Network Working Group   C. Huitema
Internet-Draft   Private Octopus Inc.
Intended status: Standards Track   M. Shore
Expires: July 5, 2018   Fastly
                                                            A. Mankin
                                                           Salesforce
                                                         S. Dickinson
                                                           Sinodun IT
                                                           J. Iyengar
                                                               Google
                                                       January 1, 2018

Specification of DNS over Dedicated QUIC Connections
draft-huitema-quic-dnsoquic-03

Abstract

   This document describes the use of QUIC to provide transport privacy
   for DNS. The encryption provided by QUIC has similar properties to
   that provided by TLS, while QUIC transport eliminates the head-of-
   line blocking issues inherent with TCP and provides more efficient
   error corrections than UDP. DNS over QUIC (DNS/QUIC) has privacy
   properties similar to DNS over TLS specified in RFC7858, and
   performance similar to classic DNS over UDP.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Domain Name System (DNS) concepts are specified in [RFC1034]. This
   document presents a mapping of the DNS protocol [RFC1035] over QUIC
   transport [I-D.ietf-quic-transport] [I-D.ietf-quic-tls]. The goals
   of this mapping are:

   1. Provide the same DNS privacy protection as DNS over TLS (DNS/TLS)
       [RFC7858]. This includes an option for the client to
       authenticate the server by means of an authentication domain name
       [I-D.ietf-dprive-dtls-and-tls-profiles].

   2. Provide an improved level of source address validation for DNS
       servers compared to DNS/UDP [RFC1035].

   3. Provide a transport that is not constrained by path MTU
       limitations on the size of DNS responses it can send.

   4. Explore the potential performance gains of using QUIC as a DNS
       transport, versus other solutions like DNS over UDP (DNS/UDP)
       [RFC1035] or DNS/TLS [RFC7858].

   5. Participate in the definition of QUIC protocols and API, by
       outlining a use case for QUIC different from HTTP over QUIC
       [I-D.ietf-quic-http].

   In order to achieve these goals, the focus of this document is
   limited to the "stub to recursive resolver" scenario also addressed
   by [RFC7858]. That is, the protocol described here works for queries
   and responses between stub clients and recursive servers. The
   specific non-goals of this document are:

   1. No attempt is made to support zone transfers [RFC5936], as these
       are not relevant to the stub to recursive resolver scenario.

   2. No attempt is made to evade potential blocking of DNS/QUIC
       traffic by middleboxes.

   Users interested in zone transfers should continue using TCP based
   solutions. Users interested in evading middleboxes should consider
   using solutions like DNS/HTTPS [I-D.ietf-doh-dns-over-https].

Specifying the transmission of an application over QUIC requires
specifying how the application's messages are mapped to QUIC streams,
and generally how the application will use QUIC. This is done for
HTTP in [I-D.ietf-quic-http]. The purpose of this document is to
define the way DNS messages can be transmitted over QUIC.

In this document, Section 3 presents the reasoning that guided the
proposed design. Section 4 specifies the actual mapping of DNS/QUIC.
Section 5 presents guidelines on the implementation, usage and
deployment of DNS/QUIC.

2. Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119].

3. Design Considerations

This section and its subsection present the design guidelines that
were used for the proposed mapping of DNS/QUIC. This section is
informative in nature.

3.1. Scope is Limited to the Stub to Resolver Scenario

Usage scenarios for the DNS protocol can be broadly classified in
three groups: stub to recursive resolver, recursive resolver to
authoritative server, and server to server. This design focuses only
on the "stub to recursive resolver" scenario following the approach
taken in [RFC7858] and [I-D.ietf-dprive-dtls-and-tls-profiles].

QUESTION: Should this document specify any aspects of configuration
of discoverability differently to DNS/TLS?

No attempt is made to address the recursive to authoritative
scenarios. Authoritative resolvers are discovered dynamically
through NS records. It is noted that at the time of writing work is
ongoing in the DPRIVE working group to attempt to address the
analogous problem for DNS/TLS [I-D.bortzmeyer-dprive-step-2]. In the
absence of an agreed way for authoritative to signal support for QUIC
transport, recursive resolvers would have to resort to some trial and
error process. At this stage of QUIC deployment, this would be
mostly errors, and does not seem attractive. This could change in
the future.

