

Domain Name System Operations
Internet-Draft
Updates: [1123](#) (if approved)
Intended status: Best Current Practice
Expires: September 11, 2021

J. Kristoff
DataPlane.org
D. Wessels
Verisign
March 10, 2021

DNS Transport over TCP - Operational Requirements
draft-ietf-dnsop-dns-tcp-requirements-07

Abstract

This document encourages the practice of permitting DNS messages to be carried over TCP on the Internet. This includes both DNS over unencrypted TCP, as well as over an encrypted TLS session. The document also considers the consequences with this form of DNS communication and the potential operational issues that can arise when this best current practice is not upheld.

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 <https://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 September 11, 2021.

Copyright Notice

Copyright (c) 2021 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 (<https://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
1.1.	Requirements Language	4
2.	Background	4
2.1.	Uneven Transport Usage and Preference	4
2.2.	Waiting for Large Messages and Reliability	5
2.3.	EDNS0	5
2.4.	Fragmentation and Truncation	6
2.5.	"Only Zone Transfers Use TCP"	7
3.	DNS over TCP Requirements	7
4.	Network and System Considerations	8
4.1.	Connection Establishment and Admission	8
4.2.	Connection Management	9
4.3.	Connection Termination	10
4.4.	DNS-over-TLS	11
5.	DNS over TCP Filtering Risks	11
5.1.	DNS Wedgie	11
5.2.	DNS Root Zone KSK Rollover	12
6.	Logging and Monitoring	12
7.	Acknowledgments	13
8.	IANA Considerations	13
9.	Security Considerations	13
10.	Privacy Considerations	13
11.	References	14
11.1.	Normative References	14
11.2.	Informative References	14
Appendix A.	Standards Related to DNS Transport over TCP	20
A.1.	IETF RFC 1035 - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION	20
A.2.	IETF RFC 1536 - Common DNS Implementation Errors and Suggested Fixes	21
A.3.	IETF RFC 1995 - Incremental Zone Transfer in DNS	21
A.4.	IETF RFC 1996 - A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)	21
A.5.	IETF RFC 2181 - Clarifications to the DNS Specification	21
A.6.	IETF RFC 2694 - DNS extensions to Network Address Translators (DNS_ALG)	21
A.7.	IETF RFC 3225 - Indicating Resolver Support of DNSSEC	21
A.8.	IETF RFC 3326 - DNSSEC and IPv6 A6 aware server/resolver message size requirements	22
A.9.	IETF RFC 4472 - Operational Considerations and Issues with IPv6 DNS	22
A.10.	IETF RFC 5452 - Measures for Making DNS More Resilient against Forged Answers	22

A.11.	IETF RFC 5507 - Design Choices When Expanding the DNS . . .	22
A.12.	IETF RFC 5625 - DNS Proxy Implementation Guidelines . . .	22
A.13.	IETF RFC 5936 - DNS Zone Transfer Protocol (AXFR)	23
A.14.	IETF RFC 5966 - DNS Transport over TCP - Implementation Requirements	23
A.15.	IETF RFC 6304 - AS112 Nameserver Operations	23
A.16.	IETF RFC 6762 - Multicast DNS	23
A.17.	IETF RFC 6891 - Extension Mechanisms for DNS (EDNS(0)) .	23
A.18.	IETF RFC 6950 - Architectural Considerations on Application Features in the DNS	23
A.19.	IETF RFC 7477 - Child-to-Parent Synchronization in DNS .	24
A.20.	IETF RFC 7720 - DNS Root Name Service Protocol and Deployment Requirements	24
A.21.	IETF RFC 7766 - DNS Transport over TCP - Implementation Requirements	24
A.22.	IETF RFC 7828 - The edns-tcp-keepalive EDNS0 Option . . .	24
A.23.	IETF RFC 7858 - Specification for DNS over Transport Layer Security (TLS)	24
A.24.	IETF RFC 7873 - Domain Name System (DNS) Cookies	24
A.25.	IETF RFC 7901 - CHAIN Query Requests in DNS	25
A.26.	IETF RFC 8027 - DNSSEC Roadblock Avoidance	25
A.27.	IETF RFC 8094 - DNS over Datagram Transport Layer Security (DTLS)	25
A.28.	IETF RFC 8162 - Using Secure DNS to Associate Certificates with Domain Names for S/MIME	25
A.29.	IETF RFC 8324 - DNS Privacy, Authorization, Special Uses, Encoding, Characters, Matching, and Root Structure: Time for Another Look?	26
A.30.	IETF RFC 8467 - Padding Policies for Extension Mechanisms for DNS (EDNS(0))	26
A.31.	IETF RFC 8483 - Yeti DNS Testbed	26
A.32.	IETF RFC 8484 - DNS Queries over HTTPS (DoH)	26
A.33.	IETF RFC 8490 - DNS Stateful Operations	26
A.34.	IETF RFC 8501 - Reverse DNS in IPv6 for Internet Service Providers	26
	Authors' Addresses	27

