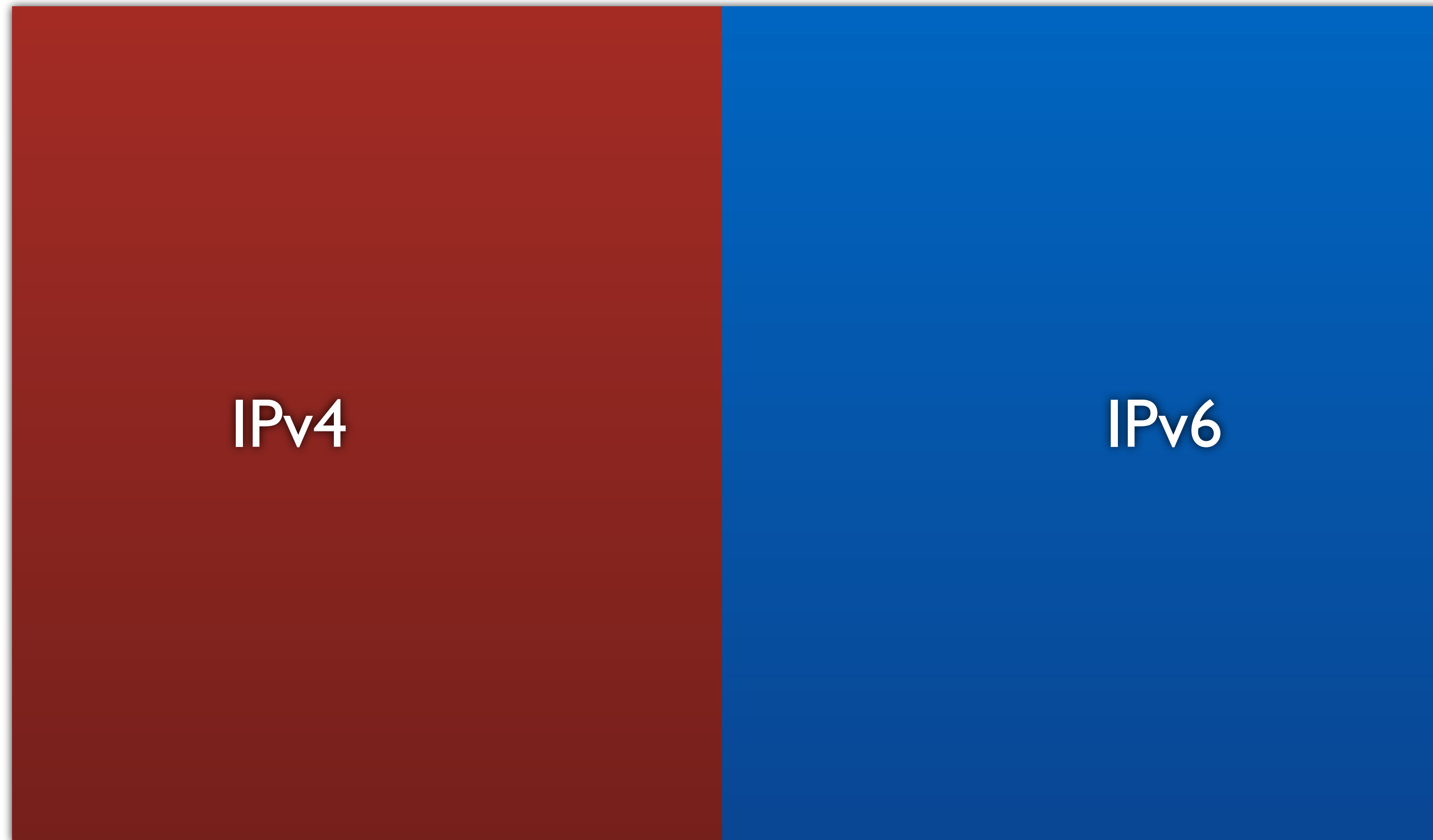
# NCP to IP/TCP

- In the 1970s, the ARPAnet had the Network Control Protocol (NCP)
  - a single protocol for layer 3 and 4
  - monolithic protocol was becoming a problem
- So IP/TCP (now known as TCP/IP or simply IP, or IPv4) 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" doesn't work...

