Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2021

C. Huitema
Private Octopus Inc.
A. Mankin
Salesforce
S. Dickinson
Sinodun IT

October 20, 2020

Specification of DNS over Dedicated QUIC Connections
draft-ietf-dprive-dnsoquic-01

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 (DoQ) has privacy properties similar to DNS over TLS (DoT) specified in RFC7858, and latency characteristics similar to classic DNS over UDP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 23, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Key Words 4
3. Design Considerations 4
  3.1. Scope is Limited to the Stub to Resolver Scenario 4
  3.2. Provide DNS Privacy 5
  3.3. Design for Minimum Latency 5
  3.4. No Specific Middlebox Bypass Mechanism 6
  3.5. No Server Initiated Transactions 6
4. Specifications 6
  4.1. Connection Establishment 7
    4.1.1. Draft Version Identification 7
    4.1.2. Port Selection 7
  4.2. Stream Mapping and Usage 7
    4.2.1. Transaction Errors 8
  4.3. DoQ Error Codes 8
  4.4. Connection Management 9
  4.5. Connection Resume and 0-RTT 9
  4.6. Message Sizes 10
5. Implementation Requirements 10
  5.1. Authentication 10
  5.2. Fall Back to Other Protocols on Connection Failure . . . 11
  5.3. Address Validation 11
  5.4. DNS Message IDs 11
  5.5. Padding 11
  5.6. Connection Handling 12
    5.6.1. Connection Reuse 12
    5.6.2. Resource Management and Idle Timeout Values 12
  5.7. Processing Queries in Parallel 13
  5.8. Flow Control Mechanisms 13
6. Security Considerations 13
7. Privacy Considerations 13
  7.1. Privacy Issues With 0-RTT data 14
  7.2. Privacy Issues With Session Resume 14
  7.3. Traffic Analysis 15
8. IANA Considerations 15
  8.1. Registration of DoQ Identification String 15
  8.2. Reservation of Dedicated Port 15
    8.2.1. Port number 784 for experimentations 16
9. Acknowledgements 16
10. References 16

  10.1. Normative References 16
  10.2. Informative References 18
Authors' Addresses 19

1. Introduction

Domain Name System (DNS) concepts are specified in "Domain names - concepts and facilities" [RFC1034]. The transmission of DNS queries and responses over UDP and TCP is specified in "Domain names - implementation and specification" [RFC1035]. This document presents a mapping of the DNS protocol over the QUIC transport [I-D.ietf-quic-transport] [I-D.ietf-quic-tls]. DNS over QUIC is referred here as DoQ, in line with the "Terminology for DNS Transports and Location" [I-D.ietf-dnsop-terminology-ter]. The goals of the DoQ mapping are:

1. Provide the same DNS privacy protection as DNS over TLS (DoT) [RFC7858]. This includes an option for the client to authenticate the server by means of an authentication domain name as specified in "Usage Profiles for DNS over TLS and DNS over DTLS" [RFC8310].

2. Provide an improved level of source address validation for DNS servers compared to classic DNS over UDP.

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

4. Explore the characteristics of using QUIC as a DNS transport, versus other solutions like DNS over UDP [RFC1035], DoT [RFC7858], or DNS over HTTPS (DoH) [RFC8484].

In order to achieve these goals, the focus of this document is limited to the "stub to recursive resolver" scenario also addressed by DoT [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 AXFR "DNS Zone Transfer Protocol (AXFR)" [RFC5936] or IXFR "Incremental Zone Transfer in DNS" [RFC1885], as these mechanisms are not relevant to the stub to recursive resolver scenario.

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

3. No attempt to support server initiated transactions, are these are not relevant for the "stub to recursive resolver" scenario, see Section 3.5.

Users interested in zone transfers should continue using TCP based solutions and will also want to take note of work in progress to support "DNS Zone Transfer-over-TLS" [I-D.ietf-dprive-xfr-over-tls].

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 "Hypertext Transfer Protocol Version 3 (HTTP/3)"[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 DoQ. Section 5 presents guidelines on the implementation, usage and deployment of DoQ.

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 BCP 14 [RFC8174].

3. Design Considerations

This section and its subsection present the design guidelines that were used for DoQ. 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 DoT [RFC7858] and "Usage Profiles for DNS over TLS and DNS over DTLS" [RFC8310].

QUESTION: Should this document specify any aspects of configuration of discoverability differently to DoT?

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 DoT [I-D.ietf-dprive-phase2-requirements]. 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 AXFR 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 using DNS over TCP. There is no attempt to support either AXFR or IXFR in this proposed mapping of DNS to QUIC.