[1.](#) Introduction

DNS messages may be delivered using UDP or TCP communications. While most DNS transactions are carried over UDP, some operators have been led to believe that any DNS over TCP traffic is unwanted or unnecessary for general DNS operation. When DNS over TCP has been restricted, a variety of communication failures and debugging challenges often arise. As DNS and new naming system features have evolved, TCP as a transport has become increasingly important for the correct and safe operation of an Internet DNS. Reflecting modern usage, the DNS standards were recently updated to declare support for

TCP is now a required part of the DNS implementation specifications [RFC7766]. This document is the formal requirements equivalent for the operational community, encouraging system administrators, network engineers, and security staff to ensure DNS over TCP communications support is on par with DNS over UDP communications.

1.1. Requirements Language

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 [RFC 2119](#) [RFC2119].

2. Background

The curious state of disagreement in operational best practices and guidance for DNS transport protocols derives from conflicting messages operators have gotten from other operators, implementors, and even the IETF. Sometimes these mixed signals have been explicit, on other occasions they have suspiciously implicit. This section presents an interpretation of the storied and conflicting history that led to this document.

2.1. Uneven Transport Usage and Preference

In the original suite of DNS specifications, [RFC1034] and [RFC1035] clearly specified that DNS messages could be carried in either UDP or TCP, but they also stated a preference for UDP as the best transport for queries in the general case. As stated in [RFC1035]:

"While virtual circuits can be used for any DNS activity, datagrams are preferred for queries due to their lower overhead and better performance."

Another early, important, and influential document, [RFC1123], marked the preference for a transport protocol more explicitly:

"DNS resolvers and recursive servers MUST support UDP, and SHOULD support TCP, for sending (non-zone-transfer) queries."

and further stipulated:

"A name server MAY limit the resources it devotes to TCP queries, but it SHOULD NOT refuse to service a TCP query just because it would have succeeded with UDP."

Culminating in [RFC1536], DNS over TCP came to be associated primarily with the zone transfer mechanism, while most DNS queries and responses were seen as the dominion of UDP.

2.2. Waiting for Large Messages and Reliability

In the original specifications, the maximum DNS over UDP message size was enshrined at 512 bytes. However, even while [\[RFC1123\]](#) preferred UDP for non-zone transfer queries, it foresaw DNS over TCP becoming more popular in the future to overcome this limitation:

"[...] it is also clear that some new DNS record types defined in the future will contain information exceeding the 512 byte limit that applies to UDP, and hence will require TCP.

At least two new, widely anticipated developments were set to elevate the need for DNS over TCP transactions. The first was dynamic updates defined in [\[RFC2136\]](#) and the second was the set of extensions collectively known as DNSSEC originally specified in [\[RFC2541\]](#). The former suggested "requestors who require an accurate response code must use TCP," while the latter warned "... larger keys increase the size of KEY and SIG RRs. This increases the chance of DNS UDP packet overflow and the possible necessity for using higher overhead TCP in responses."

Yet, defying some expectations, DNS over TCP remained little-used in real traffic across the Internet around this time. Dynamic updates saw little deployment between autonomous networks. Around the time DNSSEC was first defined, another new feature helped solidify UDP transport dominance for message transactions.

2.3. EDNS0

In 1999 the IETF published the Extension Mechanisms for DNS (EDNS0) in [\[RFC2671\]](#) (superseded in 2013 by an update in [\[RFC6891\]](#)). This document standardized a way for communicating DNS nodes to perform rudimentary capabilities negotiation. One such capability written into the base specification and present in every EDNS0-compatible message is the value of the maximum UDP payload size the sender can support. This unsigned 16-bit field specifies, in bytes, the maximum (possibly fragmented) DNS message size a node is capable of receiving. In practice, typical values are a subset of the 512- to 4096-byte range. EDNS0 became widely deployed over the next several years and numerous surveys ([\[CASTRO2010\]](#), [\[NETALYZR\]](#)) have shown many systems currently support larger UDP MTUs with EDNS0.

The natural effect of EDNS0 deployment meant DNS messages larger than 512 bytes would be less reliant on TCP than they might otherwise have been. While a non-negligible population of DNS systems lacked EDNS0 or fell back to TCP when necessary, DNS over TCP transactions remained a very small fraction of overall DNS traffic [\[VERISIGN\]](#).

