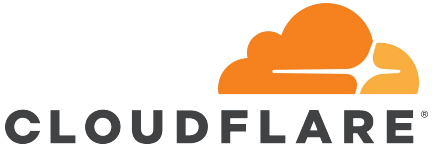
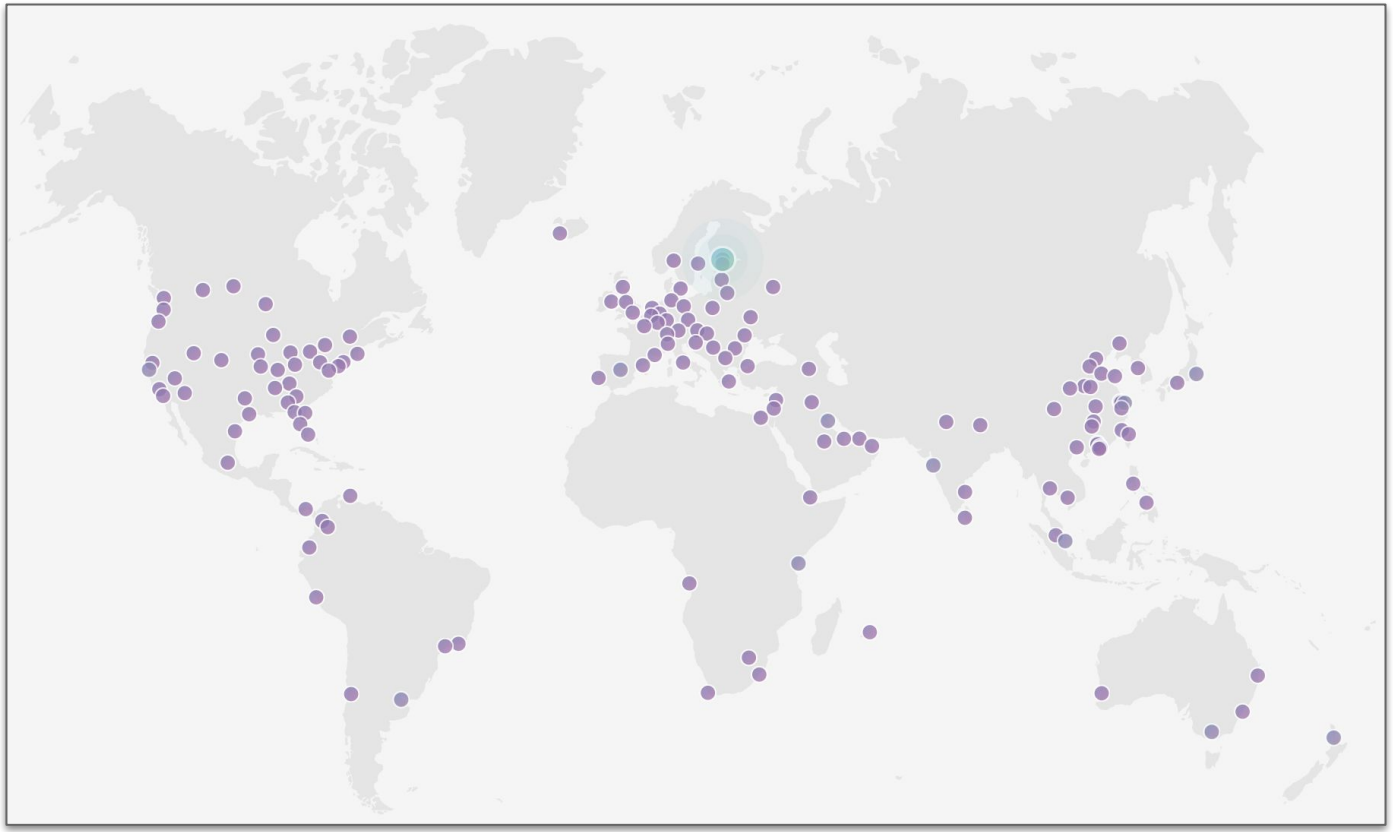


DNS over anything but UDP: Implications

Ólafur Guðmundsson

The Cloudflare network (DNS, DDoS, CDN, WAF, more)



154+

Data centers globally

154+

DNS resolver locations

154+

DNS authoritative locations

DNS challenges at scale

Authoritative DNS

- Data distribution

Resolving Origin names:

- DNS “distance” and unreliable

Public resolver:

- DNS “distance” and unreliable

DDoS, Route Hijacks, Injected Answers

- Distances
- Packet drops
- Timeouts
- Detecting EDNS0 support
- Server selection
- Retransmission policy
- Forged answers
- Firewalls

CLOUDFLARE

Why mostly UDP ?

Fast, no-state in OS on servers

works in 1 RTT

Easy to start

BUT ……..

not every place has perfect network

I was a big UDP bigot

We know how it works

Internet has changed
more bad actors

CLOUDFLARE

UDP issues: connection less

- **No flow control**
  - DNS software must implement
- **Fragments**
  - Blocked, size issues …
- **In the clear**
- **Forged answers**
  - First one wins
  - On-path attacker wins always

Retry
EDNS capability discovery
Path MTU discovery
Old broken software
Lots of state to store, and update.

DNS developers not good transport protocol designers
Bad defaults
Not updated
First world centric

Privacy leaks
Packet inspection
Easy to lie

CLOUDFLARE

Connections solve what ?

