Assessing Support for DNS-over-TCP in the Wild*

Jerome Mao
Case Western Reserve University

Michael Rabinovich
Case Western Reserve University

Kyle Schomp
Akamai Technologies

* The work of Jiarun Mao and Michael Rabinovich was supported in part by NSF through grant CNS-1647145.

Angles Considered

- DNS-over-TCP Support by Recursive Resolvers
- DNS-over-TCP Support by Authoritative DNS Servers
- Race Condition between resolvers and ADNS

TCP-Fallback Support by Recursive Resolvers: Methodology and Datasets

- General approach:
  - Compel a resolver to engage with our ADNS
  - Our ADNS forces TCP fallback through truncated UDP response without answer records
  - Judge resolver's support by the presence of TCP follow-up

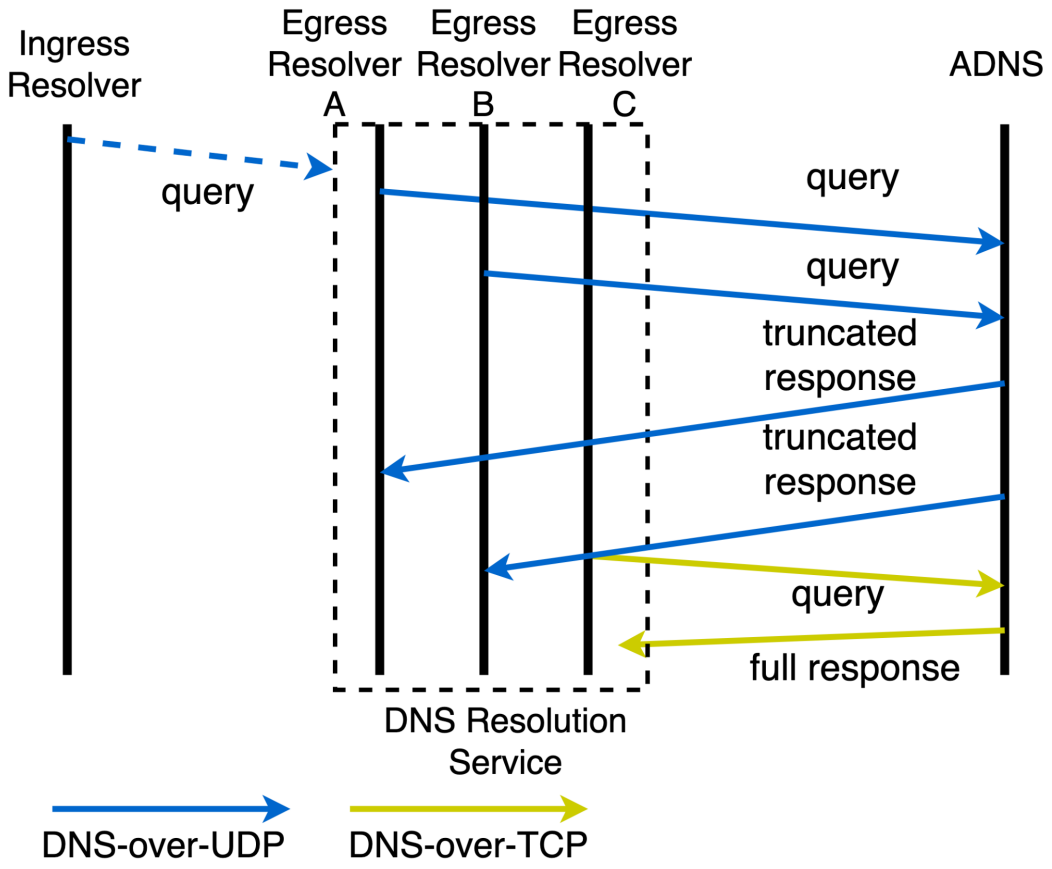
- Open IPv4 resolver scan with unique queries to our own domains

- Email bouncing scan
  - Send email to non-existing recipients at domains from the Majestic top-1M list, from our own domain
  - Corporate resolvers engage with our ADNS to send bounce messages for email delivery failures

- RIPE Atlas scan with unique queries to our own domains

- Major CDN's ADNS logs (combined from all servers)
  - Used to assess the real-world activity of resolvers from different categories

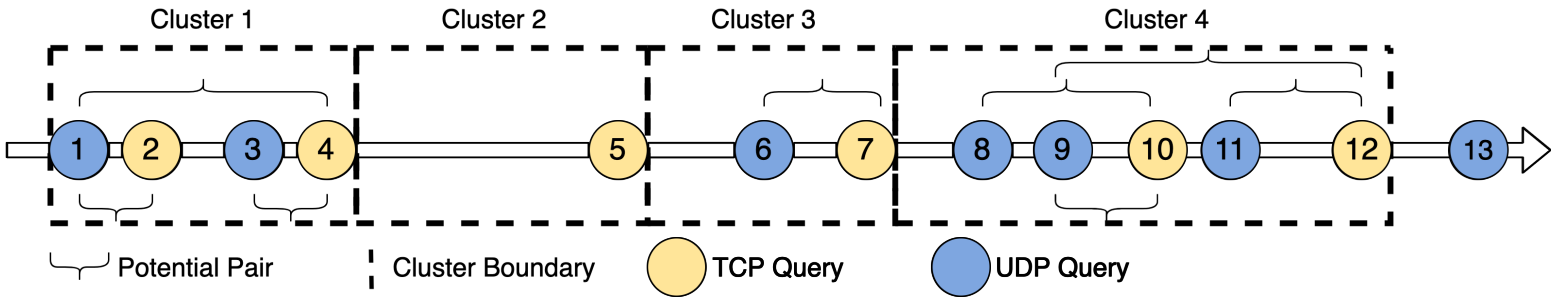
Challenge - Complex TCP-fallback Scenarios