2.4. Fragmentation and Truncation

Although EDNS0 provides a way for endpoints to signal support for DNS messages exceeding 512 bytes, the realities of a diverse and inconsistently deployed Internet may result in some large messages being unable to reach their destination. Any IP datagram whose size exceeds the MTU of a link it transits will be fragmented and then reassembled by the receiving host. Unfortunately, it is not uncommon for middleboxes and firewalls to block IP fragments. If one or more fragments do not arrive, the application does not receive the message and the request times out.

For IPv4-connected hosts, the de-facto MTU is often the Ethernet payload size of 1500 bytes. This means that the largest unfragmented UDP DNS message that can be sent over IPv4 is likely 1472 bytes. For IPv6, the situation is a little more complicated. First, IPv6 headers are 40 bytes (versus 20 without options in IPv4). Second, it seems as though some people have mis-interpreted IPv6's required minimum MTU of 1280 as a required maximum. Third, fragmentation in IPv6 can only be done by the host originating the datagram. The need to fragment is conveyed in an ICMPv6 "packet too big" message. The originating host indicates a fragmented datagram with IPv6 extension headers. Unfortunately, it is quite common for both ICMPv6 and IPv6 extension headers to be blocked by middleboxes. According to [\[HUSTON\]](#) some 35% of IPv6-capable recursive resolvers were unable to receive a fragmented IPv6 packet.

The practical consequence of all this is that DNS requestors must be prepared to retry queries with different EDNS0 maximum message size values. Administrators of BIND are likely to be familiar with seeing "success resolving ... after reducing the advertised EDNS0 UDP packet size to 512 octets" messages in their system logs.

Often, reducing the EDNS0 UDP packet size leads to a successful response. That is, the necessary data fits within the smaller message size. However, when the data does not fit, the server sets the truncated flag in its response, indicating the client should retry over TCP to receive the whole response. This is undesirable from the client's point of view because it adds more latency and potentially undesirable from the server's point of view due to the increased resource requirements of TCP.

The issues around fragmentation, truncation, and TCP are driving certain implementation and policy decisions in the DNS. Notably, Cloudflare implemented what it calls "DNSSEC black lies" [\[CLOUDFLARE\]](#) and uses ECDSA algorithms, such that their signed responses fit easily in 512 bytes. The KSK Rollover design team [\[DESIGNTEAM\]](#) spent a lot of time thinking and worrying about response sizes. There is

growing sentiment in the DNSSEC community that RSA key sizes beyond 2048-bits are impractical and that critical infrastructure zones should transition to elliptic curve algorithms to keep response sizes manageable.

More recently, renewed security concerns about fragmented DNS messages ([[AVOID FRAGS](#)], [[FRAG POISON](#)]) are leading implementors to consider lower default EDNS0 UDP payload size values for both queriers and responders.

[2.5.](#) "Only Zone Transfers Use TCP"

Today, the majority of the DNS community expects, or at least has a desire, to see DNS over TCP transactions occur without interference. However there has also been a long-held belief by some operators, particularly for security-related reasons, that DNS over TCP services should be purposely limited or not provided at all [[CHES94](#)], [[DJBDNS](#)]. A popular meme has also held the imagination of some: that DNS over TCP is only ever used for zone transfers and is generally unnecessary otherwise, with filtering all DNS over TCP traffic even described as a best practice.

The position on restricting DNS over TCP had some justification given that historic implementations of DNS nameservers provided very little in the way of TCP connection management (for example see [Section 6.1.2 of \[RFC7766\]](#) for more details). However modern standards and implementations are nearing parity with the more sophisticated TCP management techniques employed by, for example, HTTP(S) servers and load balancers.

[3.](#) DNS over TCP Requirements

An average increase in DNS message size (e.g., due to DNSSEC), the continued development of new DNS features [[Appendix A](#)], and a denial of service mitigation technique [[Section 9](#)] have suggested that DNS over TCP transactions are as important to the correct and safe operation of the Internet DNS as ever, if not more so. Furthermore, there has been serious research that argues connection-oriented DNS transactions may provide security and privacy advantages over UDP transport. [[TDNS](#)] In fact [[RFC7858](#)], a Standards Track document, is just this sort of specification. Therefore, this document makes explicit that it is undesirable for network operators to artificially inhibit DNS over TCP transport.

[Section 6.1.3.2 in \[RFC1123\]](#) is updated: All DNS resolvers and servers MUST support and service both UDP and TCP queries.

- o Authoritative servers **MUST** support and service all TCP queries so that they do not limit the size of responses to what fits in a single UDP packet.
- o Recursive servers (or forwarders) **MUST** support and service all TCP queries so that they do not prevent large responses from a TCP-capable server from reaching its TCP-capable clients.

Regarding the choice of limiting the resources a server devotes to queries, [Section 6.1.3.2 in \[RFC1123\]](#) also says:

"A name server **MAY** limit the resources it devotes to TCP queries, but it **SHOULD NOT** refuse to service a TCP query just because it would have succeeded with UDP."

