DNS-over-QUIC

Sara Dickinson, Sinodun
(support from Salesforce)
sara@sinodun.com

OARC 26, Madrid May 2017

What is QUIC?

- Experimental protocol, deployed at Google starting in 2014
  - Chrome: Improved page load latency, video rebuffer rate
  - Successful, deployed, experimental protocol today
    - **~35% of Google's egress traffic** (~7% of Internet traffic)
    - Akamai deployment in 2016
- QUIC Working Group formed at IETF in Oct 2016
  - Standardize QUIC, HTTP as initial application

How does QUIC work?

- Runs over UDP (Deployable, userspace Impl)
- Creates encrypted QUIC connection (TLS-like HS)
- Multiplexes ‘streams’ on connection (SDPY-like)
- Version negotiation (Easily evolved wire format)
- Does good stuff: No Head-of-line blocking, Congestion control, Loss recovery, resilient to NAT-rebinding, 0-RTT resumption (like TLS 1.3)

Why use it for DNS?

- Good mixture of features of UDP and TCP
    - Performant, low latency, reliable
    - Source address validation
    - No path MTU limitation
- Privacy properties similar to TLS

Ticks lots of boxes for DNS

DNS-over-QUIC

- DNS on dedicated QUIC connections
  - Stub to Recursive
    - Dedicated port: requests port 953 (oops)
    - Alternative protocol for QUIC development process
- DNS on existing QUIC connections (and HTTP/2)
  - Port 443, avoid issues with DNS port blocking

First drafts submitted
April 2017

QUIC Implementations

- Chromium (open source)
  - https://cs.chromium.org/chromium/src/net/quic/
- quic-go (open source implementation in Go)
  - https://github.com/lucas-clemente/quic-go

Early implementations, specification is still evolving