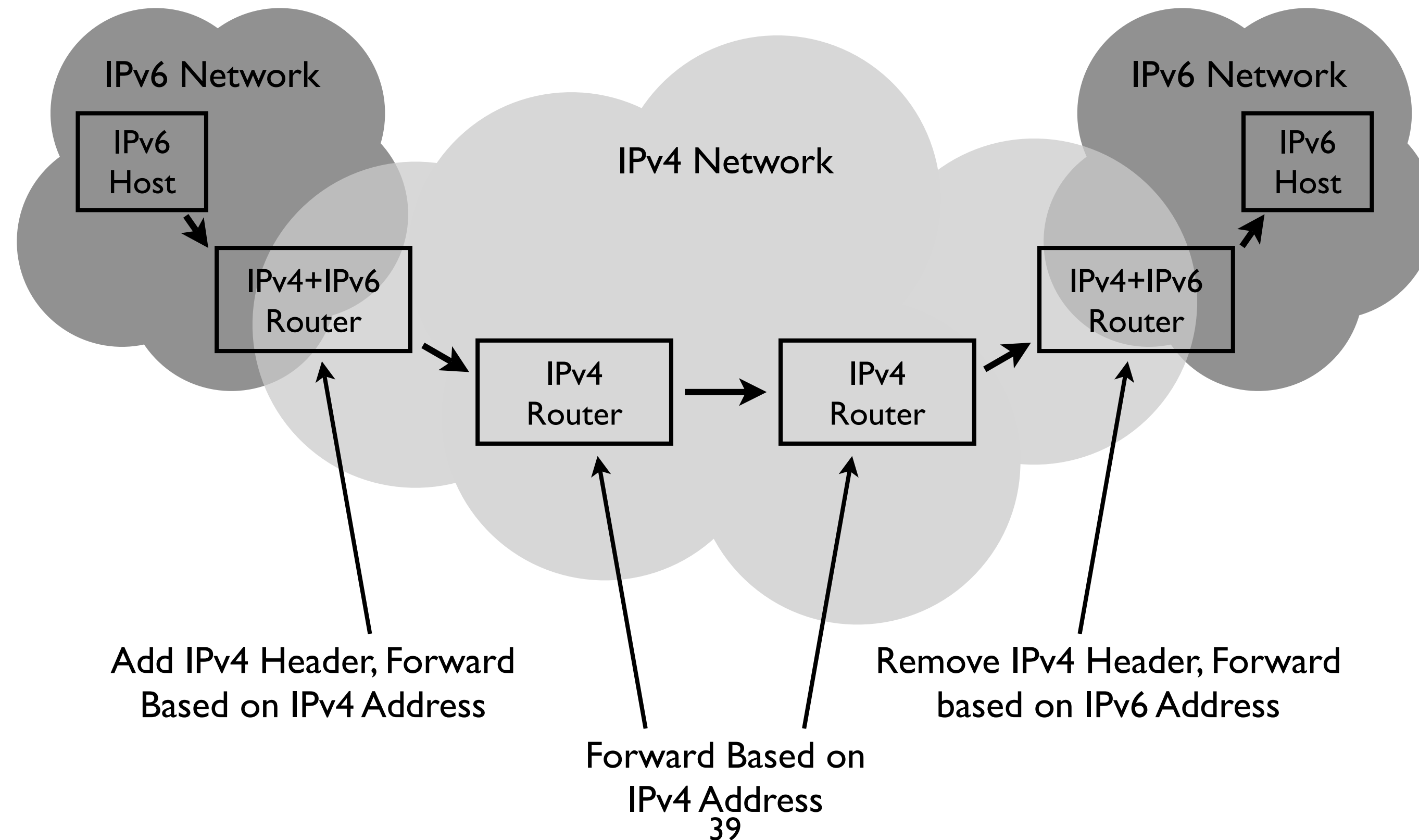


# So: transition technologies

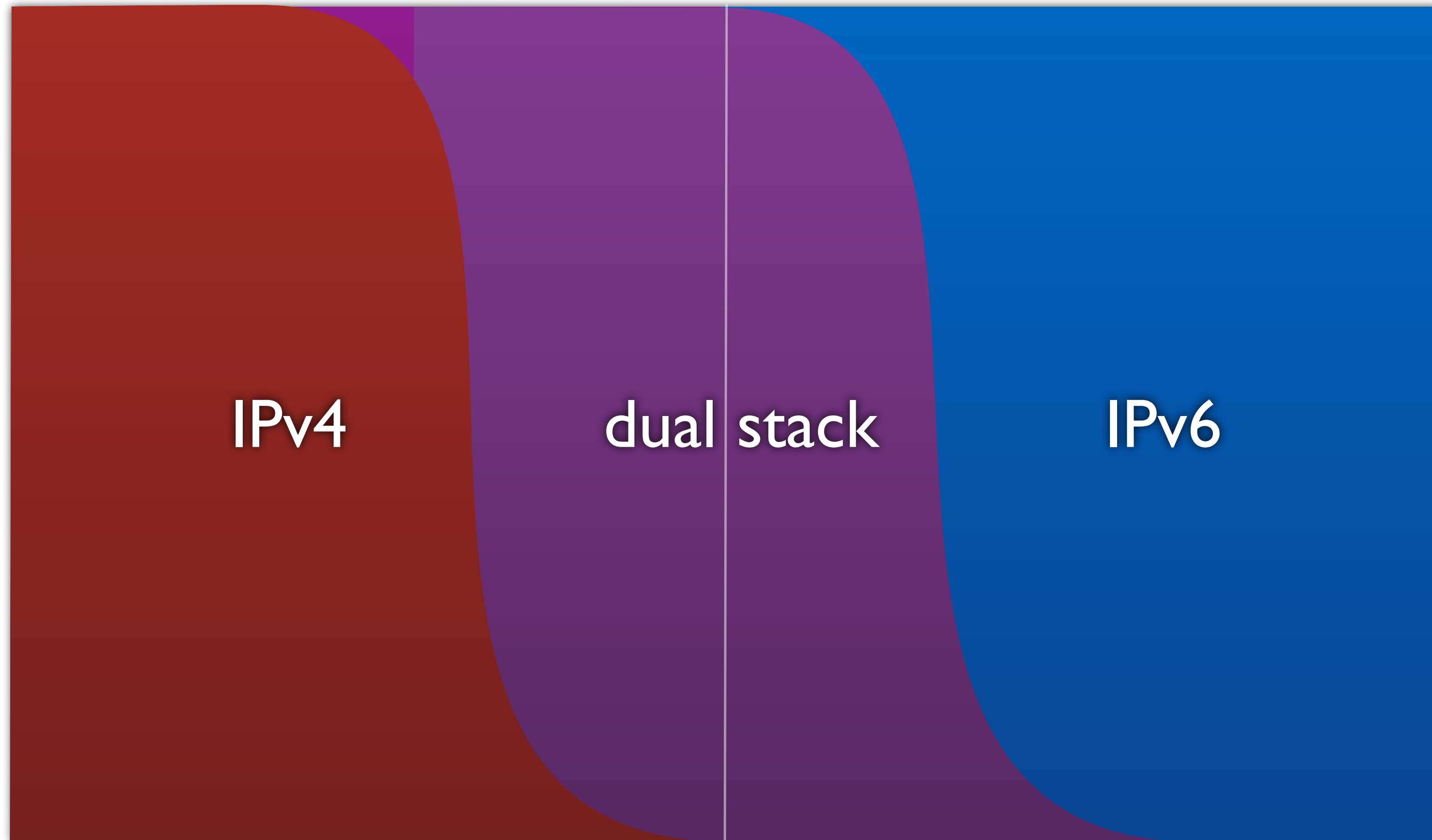
- Tunnels
  - put IPv6 packet inside IPv4 packet to cross IPv4-only parts of the network
    - no longer necessary, we no longer have "IPv6 islands"
- Dual stack
  - everyone keeps IPv4 and adds IPv6
  - then remove IPv4 when everyone has IPv6
- Translation

# Tunnels

- Put IPv6 packet inside IPv4 packet to cross IPv4-only parts of the network



# Dual Stack





# Dual stack and happy eyeballs

- What if you try IPv6 but it doesn't work?
- 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)

