

IPv6+Ruby for fun and profit




...

Though actually we'll do profit first and then fun

Toronto Ruby 2024-08-13 - Claus Lensbøl

Claus Lensbøl - Senior Software Engineer @ Humi

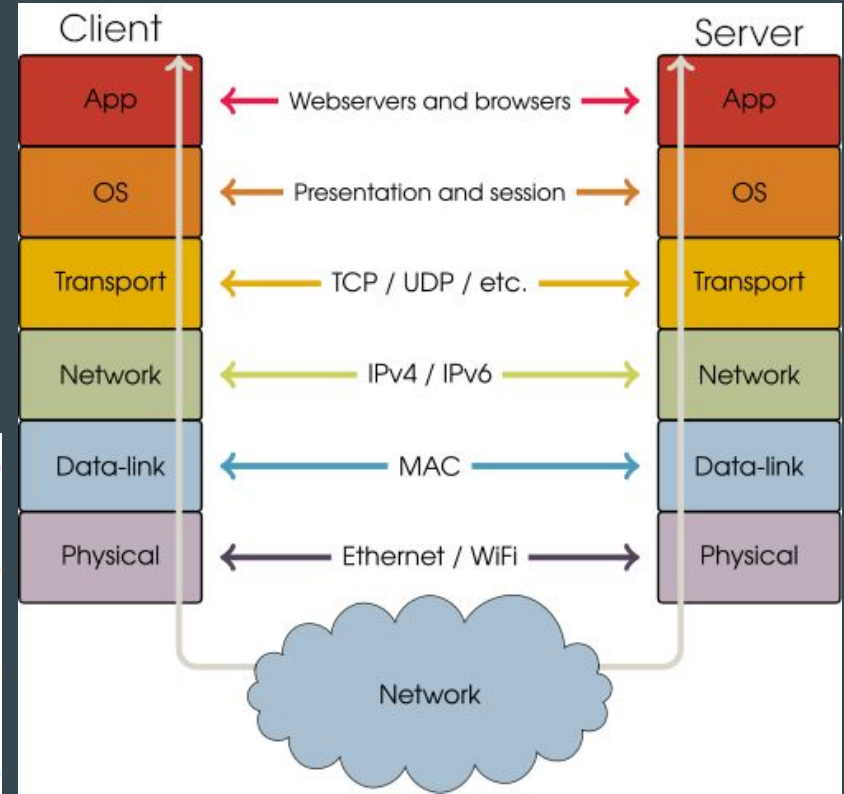
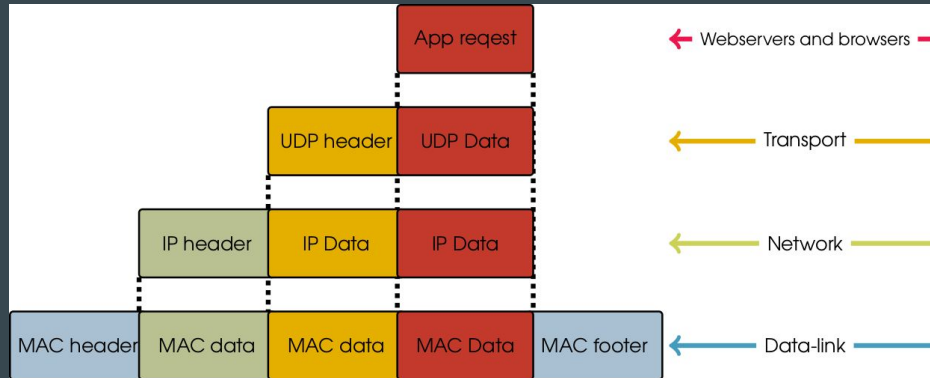
- MSc. Eng. Telecommunication from the Technical University of Denmark
- Background in networking and embedded development

-  <https://github.com/cmol>
-  <https://linkedin.com/in/lensboel>
-  <https://cmol.me> → Scan QR code



What is the internet?