3.2. Provide DNS Privacy

DoT [RFC7858] defines how to mitigate some of the issues described in "DNS Privacy Considerations" [RFC7626] by specifying how to transmit DNS messages over TLS. The "Usage Profiles for DNS over TLS and DNS over DTLS" [RFC8310] specify Strict and Opportunistic Usage Profiles for DoT including how stub resolvers can authenticate recursive resolvers.

QUIC connection setup includes the negotiation of security parameters using TLS, as specified in "Using TLS to Secure QUIC" [I-D.ietf-quic-tls], enabling encryption of the QUIC transport. Transmitting DNS messages over QUIC will provide essentially the same privacy protections as DoT [RFC7858] including Strict and Opportunistic Usage Profiles [RFC8310]. 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 "QUIC Loss Detection and Congestion Control" [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. Fast resumption of QUIC connections to manage the disconnect-on-idle feature of QUIC without incurring retransmission time-outs.

4. Mapping of each DNS Query/Response transaction to a separate stream, to mitigate head-of-line blocking. This enables servers to respond to queries "out of order". It also enables clients to process responses as soon as they arrive, without having to wait for in order delivery of responses previously posted by the server.

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

3.4. No Specific Middlebox Bypass Mechanism

The mapping of DNS over QUIC is defined for minimal overhead and maximum performance. This means a different traffic profile than HTTP3 over QUIC. This difference can be noted by firewalls and middleboxes. There may be environments in which HTTP3 over QUIC will be able to pass through, but DoQ will be blocked by these middle boxes.

3.5. No Server Initiated Transactions

As stated in Section 1, this document does not specify support for server initiated transactions because these are not relevant for the "stub to recursive resolver" scenario. Note that "DNS Stateful Operations" (DSO) [RFC8490] are only applicable for DNS over TCP and DNS over TLS. DSO is not applicable to DNS over HTTP since HTTP has its own mechanism for managing sessions, and this is incompatible with the DSO; the same is true for DNS over QUIC.

4. Specifications

4.1. Connection Establishment

DoQ connections are established as described in the QUIC transport specification [I-D.ietf-quic-transport]. During connection establishment, DoQ support is indicated by selecting the ALPN token "doq" 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 "doq". 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-ietf-dprive-dnsoquic-00 is identified using the string "doq-i00".

4.1.2. Port Selection

By default, a DNS server that supports DoQ 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 DoQ. 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 DoQ 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 DoQ. Such another port MUST NOT be port 53 or port 853. This recommendation against use of port 53 for DoQ is to avoid confusion between DoQ and the use of DNS over UDP [RFC1035]. Similarly, using port 853 would cause confusion between DoQ and DNS over DTLS [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 2 of the QUIC transport specification [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 the QUIC transport specification [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 0, the second on 4, and so on.

4.2.1. Transaction Errors

Peers normally complete transactions by sending a DNS response on the transaction's stream, including cases where the DNS response indicates a DNS error. For example, a Server Failure (SERVFAIL, [RFC1035]) SHOULD be notified to the initiator of the transaction by sending back a response with the Response Code set to SERVFAIL.

If a peer is incapable of sending a DNS response due to an internal error, it may issue a QUIC Stream Reset with error code DOQ_INTERNAL_ERROR. The corresponding transaction MUST be abandoned.

4.3. DoQ Error Codes

The following error codes are defined for use when abruptly terminating streams, aborting reading of streams, or immediately closing connections:

DOQ_NO_ERROR (0x00): No error. This is used when the connection or stream needs to be closed, but there is no error to signal.

DOQ_INTERNAL_ERROR (0x01): The DoQ implementation encountered an internal error and is incapable of pursuing the transaction or the connection.

DOQ_TRANSPORT_PARAMETER_ERROR (0x02): One or some of the transport parameters proposed by the peer are not acceptable.

4.4. Connection Management

Section 10 of the QUIC transport specification [I-D.ietf-quic-transport] specifies that connections can be closed in three ways:

- idle timeout

- immediate close

- stateless reset

Clients and servers implementing DNS over QUIC SHOULD negotiate use of the idle timeout. Closing on idle timeout is done without any packet exchange, which minimizes protocol overhead. Per section 10.2 of the QUIC transport specification, the effective value of the idle timeout is computed as the minimum of the values advertised by the two endpoints. Practical considerations on setting the idle timeout are discussed in Section 5.6.2.

Clients SHOULD monitor the idle time incurred on their connection to the server, defined by the time spent since the last packet from the server has been received. When a client prepares to send a new DNS query to the server, it will check whether the idle time is sufficient lower than the idle timer. If it is, the client will send the DNS query over the existing connection. If not, the client will establish a new connection and send the query over that connection.

Clients MAY discard their connection to the server before the idle timeout expires. If they do that, they SHOULD close the connection explicitly, using QUIC's CONNECTION_CLOSE mechanisms, and indicating the Application reason "No Error".