This requirement is hereby updated: 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. DNS resolver and server operators **MUST** support and provide DNS service over both UDP and TCP transports. Likewise, network operators **MUST** allow DNS service over both UDP and TCP transports. It is acknowledged that DNS over TCP service can pose operational challenges that are not present when running DNS over UDP alone, and vice-versa. However, it is the aim of this document to argue that the potential damage incurred by prohibiting DNS over TCP service is more detrimental to the continued utility and success of the DNS than when its usage is allowed.

4. Network and System Considerations

This section describes measures that systems and applications can take to optimize performance over TCP and to protect themselves from TCP-based resource exhaustion and attacks.

4.1. Connection Establishment and Admission

Resolvers and other DNS clients should be aware that some servers might not be reachable over TCP. For this reason, clients **MAY** want to track and limit the number of TCP connections and connection attempts to a single server. Additionally, DNS clients **MAY** want to enforce a short timeout on unestablished connections, rather than rely on the host operating system's TCP connection timeout, which is often around 60-120 seconds.

The SYN flooding attack is a denial-of-service method affecting hosts that run TCP server processes [[RFC4987](#)]. This attack can be very effective if not mitigated. One of the most effective mitigation techniques is SYN cookies, which allows the server to avoid allocating any state until the successful completion of the three-way handshake.

Services not intended for use by the public Internet, such as most recursive name servers, SHOULD be protected with access controls. Ideally these controls are placed in the network, well before before any unwanted TCP packets can reach the DNS server host or application. If this is not possible, the controls can be placed in the application itself. In some situations (e.g. attacks) it may be necessary to deploy access controls for DNS services that should otherwise be globally reachable.

The FreeBSD, OpenBSD, and NetBSD operating systems have an "accept filter" feature ([\[accept filter\]](#)) that postpones delivery of TCP connections to applications until a complete, valid request has been received. The `dns_accf(9)` filter ensures that a valid DNS message is received. If not, the bogus connection never reaches the application. Applications must be coded and configured to make use of this filter.

Per [[RFC7766](#)], applications and administrators are advised to remember that TCP MAY be used before sending any UDP queries. Networks and applications MUST NOT be configured to refuse TCP queries that were not preceded by a UDP query.

TCP Fast Open [[RFC7413](#)] (TFO) allows TCP clients to shorten the handshake for subsequent connections to the same server. TFO saves one round-trip time in the connection setup. DNS servers SHOULD enable TFO when possible. Furthermore, DNS servers clustered behind a single service address (e.g., anycast or load-balancing), SHOULD use the same TFO server key on all instances.

DNS clients MAY also enable TFO when possible. Currently, on some operating systems it is not implemented or disabled by default. [\[WIKIPEDIA TFO\]](#) describes applications and operating systems that support TFO.

4.2. Connection Management

Since host memory for TCP state is a finite resource, DNS clients and servers MUST actively manage their connections. Applications that do not actively manage their connections can encounter resource exhaustion leading to denial of service. For DNS, as in other protocols, there is a tradeoff between keeping connections open for

potential future use and the need to free up resources for new connections that will arrive.

DNS server software SHOULD provide a configurable limit on the total number of established TCP connections. If the limit is reached, the application is expected to either close existing (idle) connections or refuse new connections. Operators SHOULD ensure the limit is configured appropriately for their particular situation.

DNS server software MAY provide a configurable limit on the number of established connections per source IP address or subnet. This can be used to ensure that a single or small set of users can not consume all TCP resources and deny service to other users. Operators SHOULD ensure this limit is configured appropriately, based on their number of diversity of users.

DNS server software SHOULD provide a configurable timeout for idle TCP connections. For very busy name servers this might be set to a low value, such as a few seconds. For less busy servers it might be set to a higher value, such as tens of seconds. DNS clients and servers SHOULD signal their timeout values using the edns-tcp-keepalive option [[RFC7828](#)].

DNS server software MAY provide a configurable limit on the number of transactions per TCP connection. This document does not offer advice on particular values for such a limit.

Similarly, DNS server software MAY provide a configurable limit on the total duration of a TCP connection. This document does not offer advice on particular values for such a limit.

Since clients may not be aware of server-imposed limits, clients utilizing TCP for DNS need to always be prepared to re-establish connections or otherwise retry outstanding queries.

4.3. Connection Termination

In general, it is preferable for clients to initiate the close of a TCP connection. The TCP peer that initiates a connection close retains the socket in the TIME_WAIT state for some amount of time, possibly a few minutes. On a busy server, the accumulation of many sockets in TIME_WAIT can cause performance problems or even denial of service.