- Collection of networks
- Build on trust
- Hierarchy of protocols



TCP and UDP

TCP

- Connection tracking
- Retransmission
- Reordering
- Stream based
- Larger overhead
- Protocols using TCP
 - HTTP versions ≤ 2
 - SSH
 - SMTP
 - IMAP

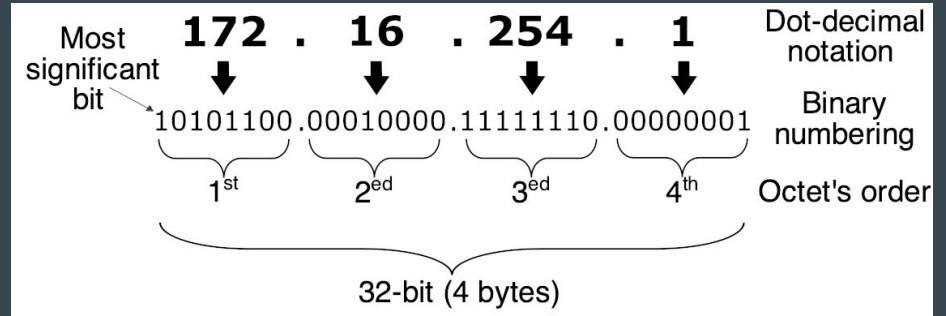
UDP

- No connection tracking
- No retransmission
- No reordering
- Datagram based
- Small overhead
- Protocols using UDP
 - HTTP/3
 - Implements all the connection tracking and retransmission inside QUIC
 - DNS
 - SIP

Two internets

The old and the new

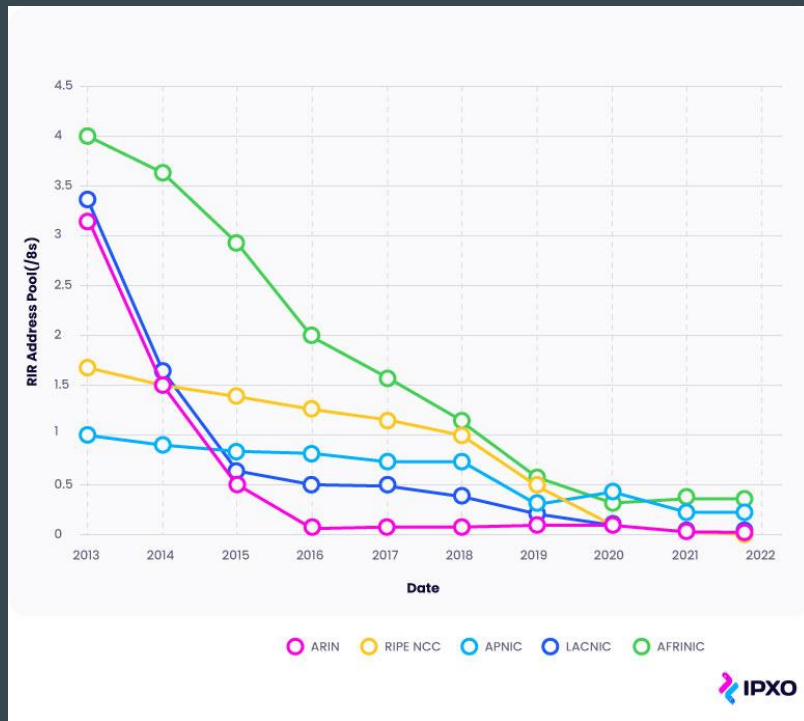
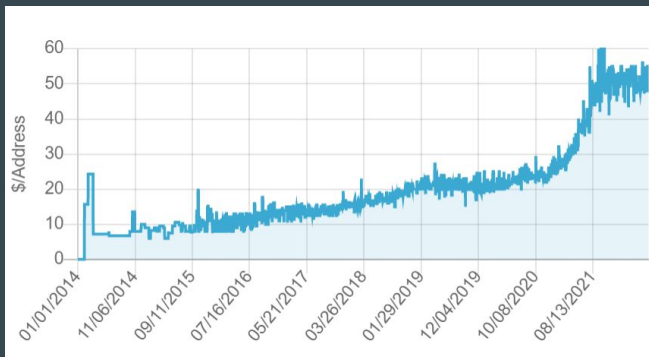
The old - IPv4



- 32-bit addresses
- Networks notated with subnet mask or CIDR
- Not enough addresses
- Dotted decimal notation
- Theoretically has 2^{32} addresses $\sim 4.3 * 10^9$
- Uses variable network sizes for end user devices