The DNS protocol is also used for zone transfers. In the zone
transfer scenario ([RFC5936]), the client emits a single AXFR query,
and the server responds with a series of AXFR responses. This

creates a unique profile, in which a query results in several
responses. Supporting that profile would complicate the mapping of
DNS queries over QUIC streams. Zone transfers are not used in the
stub to recursive scenario that is the focus here, and seem to be
currently well served by the DNS over TCP (DNS/TCP). There is no
attempt to support them in this proposed mapping of DNS to QUIC.

3.2. Provide DNS Privacy

DNS privacy considerations are described in [RFC7626]. [RFC7858]
defines how to mitigate some of these issues by transmitting DNS
messages over TLS and TCP and [I-D.ietf-dprive-dtls-and-tls-profiles]
specifies Strict and Opportunistic Usage Profiles for DNS/TLS
including how stub resolvers can authenticate recursive resolvers.

QUIC connection setup includes the negotiation of security parameters
using TLS, as specified in [I-D.ietf-quic-tls], enabling encryption
of the QUIC transport. Transmitting DNS messages over QUIC will
provide essentially the same privacy protections as [RFC7858] and
[I-D.ietf-dprive-dtls-and-tls-profiles]. Further discussion on this
is provided in Section 7.

3.3. Design for Minimum Latency

QUIC is specifically designed to reduce the delay between HTTP
queries and HTTP responses. This is achieved through three main
components:

1. Support for 0-RTT data during session resumption.

2. Support for advanced error recovery procedures as specified in
    [I-D.ietf-quic-recovery].

3. Mitigation of head-of-line blocking by allowing parallel delivery
    of data on multiple streams.

This mapping of DNS to QUIC will take advantage of these features in
three ways:

1. Optional support for sending 0-RTT data during session resumption
    (the security and privacy implications of this are discussed in
    later sections).

2. Long-lived QUIC connections over which multiple DNS transactions
    are performed, generating the sustained traffic required to
    benefit from advanced recovery features.

3. Mapping of each DNS Query/Response transaction to a separate
    stream, to mitigate head-of-line blocking.

These considerations will be reflected in the mapping of DNS traffic
to QUIC streams in Section 4.2.

3.4. Development of QUIC Protocols and API

QUIC is defined as a layered protocol, with application-specific
mapping layered on top of the generic QUIC transport. The only
mapping defined at this stage is HTTP over QUIC [I-D.ietf-quic-http].
Adding a different mapping for a different application contributes to
the development of QUIC.

HTTP/QUIC parallels the definition of HTTP/2.0, in which HTTP queries
and responses are carried as series of frames. The HTTP/QUIC mapping
provide with some simplication compared to HTTP/TLS/TCP, as QUIC
already provides concepts like stream identification or end of stream
marks. Dedicated control channel are used to carry connection data,
such as settings or the relative priority of queries. It would be
completely possible to use the HTTP/QUIC mapping to carry DNS
requests as HTTP queries, as specified in
[I-D.ietf-doh-dns-over-https]. We are somewhat concerned that this
mapping carries the overhead of HTTP into the DNS protocol, resulting
in additional complexity and overhead.

In this document a different design is deliberately explored, in
which clients and servers can initiate queries as determined by the
DNS application logic, opening new streams as necessary. This
provides for maximum parallelism between queries, as noted in
Section 3.3. It also places constraints on the API. Client and
servers will have to be notified of the opening of a new stream by
their peer. Instead of orderly closing the control stream, client
and server will have to use orderly connection closure mechanisms and
manage the potential loss of data if closing on one end conflicts
with opening of a stream on the other end.

3.5. No Specific Middlebox Bypass Mechanism

Being different from HTTP/QUIC is a design choice. The advantage is
that the mapping can be defined for minimal overhead and maximum
performance. The downside is that the difference can be noted by
firewalls and middleboxes. There may be environments in which HTTP/
QUIC will be allowed, but DNS/QUIC will be disallowed and blocked by
these middle boxes.

It is recognized that this might be a problem, but there is currently
no indication on how widespread that problem might be. The problem

might be acute enough that the only realistic solution would be to
communicate with independent recursive resolvers using DNS/HTTPS, or
maybe DNS/HTTP/QUIC. Or the problem might be rare enough and the
performance gains significant enough that the appropriate solution
would be to use DNS/QUIC most of the time, and fall back to DNS/HTTPS
some of the time. Measurements and experimentation will inform that
decision.