On systems where large numbers of sockets in TIME_WAIT are observed, it may be beneficial to tune the local TCP parameters. For example, the Linux kernel provides a number of "sysctl" parameters related to TIME_WAIT, such as net.ipv4.tcp_fin_timeout, net.ipv4.tcp_tw_recycle,

and `net.ipv4.tcp_tw_reuse`. In extreme cases, implementors and operators of very busy servers may find it necessary to utilize the `SO_LINGER` socket option ([[Stevens](#)] [Section 7.5](#)) with a value of zero so that the server doesn't accumulate `TIME_WAIT` sockets.

[4.4.](#) DNS-over-TLS

DNS messages may be sent over TLS to provide privacy between stubs and recursive resolvers. [[RFC7858](#)] is a standards track document describing how this works. Although DNS-over-TLS utilizes TCP port 853 instead of port 53, this document applies equally well to DNS-over-TLS. Note, however, DNS-over-TLS is currently only defined between stubs and recursives.

The use of TLS places even stronger operational burdens on DNS clients and servers. Cryptographic functions for authentication and encryption require additional processing. Unoptimized connection setup takes two additional round-trips compared to TCP, but can be reduced with Fast TLS connection resumption [[RFC5077](#)] and TLS False Start [[RFC7918](#)].

[5.](#) DNS over TCP Filtering Risks

Networks that filter DNS over TCP risk losing access to significant or important pieces of the DNS namespace. For a variety of reasons a DNS answer may require a DNS over TCP query. This may include large message sizes, lack of EDNS0 support, DDoS mitigation techniques, or perhaps some future capability that is as yet unforeseen will also demand TCP transport.

For example, [[RFC7901](#)] describes a latency-avoiding technique that sends extra data in DNS responses. This makes responses larger and potentially increases the effectiveness of DDoS reflection attacks. The specification mandates the use of TCP or DNS Cookies [[RFC7873](#)].

Even if any or all particular answers have consistently been returned successfully with UDP in the past, this continued behavior cannot be guaranteed when DNS messages are exchanged between autonomous systems. Therefore, filtering of DNS over TCP is considered harmful and contrary to the safe and successful operation of the Internet. This section enumerates some of the known risks known at the time of this writing when networks filter DNS over TCP.

[5.1.](#) DNS Wedgie

Networks that filter DNS over TCP may inadvertently cause problems for third-party resolvers as experienced by [[TOYAMA](#)]. If, for instance, a resolver receives a truncated answer from a server, but

when the resolver resends the query using TCP and the TCP response never arrives, not only will a complete answer be unavailable, but the resolver will incur the full extent of TCP retransmissions and timeouts. This situation might place extreme strain on resolver resources. If the number and frequency of these truncated answers are sufficiently high, the steady-state of lost resources as a result is a "DNS wedgie." A DNS wedgie is generally not easily or completely mitigated by the affected DNS resolver operator.

5.2. DNS Root Zone KSK Rollover

The plans for deploying a new root zone DNSSEC KSK highlighted a potential problem in retrieving the root zone key set [[LEWIS](#)]. During some phases of the KSK rollover process, root zone DNSKEY responses were larger than 1280 bytes, the IPv6 minimum MTU for links carrying IPv6 traffic [[RFC2460](#)]. There was some concern that any DNS server unable to receive large DNS messages over UDP, or any DNS message over TCP, would experience disruption while performing DNSSEC validation.

However, during the year-long postponement of the KSK rollover there were no reported problems that could be attributed to the 1414 octet DNSKEY response when both the old and new keys were published in the zone. Additionally, there were no reported problems during the two month period when the old key was published as revoked and the DNSKEY response was 1425 octets in size [[ROLL YOUR ROOT](#)].

6. Logging and Monitoring

Developers of applications that log or monitor DNS SHOULD NOT ignore TCP due to the perception that it is rarely used or is hard to process. Operators SHOULD ensure that their monitoring and logging applications properly capture DNS message over TCP. Otherwise, attacks, exfiltration attempts, and normal traffic may go undetected.

DNS messages over TCP are in no way guaranteed to arrive in single segments. In fact, a clever attacker might attempt to hide certain messages by forcing them over very small TCP segments. Applications that capture network packets (e.g., with libpcap [[libpcap](#)]) SHOULD be prepared to implement and perform full TCP segment reassembly. dnscap [[dnscap](#)] is an open-source example of a DNS logging program that implements TCP reassembly.

Developers SHOULD also keep in mind connection reuse, query pipelining, and out-of-order responses when building and testing DNS monitoring applications.

As an alternative to packet capture, some DNS server software supports dnstap [[dnstap](#)] as an integrated monitoring protocol intended to facilitate wide-scale DNS monitoring.