IPv4 Problems

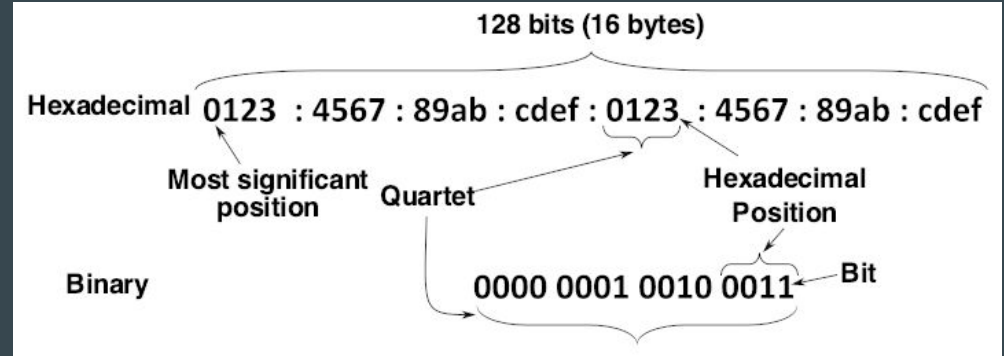
- Not enough addresses
- High price of \$45-60 USD / address
 - Passed on directly to consumers
 - Higher price per IPv4 address for VPCs
 - AWS price ~\$45 USD / year / address
- Multiple layers of NAT
 - Increasing latency and complexity
- Harder to make direct connections
 - Important, but not that relevant to web apps



ARIN experienced depletion of its IPv4 Free Pool on [24 September 2015](#), however we are still processing and approving customer IPv4 requests. Visit the [Waiting List page](#) to learn more about how IPv4 requests are being handled.

The new - IPv6

- Longer addresses (128-bit)
- Uses hexadecimal CIDR notation
- Many addresses! $\sim 3.4 \cdot 10^{34}$
- 340,282,366,920,938,463,374,607,431,768,211,456 addresses



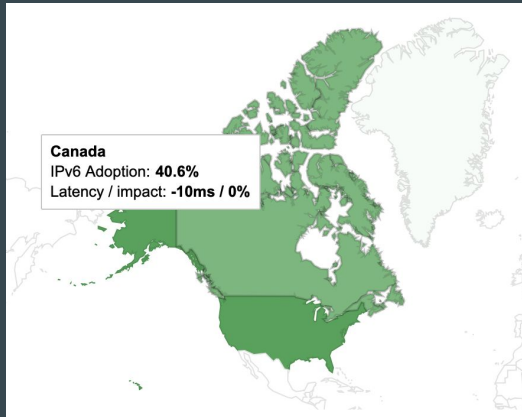
What's the IPv6 usage in Canada?

...

What's the IPv6 usage in Canada?

...

Around 40%



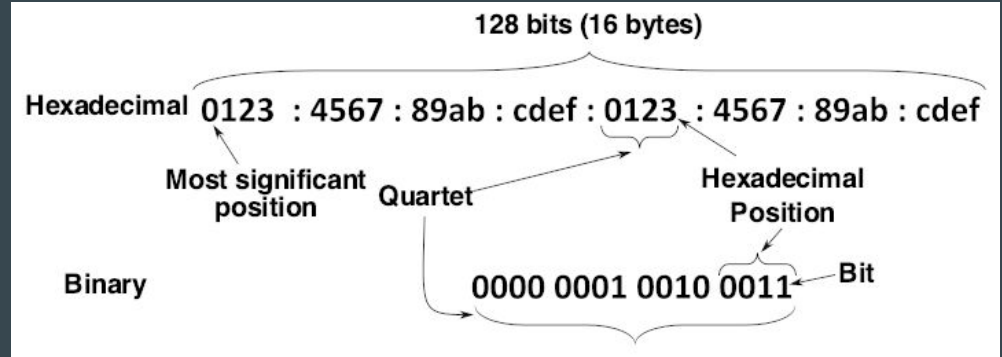
Google IPv6 Country Adoption - Canada

Country adoption graph based on data from [Google IPv6 Statistics](#) page.