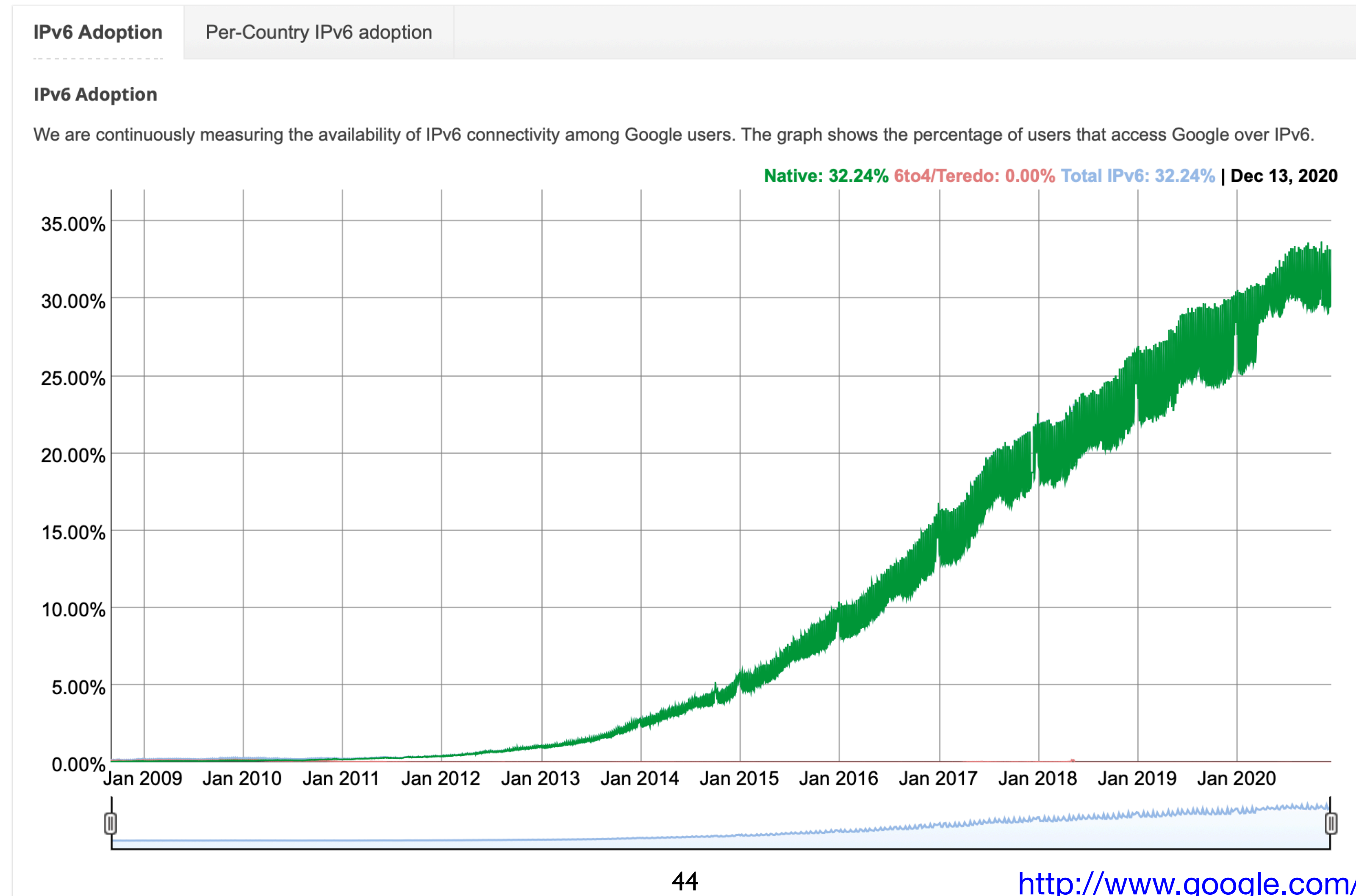
# Happy eyeballs (2)

- 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: ???