7. Acknowledgments

This document was initially motivated by feedback from students who pointed out that they were hearing contradictory information about filtering DNS over TCP messages. Thanks in particular to a teaching colleague, JPL, who perhaps unknowingly encouraged the initial research into the differences between what the community has historically said and did. Thanks to all the NANOG 63 attendees who provided feedback to an early talk on this subject.

The following individuals provided an array of feedback to help improve this document: Piet Barber, Sara Dickinson, Bob Harold, Tatuya Jinmei, and Paul Hoffman, Puneet Sood, Richard Wilhelm. The authors are also indebted to the contributions stemming from discussion in the tcpm working group meeting at IETF 104. Any remaining errors or imperfections are the sole responsibility of the document authors.

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

Ironically, returning truncated DNS over UDP answers in order to induce a client query to switch to DNS over TCP has become a common response to source address spoofed, DNS denial-of-service attacks [[RRL](#)]. Historically, operators have been wary of TCP-based attacks, but in recent years, UDP-based flooding attacks have proven to be the most common protocol attack on the DNS. Nevertheless, a high rate of short-lived DNS transactions over TCP may pose challenges. While many operators have provided DNS over TCP service for many years without duress, past experience is no guarantee of future success.

DNS over TCP is not unlike many other Internet TCP services. TCP threats and many mitigation strategies have been well-documented in a series of documents such as [[RFC4953](#)], [[RFC4987](#)], [[RFC5927](#)], and [[RFC5961](#)].

10. Privacy Considerations

Since DNS over both UDP and TCP use the same underlying message format, the use of one transport instead of the other does change the privacy characteristics of the message content (i.e., the name being

queried). DNS over TLS or DTLS is the recommended way to achieve DNS privacy.

Because TCP is somewhat more complex than UDP, some characteristics of a TCP conversation may enable fingerprinting and tracking that is not possible with UDP. For example, the choice of initial sequence numbers, window size, and options might be able to identify a particular TCP implementation, or even individual hosts behind shared resources such as network address translators (NATs).

11. References

11.1. Normative References

[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>>.

11.2. Informative References

[accept_filter]
FreeBSD, "FreeBSD accept_filter(9)", May 2018, <https://www.freebsd.org/cgi/man.cgi?query=accept_filter>.

[AVOID_FRAGS]
Fujiwara, K., "It's time to consider avoiding IP fragmentation in the DNS", Jul 2019, <<https://blog.apnic.net/2019/07/12/its-time-to-consider-avoiding-ip-fragmentation-in-the-dns/>>.

[CASTRO2010]
Castro, S., Zhang, M., John, W., Wessels, D., and k. claffy, "Understanding and preparing for DNS evolution", 2010.

[CHES94] Cheswick, W. and S. Bellovin, "Firewalls and Internet Security: Repelling the Wily Hacker", 1994.

[CLOUDFLARE]
Grant, D., "Economical With The Truth: Making DNSSEC Answers Cheap", June 2016, <<https://blog.cloudflare.com/black-lies/>>.

[DESIGNTEAM]
Design Team Report, "Root Zone KSK Rollover Plan", December 2015, <<https://www.iana.org/reports/2016/root-ksk-rollover-design-20160307.pdf>>.

- [DJBDNS] D.J. Bernstein, "When are TCP queries sent?", 2002, <<https://cr.yp.to/djbdns/tcp.html#why>>.
- [dnscap] DNS-OARC, "DNSCAP", May 2018, <<https://www.dns-oarc.net/tools/dnscap>>.
- [dnstap] Edmonds, R. and P. Vixie, "dnstap", May 2018, <<https://dnstap.info>>.
- [FRAG_POISON] Herzberg, A. and H. Shulman, "Fragmentation Considered Poisonous", May 2012, <<https://u.cs.biu.ac.il/~herzbea/security/13-03-frag.pdf>>.
- [HUSTON] Huston, G., "Dealing with IPv6 fragmentation in the DNS", August 2017, <<https://blog.apnic.net/2017/08/22/dealing-ipv6-fragmentation-dns/>>.
- [LEWIS] Lewis, E., "2017 DNSSEC KSK Rollover", RIPE 74 Budapest, Hungary, May 2017, <<https://ripe74.ripe.net/presentations/25-RIPE74-lewis-submission.pdf>>.
- [libpcap] Tcpdump/Libpcap, "Tcpdump and Libpcap", May 2018, <<https://www.tcpdump.org>>.
- [NETALYZR] Kreibich, C., Weaver, N., Nechaev, B., and V. Paxson, "Netalyzer: Illuminating The Edge Network", 2010.
- [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>>.
- [RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, [RFC 1123](#), DOI 10.17487/RFC1123, October 1989, <<https://www.rfc-editor.org/info/rfc1123>>.
- [RFC1536] Kumar, A., Postel, J., Neuman, C., Danzig, P., and S. Miller, "Common DNS Implementation Errors and Suggested Fixes", [RFC 1536](#), DOI 10.17487/RFC1536, October 1993, <<https://www.rfc-editor.org/info/rfc1536>>.

