draft-ietf-dnsop-dns-tcp-requirements

IETF 104 Prague

tcmp working group

Duane Wessels, John Kristoff

Abstract

This document encourages the practice of permitting DNS messages to be carried over TCP on the Internet. It also describes some of the consequences of this behavior and the potential operational issues that can arise when this best common practice is not upheld.

History

- RFC 1123 [1989]:
  - DNS resolvers and recursive servers MUST support UDP, and SHOULD support TCP, for sending (non-zone-transfer) queries.
- RFC 1536 [1993]
  - UDP is, therefore, the chosen protocol for communication though TCP is used for zone transfers.
- Commonly believed that TCP in DNS is necessary only for zone transfers
- Blocked by firewalls, to protect against unauthorized zone transfer?
- Late 1990s: dynamic updates, DNSSEC, EDNS0.
- RFC 5966 [2010] addresses misconceptions.
  - But sort of goes unnoticed.
  - Doesn’t go far enough.

Dealing with Large Responses

- Beyond MTU sizes, the choice is to fragment or truncate.
- Fragments sometimes blocked by firewalls.
- Truncation generally leads to TCP.
  - also sometimes blocked
  - retry adds latency
- DNS clients need complex retry logic, including EDNS0 buffer size hunting.

RFC 7766 – DNS Transport over TCP Implementation Requirements

- Implementation Requirements.
- Revises and Obsoletes RFC 5966.
- Makes no recommendations to operators.

- All general purpose DNS implementations MUST support both UDP and TCP.
- Resolver MAY elect to use TCP first, without first using UDP.
- RECOMMENDED to keep idle connection open for “seconds.”
  - RFC 1035 said “two minutes.”
- Servers MAY impose limits on number of TCP connections.
- Clients MUST be able to handle out-of-order responses.

This Draft – Operational Requirements

- Encourage operators to ensure that DNS over TCP is on par with UDP.
- Authoritative servers MUST service TCP queries so that they do not limit the size of responses to what fits in a single UDP packet.
- Recursive servers MUST service TCP queries [for similar reasons].
- A name server MAY limit the resources it devotes to queries, but it MUST NOT refuse to service a query just because it would have succeeded with another transport protocol.
- Filtering of DNS over TCP is considered harmful in the general case.
- Network operators MUST allow DNS service over both UDP and TCP transports.

Connection Admission

- SYN cookies are effective in mitigating SYN flood attacks.
- Services not intended for use by the public Internet SHOULD be protected with access controls.
- FreeBSD has the dns_accf(9) accept filter.
- Applications MUST NOT be configured to refuse TCP queries that were not preceded by a UDP query.
- DNS servers SHOULD enable TCP Fast Open when possible.
    - and clusters SHOULD use the same key on all instances
- DNS clients SHOULD enable TFO when possible.

Connection Management

- DNS servers MUST actively manage their connections.
- Server software SHOULD provide a configurable limit on the total number of established TCP connections.
  - Operators SHOULD ensure the configured limit is appropriate.
- Server software MAY provide a configurable limit on the number of connections per source IP address or subnet.
- Server software SHOULD provide a configurable timeout for idle connections.
  - Clients and servers SHOULD signal their timeouts with edns-tcp-keepalive
- Server software MAY provide a configurable limit on the number of transactions per connection.
- Server software MAY provide a configurable limit on the total duration of a TCP connection.

Connection Termination

- Preferable for clients to initiate the close of a TCP connection.
- On systems where lots of TIME_WAIT is observed, it may be beneficial to tune local TCP parameters.
- In extreme cases, busy serves may find it necessary to use SO_LINGER with a value of zero.

Miscellany

- These recommendations also apply to DNS-over-TLS (RFC 7858).
- The draft does not make any reference to RFC 8484 “DNS Queries over HTTPS.”
- Developers of applications that log or monitor DNS are advised to not ignore TCP, keeping in mind:
  - connection reuse
  - pipelining
  - out-of-order responses
  - small segments / slow writes
  - overlapping segments

Questions?