**So where are we?**

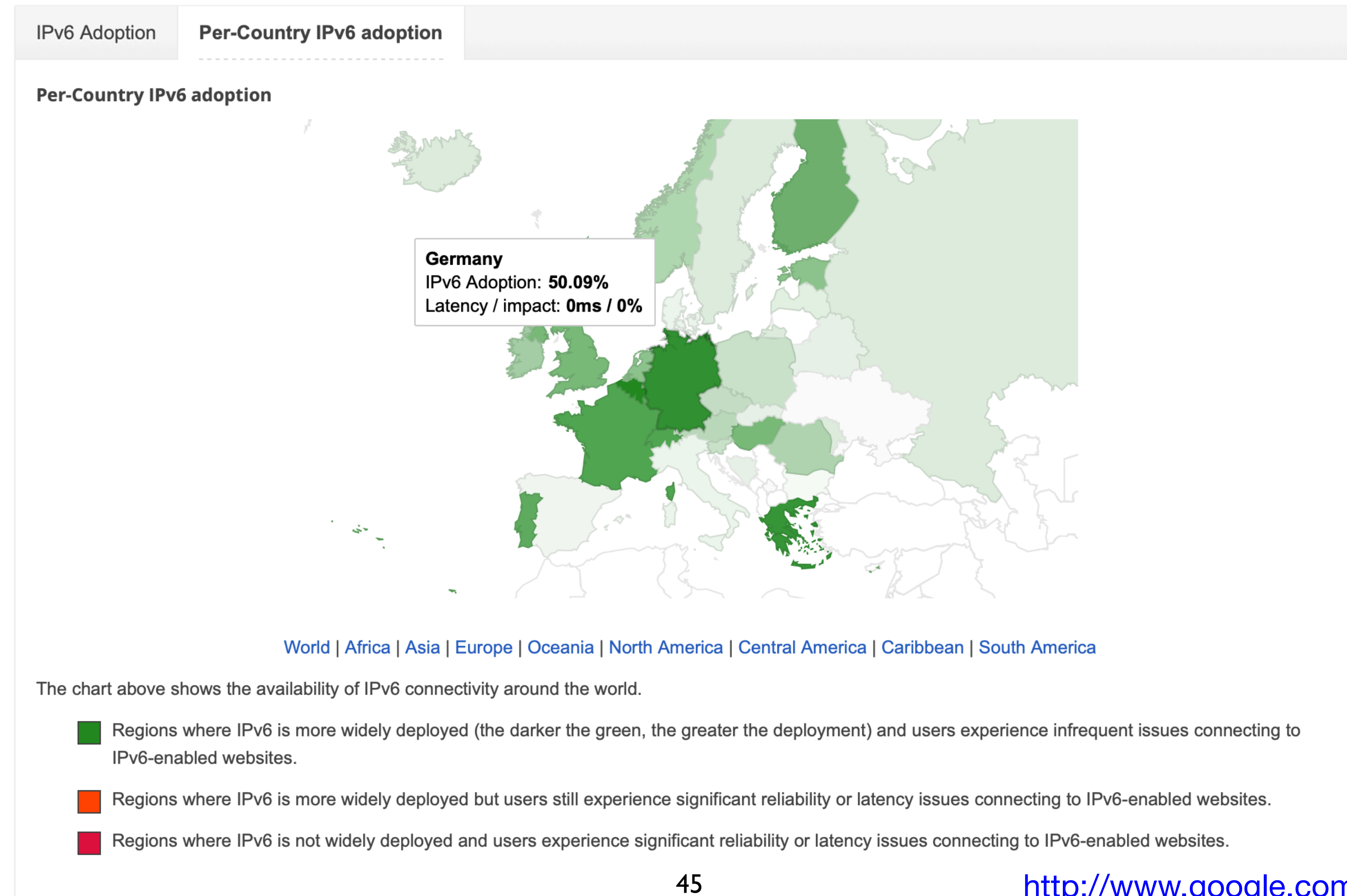
## Statistics

Google collects statistics about IPv6 adoption in the Internet on an ongoing basis. We hope that publishing this information will help Internet providers, website owners, and policy makers as the industry rolls out IPv6.