It may indeed turn out that the complexity and overhead concerns are
negligible compared to the potential advantages of DNS/HTTPS, such as
integration with web services or firewall traversal, and that DNS/
QUIC does not provide sufficient performance gains to justify a new
protocol. We will evaluate that once implementations are available
and can be compared. In the meanwhile, we believe that a clean
design is most likely to inform the QUIC development, as explained in
Section 3.4.

4. Specifications

4.1. Connection Establishment

DNS/QUIC connections are established as described in
[I-D.ietf-quic-transport]. During connection establishment, DNS/QUIC
support is indicated by selecting the ALPN token "dq" in the crypto
handshake.

4.1.1. Draft Version Identification

RFC Editor's Note: Please remove this section prior to publication
of a final version of this document.

Only implementations of the final, published RFC can identify
themselves as "dq". Until such an RFC exists, implementations MUST
NOT identify themselves using this string.

Implementations of draft versions of the protocol MUST add the string
"-" and the corresponding draft number to the identifier. For
example, draft-huitema-quic-dnsoquic-01 is identified using the
string "dq-h01".

4.1.2. Port Selection

By default, a DNS server that supports DNS/QUIC MUST listen for and
accept QUIC connections on the dedicated UDP port TBD (number to be
defined in Section 8), unless it has mutual agreement with its
clients to use a port other than TBD for DNS/QUIC. In order to use a
port other than TBD, both clients and servers would need a
configuration option in their software.

By default, a DNS client desiring to use DNS/QUIC with a particular
server MUST establish a QUIC connection to UDP port TBD on the
server, unless it has mutual agreement with its server to use a port
other than port TBD for DNS/QUIC. Such another port MUST NOT be port
53 or port 853. This recommendation against use of port 53 for DNS/
QUIC is to avoid confusion between DNS/QUIC and DNS/UDP as specified
in [RFC1035]. Similarly, using port 853 would cause confusion
between DNS/QUIC and DNS/DTLS as specified in [RFC8094].

4.2. Stream Mapping and Usage

The mapping of DNS traffic over QUIC streams takes advantage of the
QUIC stream features detailed in Section 10 of
[I-D.ietf-quic-transport].

The stub to resolver DNS traffic follows a simple pattern in which
the client sends a query, and the server provides a response. This
design specifies that for each subsequent query on a QUIC connection
the client MUST select the next available client-initiated
bidirectional stream, in conformance with [I-D.ietf-quic-transport].

The client MUST send the DNS query over the selected stream, and MUST
indicate through the STREAM FIN mechanism that no further data will
be sent on that stream.

The server MUST send the response on the same stream, and MUST
indicate through the STREAM FIN mechanism that no further data will
be sent on that stream.

Therefore, a single client initiated DNS transaction consumes a
single stream. This means that the client's first query occurs on
QUIC stream 4, the second on 8, and so on.

4.2.1. Server Initiated Transactions

There are planned traffic patterns in which a server sends
unsolicited queries to a client, such as for example PUSH messages
defined in [I-D.ietf-dnssd-push]. These occur when a client
subscribes to changes for a particular DNS RRset or resource record
type. When a PUSH server wishes to send such updates it MUST select
the next available server initiated bidirectional stream, in
conformance with [I-D.ietf-quic-transport].

The server MUST send the DNS query over the selected stream, and MUST
indicate through the STREAM FIN mechanism that no further data will
be sent on that stream.

The client MUST send the response on the same stream, and MUST
indicate through the STREAM FIN mechanism that no further data will
be sent on that stream.

Therefore a single server initiated DNS transaction consumes a single
stream. This means that the servers's first query occurs on QUIC
stream 1, the second on 5, and so on.

4.2.2. Stream Reset

Stream transmission may be abandoned by either party, using the
stream "reset" facility. A stream reset indicates that one party is
unwilling to continue processing the transaction associated with the
stream. The corresponding transaction MUST be abandoned. A Server
Failure (SERVFAIL, [RFC1035]) SHOULD be notified to the initiator of
the transaction.

4.3. Closing the DNS/QUIC Connection

QUIC connections are closed using the CONNECTION_CLOSE mechanisms
specified in [I-D.ietf-quic-transport]. Connections can be closed at
the initiative of either the client or the server (also see
Section 5.6.2). The party initiating the connection closure SHOULD
use the QUIC GOAWAY mechanism to initiate a graceful shutdown of a
connection.