- [RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", [RFC 1995](#), DOI 10.17487/RFC1995, August 1996, <<https://www.rfc-editor.org/info/rfc1995>>.
- [RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", [RFC 1996](#), DOI 10.17487/RFC1996, August 1996, <<https://www.rfc-editor.org/info/rfc1996>>.
- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", [RFC 2136](#), DOI 10.17487/RFC2136, April 1997, <<https://www.rfc-editor.org/info/rfc2136>>.
- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", [RFC 2181](#), DOI 10.17487/RFC2181, July 1997, <<https://www.rfc-editor.org/info/rfc2181>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [RFC2541] Eastlake 3rd, D., "DNS Security Operational Considerations", [RFC 2541](#), DOI 10.17487/RFC2541, March 1999, <<https://www.rfc-editor.org/info/rfc2541>>.
- [RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", [RFC 2671](#), DOI 10.17487/RFC2671, August 1999, <<https://www.rfc-editor.org/info/rfc2671>>.
- [RFC2694] Srisuresh, P., Tsirtsis, G., Akkiraju, P., and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)", [RFC 2694](#), DOI 10.17487/RFC2694, September 1999, <<https://www.rfc-editor.org/info/rfc2694>>.
- [RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", [RFC 3225](#), DOI 10.17487/RFC3225, December 2001, <<https://www.rfc-editor.org/info/rfc3225>>.
- [RFC3226] Gudmundsson, O., "DNSSEC and IPv6 A6 aware server/resolver message size requirements", [RFC 3226](#), DOI 10.17487/RFC3226, December 2001, <<https://www.rfc-editor.org/info/rfc3226>>.
- [RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", [RFC 4472](#), DOI 10.17487/RFC4472, April 2006, <<https://www.rfc-editor.org/info/rfc4472>>.

- [RFC4953] Touch, J., "Defending TCP Against Spoofing Attacks", [RFC 4953](#), DOI 10.17487/RFC4953, July 2007, <<https://www.rfc-editor.org/info/rfc4953>>.
- [RFC4987] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", [RFC 4987](#), DOI 10.17487/RFC4987, August 2007, <<https://www.rfc-editor.org/info/rfc4987>>.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", [RFC 5077](#), DOI 10.17487/RFC5077, January 2008, <<https://www.rfc-editor.org/info/rfc5077>>.
- [RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", [RFC 5452](#), DOI 10.17487/RFC5452, January 2009, <<https://www.rfc-editor.org/info/rfc5452>>.
- [RFC5507] IAB, Faltstrom, P., Ed., Austein, R., Ed., and P. Koch, Ed., "Design Choices When Expanding the DNS", [RFC 5507](#), DOI 10.17487/RFC5507, April 2009, <<https://www.rfc-editor.org/info/rfc5507>>.
- [RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", [BCP 152](#), [RFC 5625](#), DOI 10.17487/RFC5625, August 2009, <<https://www.rfc-editor.org/info/rfc5625>>.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", [RFC 5927](#), DOI 10.17487/RFC5927, July 2010, <<https://www.rfc-editor.org/info/rfc5927>>.
- [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>>.
- [RFC5961] Ramaiah, A., Stewart, R., and M. Dalal, "Improving TCP's Robustness to Blind In-Window Attacks", [RFC 5961](#), DOI 10.17487/RFC5961, August 2010, <<https://www.rfc-editor.org/info/rfc5961>>.
- [RFC5966] Bellis, R., "DNS Transport over TCP - Implementation Requirements", [RFC 5966](#), DOI 10.17487/RFC5966, August 2010, <<https://www.rfc-editor.org/info/rfc5966>>.
- [RFC6304] Abley, J. and W. Maton, "AS112 Nameserver Operations", [RFC 6304](#), DOI 10.17487/RFC6304, July 2011, <<https://www.rfc-editor.org/info/rfc6304>>.

- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", [RFC 6762](#), DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [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>>.
- [RFC6950] Peterson, J., Kolkman, O., Tschafenig, H., and B. Aboba, "Architectural Considerations on Application Features in the DNS", [RFC 6950](#), DOI 10.17487/RFC6950, October 2013, <<https://www.rfc-editor.org/info/rfc6950>>.
- [RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", [RFC 7413](#), DOI 10.17487/RFC7413, December 2014, <<https://www.rfc-editor.org/info/rfc7413>>.
- [RFC7477] Hardaker, W., "Child-to-Parent Synchronization in DNS", [RFC 7477](#), DOI 10.17487/RFC7477, March 2015, <<https://www.rfc-editor.org/info/rfc7477>>.
- [RFC7720] Blanchet, M. and L-J. Liman, "DNS Root Name Service Protocol and Deployment Requirements", [BCP 40](#), [RFC 7720](#), DOI 10.17487/RFC7720, December 2015, <<https://www.rfc-editor.org/info/rfc7720>>.
- [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>>.
- [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>>.