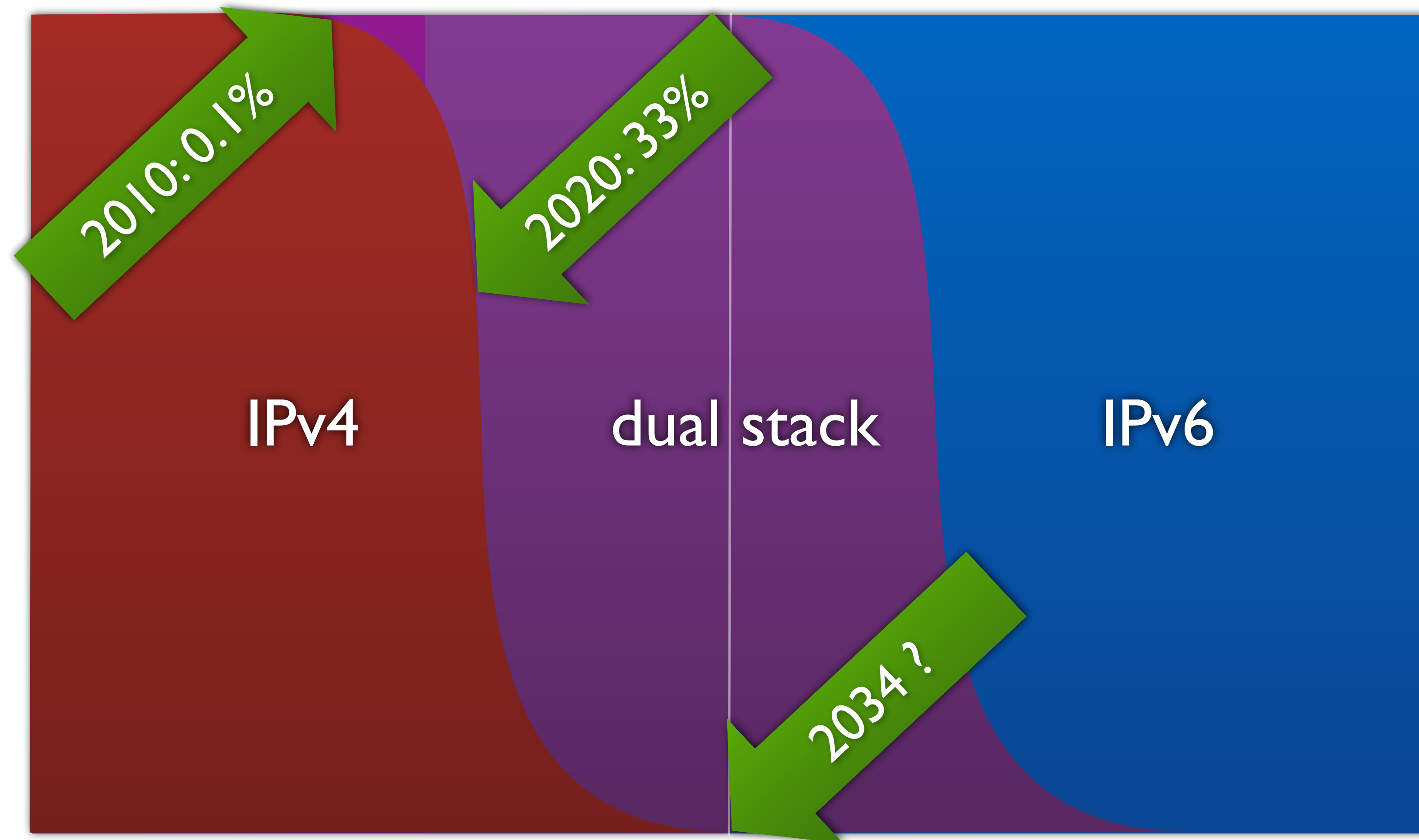


## Statistics

Google collects statistics about IPv6 adoption in the Internet on an ongoing basis. We hope that publishing this information will help Internet providers, website owners, and policy makers as the industry rolls out IPv6.



# This may take some time...



# Staying on IPv4: 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)

# Translating IPv4-IPv6: 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



# IPv6-IPv6 NAT?

- Today almost all IPv4 packets go through one or more Network Address Translators
- NAT was introduced to conserve IPv4 addresses
- There are enough IPv6 addresses, so no need for NAT
- especially as NAT makes protocols more complex and less reliable