The transactions corresponding to stream number higher than indicated
in the GO AWAY frames MUST be considered failed. Similarly, if
streams are still open when the CONNECTION_CLOSE is received, the
corresponding transactions MUST be considered failed. In both cases,
a Server Failure (SERVFAIL, [RFC1035]) SHOULD be notified to the
initiator of the transaction.

4.4. Connection Resume and 0-RTT

A stub resolver MAY take advantage of the connection resume
mechanisms supported by QUIC transport [I-D.ietf-quic-transport] and
QUIC TLS [I-D.ietf-quic-tls]. Stub resolvers SHOULD consider
potential privacy issues associated with session resume before
deciding to use this mechanism. These privacy issues are detailed in
Section 7.2.

When resuming a session, a stub resolver MAY take advantage of the
0-RTT mechanism supported by QUIC. The 0-RTT mechanism MUST NOT be
used to send data that is not "replayable" transactions. For
example, a stub resolver MAY transmit a Query as 0-RTT, but MUST NOT
transmit an Update.

5. Implementation Requirements

5.1. Authentication

For the stub to recursive resolver scenario, the authentication
requirements are the same as described in [RFC7858] and
[I-D.ietf-dprive-dtls-and-tls-profiles]. There is no need to
authenticate the client's identity in either scenario.

5.2. Fall Back to Other Protocols on Connection Failure

If the establishment of the DNS/QUIC connection fails, clients SHOULD
attempt to fall back to DNS/TLS and then potentially clear text, as
specified in [RFC7858] and [I-D.ietf-dprive-dtls-and-tls-profiles],
depending on their privacy profile.

DNS clients SHOULD remember server IP addresses that don't support
DNS/QUIC, including timeouts, connection refusals, and QUIC handshake
failures, and not request DNS/QUIC from them for a reasonable period
(such as one hour per server). DNS clients following an out-of-band
key-pinned privacy profile ([RFC7858]) MAY be more aggressive about
retrying DNS/QUIC connection failures.

5.3. Response Sizes

DNS/QUIC does not suffer from the limitation on the size of responses
that can be delivered as DNS/UDP [RFC1035] does, since large
responses will be sent in separate STREAM frames in separate packets.

QUESTION: However, this raises a new issue because the responses sent
over QUIC can be significantly larger than those sent over TCP
(65,635 bytes). According to [I-D.ietf-quic-transport] "The largest
offset delivered on a stream - the sum of the re-constructed offset
and data length - MUST be less than 2^{62}". Should a specific limit
be applied for DNS/QUIC responses or not?

5.4. DNS Message IDs

When sending multiple queries over a QUIC connection, clients MUST
NOT reuse the DNS Message ID of an in-flight query on that connection
in order to avoid Message ID collisions.

Clients MUST match responses to outstanding queries using the STREAM
ID and Message ID and if the response contains a question section,
the client MUST match the QNAME, QCLASS, and QTYPE fields. Failure
to match is a DNS/QUIC protocol error. Clients observing such errors
SHOULD close the connection immediately, indicating the application

specific error code 0x00000001. The client should also mark the
server as inappropriate for future use of DNS/QUIC.

5.5. Padding

There are mechanisms specified for both padding individual DNS
messages [RFC7830], [I-D.ietf-dprive-padding-policy] and padding
within QUIC packets (see Section 8.6 of [I-D.ietf-quic-transport]),
which may contain multiple frames.

Implementations SHOULD NOT use DNS options for padding individual DNS
messages, because QUIC transport MAY transmit multiple STREAM frames
containing separate DNS messages in a single QUIC packet. Instead,
implementations SHOULD use QUIC PADDING frames to align the packet
length to a small set of fixed sizes, aligned with the
recommendations of [I-D.ietf-dprive-padding-policy].

5.6. Connection Handling

[RFC7766] provides updated guidance on DNS/TCP much of which is
applicable to DNS/QUIC. This section attempts to specify how those
considerations apply to DNS/QUIC.

5.6.1. Connection Reuse

Historic implementations of DNS stub resolvers are known to open and
close TCP connections for each DNS query. To avoid excess QUIC
connections, each with a single query, clients SHOULD reuse a single
QUIC connection to the recursive resolver.