Clients and servers MAY close the connection for a variety of other reasons, indicated using QUIC's CONNECTION_CLOSE. Client and servers that send packets over a connection discarded by their peer MAY receive a stateless reset indication. If a connection fails, all queries in progress over the connection MUST be considered failed, and a Server Failure (SERVFAIL, [RFC1035]) SHOULD be notified to the initiator of the transaction.

4.5. 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.

4.6. Message Sizes

DoQ Queries and Responses are sent on QUIC streams, which in theory can carry up to 2^62 bytes. However, DNS messages are restricted in practice to a maximum size of 65535 bytes. This maximum size is enforced by the use of a two-octet message length field in DNS over TCP [RFC1035] and DNS over TLS [RFC7858], and by the definition of the "application/dns-message" for DNS over HTTP [RFC8484]. DoQ enforces the same restriction.

The maximum size of messages is controlled in QUIC by the transport parameters:

- initial_max_stream_data_bidi_local: when set by the client, specifies the amount of data that servers can send on a "response" stream without waiting for a MAX_STREAM_DATA frame.
- initial_max_stream_data_bidi_remote: when set by the server, specifies the amount of data that clients can send on a "query" stream without waiting for a MAX_STREAM_DATA frame.

Clients and servers MUST set these two parameters to the value 65535. If they receive a different value, they SHOULD close the QUIC connection with an application error "Invalid Parameter".

The Extension Mechanisms for DNS (EDNS) [RFC6891] allow peers to specify the UDP message size. This parameter is ignored by DoQ. DoQ implementations always assume that the maximum message size is 65535 bytes.

5. Implementation Requirements

5.1. Authentication

For the stub to recursive resolver scenario, the authentication requirements are the same as described in DoT [RFC7858] and "Usage Profiles for DNS over TLS and DNS over DTLS" [RFC8310]. 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 DoQ connection fails, clients SHOULD attempt to fall back to DoT and then potentially clear text, as specified in DoT [RFC7858] and "Usage Profiles for DNS over TLS and DNS over DTLS" [RFC8310], depending on their privacy profile.

DNS clients SHOULD remember server IP addresses that don't support DoQ, including timeouts, connection refusals, and QUIC handshake failures, and not request DoQ 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 DoQ connection failures.

5.3. Address Validation

Section 8 of the QUIC transport specification [I-D.ietf-quic-transport] defines Address Validation procedures to avoid servers being used in address amplification attacks. DoQ implementations MUST conform to this specification, which limits the worst case amplification to a factor 3.

DoQ implementations SHOULD consider configuring servers to use the Address Validation using Retry Packets procedure defined in section 8.1.2 of the QUIC transport specification [I-D.ietf-quic-transport]). This procedure imposes a 1-RTT delay for verifying the return routability of the source address of a client, similar to the DNS Cookies mechanism [RFC7873].

DoQ implementations that configure Address Validation using Retry Packets SHOULD implement the Address Validation for Future Connections procedure defined in section 8.1.3 of the QUIC transport specification [I-D.ietf-quic-transport]). This defines how servers can send NEW TOKEN frames to clients after the client address is validated, in order to avoid the 1-RTT penalty during subsequent connections by the client from the same address.

5.4. DNS Message IDs

When sending queries over a QUIC connection, the DNS Message ID MUST be set to zero.

5.5. Padding

There are mechanisms specified for padding individual DNS messages in "The EDNS(0) Padding Option" [RFC7830] and for padding within QUIC packets (see Section 8.6 of the QUIC transport specification [I-D.ietf-quic-transport]).

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 the "Padding Policies for Extension Mechanisms for DNS (EDNS(0))" [RFC8467].

5.6. Connection Handling

"DNS Transport over TCP - Implementation Requirements" [RFC7766] provides updated guidance on DNS over TCP, some of which is applicable to DoQ. This section attempts to specify which and how those considerations apply to DoQ.

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. Resource Management and Idle Timeout Values

Proper management of established and idle connections is important to the healthy operation of a DNS server. An implementation of DoQ SHOULD follow best practices similar to those specified for DNS over TCP [RFC7766], in particular with regard to:

- Concurrent Connections (Section 6.2.2)

- Security Considerations (Section 10)

Failure to do so may lead to resource exhaustion and denial of service.

Clients that want to maintain long duration DoQ connections SHOULD use the idle timeout mechanisms defined in Section 10.2 of the QUIC transport specification [I-D.ietf-quic-transport]. Clients and servers MUST NOT send the edns-tcp-keepalive EDNS(0) Option [RFC7828] in any messages sent on a DoQ connection (because it is specific to the use of TCP/TLS as a transport). If any message sent on a DoQ

connection contains an edns-tcp-keepalive EDNS(0) Option, this is a fatal error and the recipient of the defective message MUST forcibly abort the connection immediately.