- **Fragmentations and size issues**
- **Flow control and retry policy**
- **Better integrity in answers**
- **Get firewalls out of the way**

- **Simpler clients and resolvers**

Lots of DNS “servers” do not answer over TCP or any other connection oriented protocol

TCP is badly/not supported in some existing code bases

Firewalls do may or may not pass through

CLOUDFLARE

Connection oriented transports

There are many different transport protocols
Each address different solution spaces

DNS over reliable transport

DNS inherits modern properties

- Flow control
- No more fragmenting
- Authenticated connections

Drawback: Connection setup and teardown

Do One Thing Well, outsource others

Need to separate flow control from message integrity

CLOUDFLARE

DNS over TCP

Part of DNS from day one

Considered: Slow and high overhead

But: “Long” lived connections with out-of-order processing bring cost down to net gain

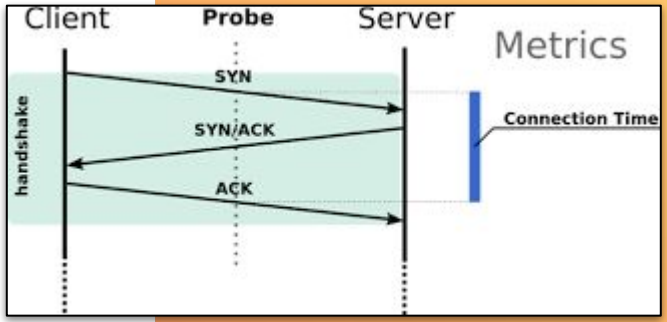
TCP has got much better than you learned in school!!

Adds:
Flow Control,

Minimal integrity,

Eliminates retries to same address

No Fragments or size issues



CLOUDFLARE

DoT: DNS over TLS

- **Defined for stub to recursor**
  - RFC7858[1] & RFC8310[2] on port **853**

- **Acts like normal TCP connection**
  - setup is different and more expensive
  - Session resumption is essential
  - Many implementations

Highlight features:
Data integrity
Assurance of connected party in strict mode

Can be discovered and used in optimistic mode

TLS termination can be done by external plugin like Nginx

Certificate managment overhead

CLOUDFLARE

[1] https://tools.ietf.org/html/rfc7858
[2] https://tools.ietf.org/html/rfc8310

DoH: DNS over HTTPS

UDP or Json blob in HTTP on port 443

DoH[1] pending RFC publication

Envisioned as Application to Resolver protocol

Can work for stub to Recursor

Will get through any firewall that passes HTTPS

Firefox, Chrome, and some applicaitons support

1.1.1.1, 8.8.8.8, 9.9.9.9

Uses UDP wire format or simple JSON

Depends on HTTP2 for good performance

Requires knowledge to find servers

May allow migrating DNS traffic to same connection as HTTP traffic

TLS1.3 has 0RTT

CLOUDFLARE

[1] https://datatracker.ietf.org/doc/draft-ietf-doh-dns-over-https/

DoQ: DNS over QUIC

Proposed work[1]

QUIC is datagram protocol with TLS built in

Matches DNS properties well

Not ready for standardization

QUIC is raises interesting options for extending DNS

No implementations

CLOUDFLARE

[1] https://datatracker.ietf.org/doc/draft-huitema-quic-dnsoquic/?include_text=1

Performance implications

The world changes over time, what we hold as true may not stay the same due to advances.
Without looking at the facts we are doomed to failure
Change can be quick or slow but change will happen

Main factors

Connection multiplexing

- **out-of-order answers**

Distance

Connection Resumption

BAD:
Connection for one query
Answers in same order
More distant server selected

Good:
Long lived connection
Out-of-order answers
Closest server
Reduced complexity

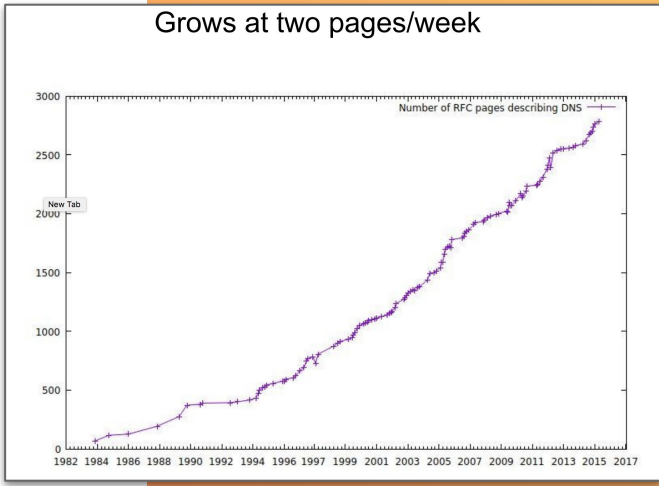
CLOUDFLARE

Simplicity and reliability

DNS software is too complex

Route Hijacks: Secure connections will detect and fail

Resolver to Authority: work in progress



CLOUDFLARE

What do you want from upstream DNS?

- Assurance you are talking to the right one
- Fast and accurate answers
- Reliability

What's missing

Discovery of "local" servers

Expression of resolver policies

Recursive 

Authority

Users accept DCHP supplied DNS

Users configure addresses

Applications have URL's for DNS

Getting around stupid network setups (1.1.1.1), only port 443

Confidence this will work

CLOUDFLARE

Q/A

Open floor for any questions that you may have