In order to achieve performance on par with UDP, DNS clients SHOULD
send their queries concurrently over the QUIC streams on a QUIC
connection. That is, when a DNS client sends multiple queries to a
server over a QUIC connection, it SHOULD NOT wait for an outstanding
reply before sending the next query.

5.6.2. Connection Close

In order to amortize QUIC and TLS connection setup costs, clients and
servers SHOULD NOT immediately close a QUIC connection after each
response. Instead, clients and servers SHOULD reuse existing QUIC
connections for subsequent queries as long as they have sufficient
resources. In some cases, this means that clients and servers may
need to keep idle connections open for some amount of time.

Under normal operation DNS clients typically initiate connection
closing on idle connections; however, DNS servers can close the
connection if the idle timeout set by local policy is exceeded.

   Also, connections can be closed by either end under unusual
   conditions such as defending against an attack or system failure/
   reboot.

   Clients and servers that keep idle connections open MUST be robust to
   termination of idle connection by either party. As with current DNS
   over TCP, DNS servers MAY close the connection at any time (perhaps
   due to resource constraints). As with current DNS/TCP, clients MUST
   handle abrupt closes and be prepared to reestablish connections and/
   or retry queries.

5.6.3. Idle Timeouts

   Proper management of established and idle connections is important to
   the healthy operation of a DNS server. An implementation of DNS/QUIC
   SHOULD follow best practices for DNS/TCP, as described in [RFC7766].
   Failure to do so may lead to resource exhaustion and denial of
   service.

   This document does not make specific recommendations for timeout
   values on idle connections. Clients and servers should reuse and/or
   close connections depending on the level of available resources.
   Timeouts may be longer during periods of low activity and shorter
   during periods of high activity. Current work in this area may also
   assist DNS/TLS clients and servers in selecting useful timeout values
   [RFC7828] [I-D.ietf-dnsop-session-signal] [TDNS].

   TODO: Clarify what timers (idle timeouts, response timeouts) apply at
   the stream level and at the connection level.

   TODO: QUIC provides an efficient mechanism for resuming connections,
   including the possibility of sending 0-RTT data. Does that change
   the tradeoff? Is it plausible to use shorter timers than specified
   for TCP?

5.7. Flow Control Mechanisms

   Servers MAY use the "maximum stream ID" option of the QUIC transport
   to limit the number of streams opened by the client. This mechanism
   will effectively limit the number of DNS queries that a client can
   send.

6. Security Considerations

   The security considerations of DNS/QUIC should be comparable to those
   of DNS/TLS [RFC7858].

7. Privacy Considerations

   DNS/QUIC is specifically designed to protect the DNS traffic between
   stub and resolver from observations by third parties, and thus
   protect the privacy of queries from the stub. However, the recursive
   resolver has full visibility of the stub's traffic, and could be used
   as an observation point, as discussed in [RFC7626]. These
   considerations do not differ between DNS/TLS and DNS/QUIC and are not
   discussed further here.

   QUIC incorporates the mechanisms of TLS 1.3 [I-D.ietf-tls-tls13] and
   this enables QUIC transmission of "Zero-RTT" data. This can provide
   interesting latency gains, but it raises two concerns:

   1. Adversaries could replay the zero-RTT data and infer its content
       from the behavior of the receiving server.

   2. The zero-RTT mechanism relies on TLS resume, which can provide
       linkability between successive client sessions.

   These issues are developed in Section 7.1 and Section 7.2.

7.1. Privacy Issues With Zero RTT data

   The zero-RTT data can be replayed by adversaries. That data may
   triggers a query by a recursive resolver to an authoritative
   resolvers. Adversaries may be able to pick a time at which the
   recursive resolver outgoing traffic is observable, and thus find out
   what name was queried for in the 0-RTT data.

   This risk is in fact a subset of the general problem of observing the
   behavior of the recursive resolver discussed in [RFC7626]. The
   attack is partially mitigated by reducing the observability of this
   traffic. However, the risk is amplified for 0-RTT data, because the
   attacker might replay it at chosen times, several times.

   The recommendation in [I-D.ietf-tls-tls13] is that the capability to
   use 0-RTT data should be turned off by default, on only enabled if
   the user clearly understands the associated risks.