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.

Clients that are willing to use QUIC's 0-RTT mechanism can reestablish connections and send transactions on the new connection with minimal delay overhead. These clients MAY chose low values of the idle timer.

5.7. Processing Queries in Parallel

As specified in Section 7 of "DNS Transport over TCP - Implementation Requirements" [RFC7766], resolvers are RECOMMENDED to support the preparing of responses in parallel and sending them out of order. In DoQ, they do that by sending responses on their specific stream as soon as possible, without waiting for availability of responses for previously opened streams.

5.8. Flow Control Mechanisms

Servers and Clients manage flow control as specified in QUIC.

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 on a single DoQ connection.

6. Security Considerations

The security considerations of DoQ should be comparable to those of DoT [RFC7858].

7. Privacy Considerations

DoQ is specifically designed to protect the DNS traffic between stub and resolver from observations by third parties, and thus protect the privacy of queries sent by 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 the revision of "DNS Privacy Considerations" [I-D.ietf-dprive-rfc7626-bis]. These considerations do not differ between DoT and DoQ and are not discussed further here.

QUIC incorporates the mechanisms of TLS 1.3 [RFC8446] and this enables QUIC transmission of "0-RTT" data. This can provide interesting latency gains, but it raises two concerns:

1. Adversaries could replay the 0-RTT data and infer its content from the behavior of the receiving server.

2. The 0-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 0-RTT data

The 0-RTT data can be replayed by adversaries. That data may trigger queries by a recursive resolver to 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 "DNS Privacy Considerations" [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 for TLS 1.3 [RFC8446] is that the capability to use 0-RTT data should be turned off by default, and 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 re-establishing sessions and enables 0-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 performance 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 DoQ Identification String

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

The "doq" string identifies DoQ:

Protocol: DoQ

Identification Sequence: 0x64 0x6F 0x71 ("doq")

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 the HTTP-3 specification [I-D.ietf-quic-http] edited by Mike Bishop, and from the DoT specification [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 "DPRIVE" working group [DNS0RTT].

Thanks to Tony Finch for an extensive review of the initial version of this draft.

10. References

10.1. Normative References

[I-D.ietf-dnsop-terminology-ter]
Hoffman, P., "Terminology for DNS Transports and Location", draft-ietf-dnsop-terminology-ter-02 (work in progress), August 2020.

[I-D.ietf-quic-tls]
Thomson, M. and S. Turner, "Using TLS to Secure QUIC", draft-ietf-quic-tls-31 (work in progress), September 2020.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-31 (work in progress), September 2020.

[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>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[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>.

[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>.

[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>.

[RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016, <https://www.rfc-editor.org/info/rfc7873>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <https://www.rfc-editor.org/info/rfc8310>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

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.ietf-dprive-phase2-requirements]
Livingood, J., Mayrhofer, A., and B. Overeinder, "DNS Privacy Requirements for Exchanges between Recursive Resolvers and Authoritative Servers", draft-ietf-dprive-phase2-requirements-01 (work in progress), June 2020.

[I-D.ietf-dprive-rfc7626-bis]
Wicinski, T., "DNS Privacy Considerations", draft-ietf-dprive-rfc7626-bis-07 (work in progress), October 2020.

[I-D.ietf-dprive-xfr-over-tls]
Toorop, W., Dickinson, S., Sahib, S., Aras, P., and A. Mankin, "DNS Zone Transfer-over-TLS", draft-ietf-dprive-xfr-over-tls-02 (work in progress), July 2020.

[I-D.ietf-quic-http]
Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", draft-ietf-quic-http-31 (work in progress), September 2020.

[I-D.ietf-quic-recovery]
Iyengar, J. and I. Swett, "QUIC Loss Detection and Congestion Control", draft-ietf-quic-recovery-31 (work in progress), September 2020.

[RFC1885] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)", RFC 1885, DOI 10.17487/RFC1885, December 1995, <https://www.rfc-editor.org/info/rfc1885>.

[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>.

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

[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>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8467] Mayrhofer, A., "Padding Policies for Extension Mechanisms for DNS (EDNS(0))", RFC 8467, DOI 10.17487/RFC8467, October 2018, <https://www.rfc-editor.org/info/rfc8467>.

[RFC8490] Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S., Lemon, T., and T. Pusateri, "DNS Stateful Operations", RFC 8490, DOI 10.17487/RFC8490, March 2019, <https://www.rfc-editor.org/info/rfc8490>.

Authors' Addresses

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

Email: huitema@huitema.net

Allison Mankin
Salesforce

Email: amankin@salesforce.com

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

Email: sara@sinodun.com