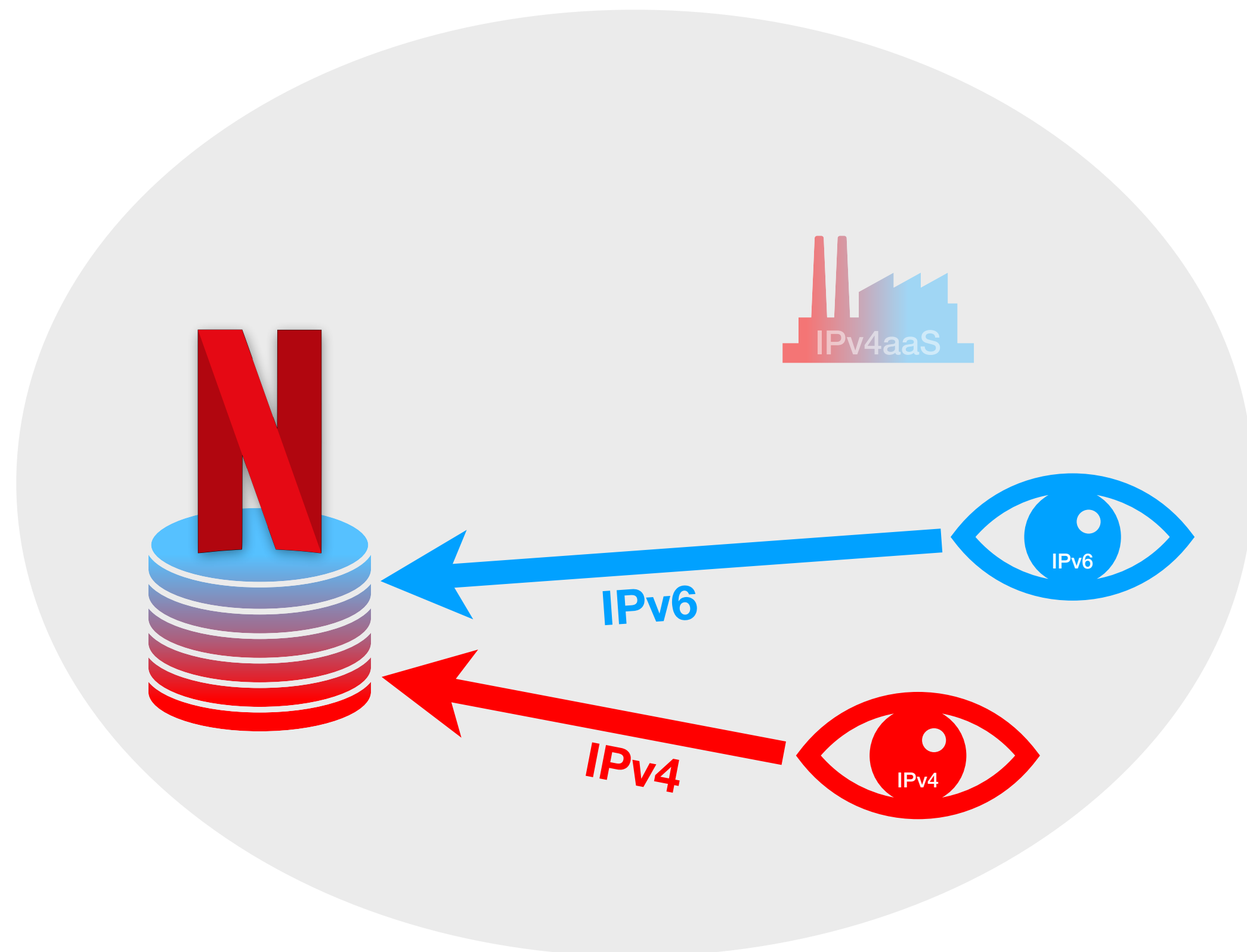
# NAT64 vs NAT444

	<b>NAT64</b>	<b>NAT444</b>
Translated traffic	IPv4 destinations	all traffic
IPv6	supported	orthogonal and breaks most tunnels
IPv4-only applications	only with 464XLAT	supported
DNSSEC	mostly supported	supported
IPv4 literals	unsupported*	supported
Network topology	(can be) simple	complex
Exit strategy	very good	problematic