   QUESTION: Should 0-RTT only be used with Opportunistic profiles (i.e.
   disabled by default for Strict only)?

7.2. Privacy Issues With Session Resume

   The QUIC session resume mechanism reduces the cost of reestablishing
   sessions and enables zero-RTT data. There is a linkability issue
   associated with session resume, if the same resume token is used

several times, but this risk is mitigated by the mechanisms
incorporated in QUIC and in TLS 1.3. With these mechanisms, clients
and servers can cooperate to avoid linkability by third parties.
However, the server will always be able to link the resumed session
to the initial session. This creates a virtual long duration
session. The series of queries in that session can be used by the
server to identify the client.

Enabling the server to link client sessions through session resume is
probably not a large additional risk if the client's connectivity did
not change between the sessions, since the two sessions can probably
be correlated by comparing the IP addresses. On the other hand, if
the addresses did change, the client SHOULD consider whether the
linkability risk exceeds the privacy benefits. This evaluation will
obviously depend on the level of trust between stub and recursive.

7.3. Traffic Analysis

Even though QUIC packets are encrypted, adversaries can gain
information from observing packet lengths, in both queries and
responses, as well as packet timing. Many DNS requests are emitted
by web browsers. Loading a specific web page may require resolving
dozen of DNS names. If an application adopts a simple mapping of one
query or response per packet, or "one QUIC STREAM frame per packet",
then the succession of packet lengths may provide enough information
to identify the requested site.

Implementations SHOULD use the mechanisms defined in Section 5.5 to
mitigate this attack.

8. IANA Considerations

8.1. Registration of DNS/QUIC Identification String

This document creates a new registration for the identification of
DNS/QUIC in the "Application Layer Protocol Negotiation (ALPN)
Protocol IDs" registry established in [RFC7301].

The "dq" string identifies DNS/QUIC:

Protocol: DNS/QUIC

Identification Sequence: 0x64 0x71 ("dq")

Specification: This document

8.2. Reservation of Dedicated Port

IANA is required to add the following value to the "Service Name and Transport Protocol Port Number Registry" in the System Range. The registry for that range requires IETF Review or IESG Approval [RFC6335], and such a review was requested using the early allocation process [RFC7120] for the well-known UDP port in this document. Since port 853 is reserved for 'DNS query-response protocol run over TLS' consideration is requested for reserving port TBD for 'DNS query-response
protocol run over QUIC'.

    Service Name   domain-s

    Transport Protocol(s) TCP/UDP

    Assignee   IESG

    Contact   IETF Chair

    Description   DNS query-response protocol run over QUIC

    Reference   This document

8.2.1. Port number 784 for experimentations

RFC Editor's Note: Please remove this section prior to publication of a final version of this document.

Early experiments MAY use port 784. This port is marked in the IANA registry as unassigned.

9. Acknowledgements

This document liberally borrows text from [I-D.ietf-quic-http] edited by Mike Bishop, and from [RFC7858] authored by Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul Hoffman.

The privacy issue with 0-RTT data and session resume were analyzed by Daniel Kahn Gillmor (DKG) in a message to the IETF "DPRIV" working group [DNS0RTT].

Thanks to our wide cast of supporters.

10. References

10.1. Normative References

[I-D.ietf-dprive-dtls-and-tls-profiles]
          Dickinson, S., Gillmor, D., and T. Reddy, "Usage and
          (D)TLS Profiles for DNS-over-(D)TLS", draft-ietf-dprive-
          dtls-and-tls-profiles-11 (work in progress), September
          2017.

[I-D.ietf-quic-tls]
          Thomson, M. and S. Turner, "Using Transport Layer Security
          (TLS) to Secure QUIC", draft-ietf-quic-tls-08 (work in
          progress), December 2017.

[I-D.ietf-quic-transport]
          Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed
          and Secure Transport", draft-ietf-quic-transport-08 (work
          in progress), December 2017.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities",
          STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987,
          <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and
          specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
          November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119,
          DOI 10.17487/RFC2119, March 1997,
          <https://www.rfc-editor.org/info/rfc2119>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan,
          "Transport Layer Security (TLS) Application-Layer Protocol
          Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301,
          July 2014, <https://www.rfc-editor.org/info/rfc7301>.