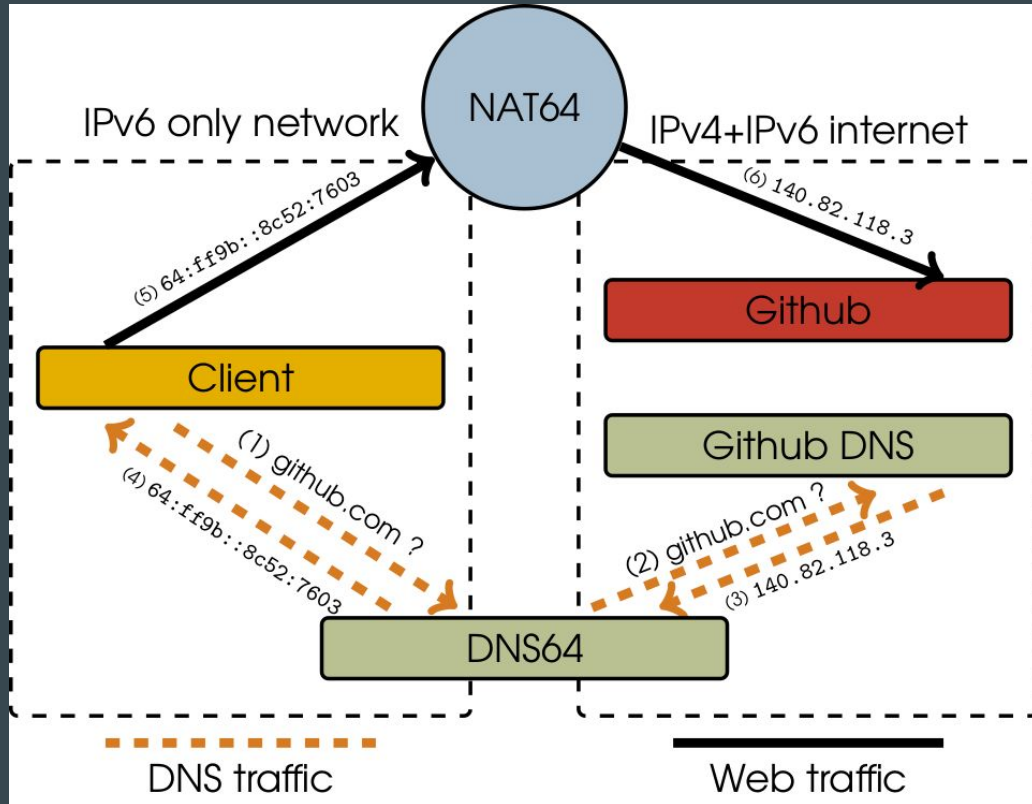
IPv6 advantages

- Lower latency (according to google/apple/microsoft)
- Better for SEO (because of lower latency)
- More or less free
- Simpler protocol



IPv6 only networks access to IPv4 service

- Uses NAT64+DNS64
 - (or other similar options)
- Does not work with DNSSEC or DoH (no security)
- Short story, it's not great for the end user
- Adds no complication to IPv6 enabled services



(So we should add IPv6 to our applications!)

IPv6 Myths - 1

- No NAT is like no security
 - NAT is not a firewall
 - NAT slipstreaming exists
 - Companies like ebay scans your “NAT protected” network anyway:
<https://blog.nem.ec/2020/05/24/ebay-port-scanning/>

IPv6 Myths - 2

- You can be tracked everywhere with IPv6
 - Early days IPv6 used MAC addresses to generate IPv6 addresses in a bad way
 - Fixed in [RFC4941](#) (2007) and [RFC7217](#) (2014)

IPv6 Myths - 3

- We don't need to support IPv6, we have plenty of IPv4 addresses / No one uses it
 - See rest of presentation

Do you have IPv6?

<https://ipv6check.me>



In brief

- IPv4 is
 - a. Expensive
 - b. Slow(er than IPv6)
- IPv6 is
 - a. Fast(er than IPv4)
 - b. Free
 - c. Good for your SEO

Now fun!

TCP unicast

Demo of classic client server interaction

Code at: <https://github.com/cmol/to-ruby-2024-08>

UDP unicast

Demo of classic client server interaction

Code at: <https://github.com/cmolt/to-ruby-2024-08>

UDP Multicast

Demo of 1 to many communication

Code at: <https://github.com/cmol/to-ruby-2024-08>

Discussion! / Questions?

Resources

- <https://www.google.com/intl/en/ipv6/statistics.html>
- <https://blog.cmol.me/ipv6-ruby-part-1-introduction-to-ipv6-f45c7cf18151>
- https://en.wikipedia.org/wiki/Internet_Protocol
- https://en.wikipedia.org/wiki/IPv4_address_exhaustion
- <https://www.rfc-editor.org/rfc/rfc3493>

Resources - Multicast sniffer

- <https://github.com/cmol/dns/blob/main/examples/mdns/main.go>
 - Run with: `go run ./examples/mdns [interface_name]`