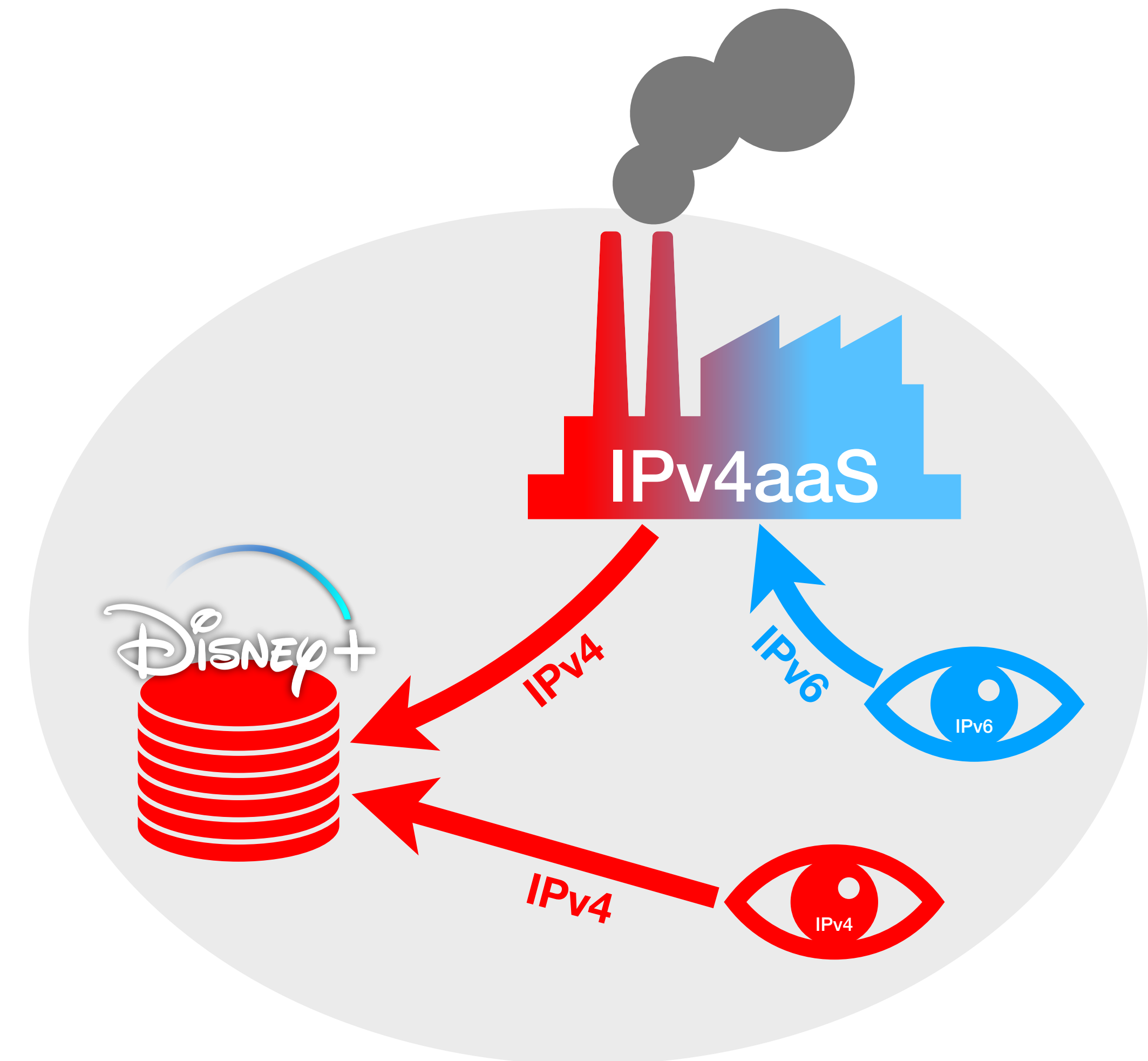
# 464XLAT

- Why are some applications still IPv4-only??
- Apple solves this by requiring iOS applications to support IPv6
- Other OSes try to work around this: translate IPv4 system calls to IPv6, then NAT64 back to IPv4
- (however, those apps will still not be able to talk to IPv6 servers, so more load on the NAT64)

# NAT64 = "IPv4 as a Service"



**vs.**



# NAT64 endgame

- As more services are available on IPv6, NAT64 is used less
- So if IPv6 adoption grows faster than traffic increases, ISPs only face a one-time investment
- Most large services (Google/YouTube, Facebook, Netflix) already use IPv6
- Will be heavy pressure from ISPs on big services that are still IPv4-only to adopt IPv6 so those don't use up NAT64 capacity
- Probably don't care about small stuff remaining IPv4-only

# Questions?

[www.bgpexpert.com](http://www.bgpexpert.com)

# Additional links

- U/L: unique/local bit in MAC addresses
- "Internetwork Packet Exchange (IPX) is the network layer protocol in the IPX/SPX protocol suite. IPX is derived from Xerox Network Systems' IDP."
- GOSIP: Government Open Systems Interconnection Profile
- ULA: Unique local address