10.2. Informative References

[DNS0RTT] Kahn Gillmor, D., "DNS + 0-RTT", Message to DNS-Privacy WG
          mailing list, April 2016, <https://www.ietf.org/mail-
          archive/web/dns-privacy/current/msg01276.html>.

[I-D.bortzmeyer-dprive-step-2]
          Bortzmeyer, S., "Next step for DPRIVE: resolver-to-auth
          link", draft-bortzmeyer-dprive-step-2-05 (work in
          progress), December 2016.

   [I-D.ietf-dnsop-session-signal]
              Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S.,
              Mankin, A., and T. Pusateri, "DNS Stateful Operations",
              draft-ietf-dnsop-session-signal-04 (work in progress),
              September 2017.

   [I-D.ietf-dnssd-push]
              Pusateri, T. and S. Cheshire, "DNS Push Notifications",
              draft-ietf-dnssd-push-13 (work in progress), October 2017.

   [I-D.ietf-doh-dns-over-https]
              Hoffman, P. and P. McManus, "DNS Queries over HTTPS",
              draft-ietf-doh-dns-over-https-02 (work in progress),
              November 2017.

   [I-D.ietf-dprive-padding-policy]
              Mayrhofer, A., "Padding Policy for EDNS(0)", draft-ietf-
              dprive-padding-policy-02 (work in progress), September
              2017.

   [I-D.ietf-quic-http]
              Bishop, M., "Hypertext Transfer Protocol (HTTP) over
              QUIC", draft-ietf-quic-http-08 (work in progress),
              December 2017.

   [I-D.ietf-quic-recovery]
              Iyengar, J. and I. Swett, "QUIC Loss Detection and
              Congestion Control", draft-ietf-quic-recovery-08 (work in
              progress), December 2017.

   [I-D.ietf-tls-tls13]
              Rescorla, E., "The Transport Layer Security (TLS) Protocol
              Version 1.3", draft-ietf-tls-tls13-22 (work in progress),
              November 2017.

   [RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol
              (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010,
              <https://www.rfc-editor.org/info/rfc5936>.

   [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S.
              Cheshire, "Internet Assigned Numbers Authority (IANA)
              Procedures for the Management of the Service Name and
              Transport Protocol Port Number Registry", BCP 165,
              RFC 6335, DOI 10.17487/RFC6335, August 2011,
              <https://www.rfc-editor.org/info/rfc6335>.

   [RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code
              Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January
              2014, <https://www.rfc-editor.org/info/rfc7120>.

   [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626,
              DOI 10.17487/RFC7626, August 2015,
              <https://www.rfc-editor.org/info/rfc7626>.

   [RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and
              D. Wessels, "DNS Transport over TCP - Implementation
              Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016,
              <https://www.rfc-editor.org/info/rfc7766>.

   [RFC7828] Wouters, P., Abley, J., Dickinson, S., and R. Bellis, "The
              edns-tcp-keepalive EDNS0 Option", RFC 7828,
              DOI 10.17487/RFC7828, April 2016,
              <https://www.rfc-editor.org/info/rfc7828>.

   [RFC7830] Mayrhofer, A., "The EDNS(0) Padding Option", RFC 7830,
              DOI 10.17487/RFC7830, May 2016,
              <https://www.rfc-editor.org/info/rfc7830>.

   [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D.,
              and P. Hoffman, "Specification for DNS over Transport
              Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May
              2016, <https://www.rfc-editor.org/info/rfc7858>.

   [RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram
              Transport Layer Security (DTLS)", RFC 8094,
              DOI 10.17487/RFC8094, February 2017,
              <https://www.rfc-editor.org/info/rfc8094>.

   [TDNS]   Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A.,
              and N. Somaiya, "Connection-Oriented DNS to Improve
              Privacy and Security", 2015 IEEE Symposium on Security and
              Privacy (SP), DOI 10.1109/SP.2015.18,
              <http://dx.doi.org/10.1109/SP.2015.18>.

Authors' Addresses

   Christian Huitema
   Private Octopus Inc.
   Friday Harbor WA 98250
   U.S.A

   Email: huitema@huitema.net

Melinda Shore
Fastly

Email: mshore@fastly.com

Allison Mankin
Salesforce

Email: amankin@salesforce.com

Sara Dickinson
Sinodun IT
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
U.K.

Email: sara@sinodun.com

Jana Iyengar
Google

Email: jri@google.com