- TCP-fallback capable - either a resolver itself is capable of fallback to TCP, or has a peer that falls back to TCP for it

Canonical and Non-Canonical Scenarios

- Non canonical scenarios are common
  - Only 46.8% of all resolutions are canonical
  - Even among canonical scenarios, 18.9% have the two queries coming from different IP addresses
- Non canonical scenarios are common and can be complicated to match:
  - Real example 1: $U_{r1}U_{r2}U_{r3}U_{r4}U_{r3}T_{r5}T_{r6}T_{r4}T_{r3}$
  - Real example 2: $U_{r1}U_{r1}T_{r1}U_{r2}T_{r2}U_{r3}T_{r3}$
- Algorithm - Group queries by their potential fallback-relationships:



TCP Fallback Support by Resolvers: Results

- Some DNS transactions don't allow unambiguous inference of TCP-fallback capability of a resolver
  - *optimistic*: consider "indeterminate" as TCP-fallback capable
  - *pessimistic*: consider "indeterminate" as TCP-fallback incapable

- Total # of resolvers studied: 116,851
  - ~95 - 97% of resolvers are TCP-fallback capable
  - TCP-fallback capable resolvers contribute to ~96 - 99% of the CDN traffic from all the resolvers studied

- There is non-negligible # of TCP-fallback incapable resolvers and they are about equally active as TCP-fallback capable resolvers

TCP Support by ADNS: Methodology and Datasets

- General approach: attempt to send TCP queries to ADNS serving certain domains from a testing machine on campus

- Domains from queries handled by the resolution service operated by the major CDN
  - Engage all ADNS for the domain

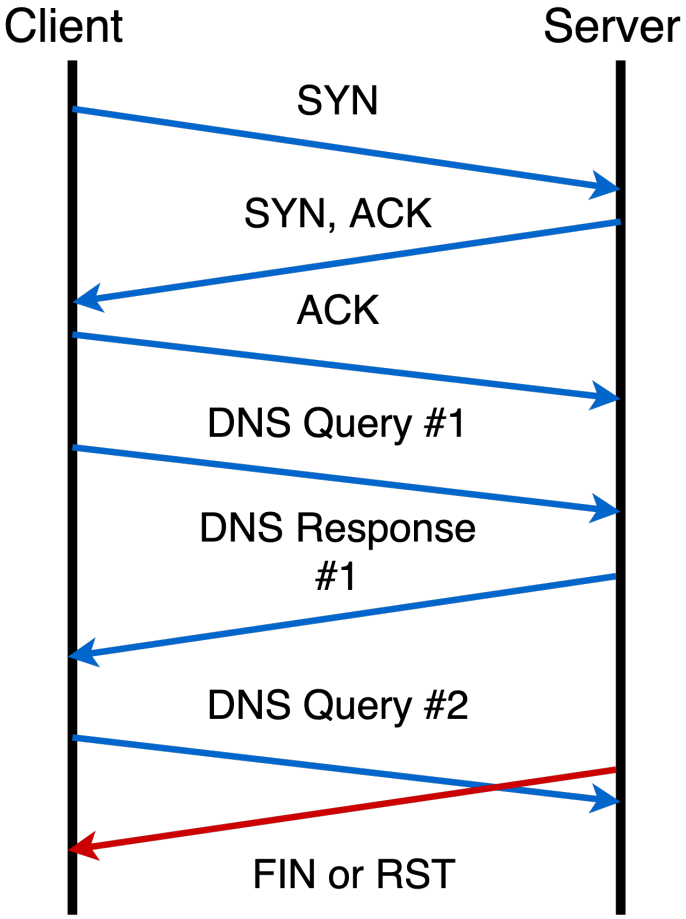
- Majestic top 1000 “root domain” websites (“popular websites”)
  - Engage all ADNS for the domain

- CDN-accelerated domains
  - one domain per CDN

TCP Support by ADNS: Results

- Domains from queries handled by the resolution service operated by the major CDN
  - >5% domains fail to resolve a TCP query through some ADNS
- Majestic top 1000 “root domain” websites
  - >3% domains fail to resolve a TCP query through some ADNS
- CDN-accelerated domains
  - 11 CDNs (out of 47 CDNs studied) deployed ADNS that do not support DNS-over-TCP

Resolver v. ADNS Race Condition (Connection Reuse Inconsistency)



- RFC 7766 recommends reusing established TCP connections
- Resolvers do reuse connections (13.5% enterprise resolvers have been successfully induced to reuse TCP connections)
- Race: the server closes the connection after sending a response, the client reuses the connection for further queries before learning of the closure
- ~33% popular websites, and 4 CDN providers deploy ADNS that close connections immediately

Addressing the Connection Reuse/ Closing Race

1. A resolver MUST NOT reuse a TCP connection unless an explicit edns-tcp-keepalive negotiation has been completed.

2. A resolver MUST NOT reuse a connection beyond the negotiated keepalive duration.

3. An ADNS MUST retain an active connection for 2 MSL beyond the negotiated keepalive duration.

4. Potential optimization:

    - A resolver may indicate its support for TCP connection reuse in a (new) EDNS0 option with its initial UDP query.
    - An ADNS may then indicate a default keepalive value with its UDP TC response.
    - The client can choose any keepalive value that does not exceed the indicated default. The ADNS MUST accept this value during the TCP interaction.

Conclusion

- A small but non-negligible number of recursive resolvers do not support TCP fallback, and they are active
- A non-negligible number of top websites and CDN providers use authoritative servers that do not support DNS-over-TCP
- Many authoritative servers that do support DNS-over-TCP are vulnerable to race condition