- [RFC7901] Wouters, P., "CHAIN Query Requests in DNS", [RFC 7901](#), DOI 10.17487/RFC7901, June 2016, <<https://www.rfc-editor.org/info/rfc7901>>.
- [RFC7918] Langley, A., Modadugu, N., and B. Moeller, "Transport Layer Security (TLS) False Start", [RFC 7918](#), DOI 10.17487/RFC7918, August 2016, <<https://www.rfc-editor.org/info/rfc7918>>.
- [RFC8027] Hardaker, W., Gudmundsson, O., and S. Krishnaswamy, "DNSSEC Roadblock Avoidance", [BCP 207](#), [RFC 8027](#), DOI 10.17487/RFC8027, November 2016, <<https://www.rfc-editor.org/info/rfc8027>>.
- [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>>.
- [RFC8162] Hoffman, P. and J. Schlyter, "Using Secure DNS to Associate Certificates with Domain Names for S/MIME", [RFC 8162](#), DOI 10.17487/RFC8162, May 2017, <<https://www.rfc-editor.org/info/rfc8162>>.
- [RFC8324] Klensin, J., "DNS Privacy, Authorization, Special Uses, Encoding, Characters, Matching, and Root Structure: Time for Another Look?", [RFC 8324](#), DOI 10.17487/RFC8324, February 2018, <<https://www.rfc-editor.org/info/rfc8324>>.
- [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>>.
- [RFC8483] Song, L., Ed., Liu, D., Vixie, P., Kato, A., and S. Kerr, "Yeti DNS Testbed", [RFC 8483](#), DOI 10.17487/RFC8483, October 2018, <<https://www.rfc-editor.org/info/rfc8483>>.
- [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>>.
- [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>>.

- [RFC8501] Howard, L., "Reverse DNS in IPv6 for Internet Service Providers", [RFC 8501](#), DOI 10.17487/RFC8501, November 2018, <<https://www.rfc-editor.org/info/rfc8501>>.
- [ROLL_YOUR_ROOT]
Mueller, M., Thomas, M., Wessels, D., Hardaker, W., Chung, T., Toorop, W., and R. Rijswijk-Deij, "Roll, Roll, Roll Your Root: A Comprehensive Analysis of the First Ever DNSSEC Root KSK Rollover", Oct 2019, <TBD>.
- [RRL] Vixie, P. and V. Schryver, "DNS Response Rate Limiting (DNS RRL)", ISC-TN 2012-1 Draft1, April 2012.
- [Stevens] Stevens, W., Fenner, B., and A. Rudoff, "UNIX Network Programming Volume 1, Third Edition: The Sockets Networking API", November 2003.
- [TDNS] Zhu, L., Heidemann, J., Wessels, D., Mankin, A., and N. Somaiya, "Connection-oriented DNS to Improve Privacy and Security", 2015.
- [TOYAMA] Toyama, K., Ishibashi, K., Ishino, M., Yoshimura, C., and K. Fujiwara, "DNS Anomalies and Their Impacts on DNS Cache Servers", NANOG 32 Reston, VA USA, 2004.
- [VERISIGN]
Thomas, M. and D. Wessels, "An Analysis of TCP Traffic in Root Server DITL Data", DNS-OARC 2014 Fall Workshop Los Angeles, 2014.
- [WIKIPEDIA_TFO]
Wikipedia, "TCP Fast Open", May 2018, <https://en.wikipedia.org/wiki/TCP_Fast_Open>.

Appendix A. Standards Related to DNS Transport over TCP

This section enumerates all known IETF RFC documents that are currently of status standard, informational, best current practice, or experimental and either implicitly or explicitly make assumptions or statements about the use of TCP as a transport for the DNS germane to this document.

A.1. IETF [RFC 1035](#) - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION

The internet standard [[RFC1035](#)] is the base DNS specification that explicitly defines support for DNS over TCP.

[A.2.](#) IETF [RFC 1536](#) - Common DNS Implementation Errors and Suggested Fixes

The informational document [[RFC1536](#)] states UDP is the "chosen protocol for communication though TCP is used for zone transfers." That statement should now be considered in its historical context and is no longer a proper reflection of modern expectations.

[A.3.](#) IETF [RFC 1995](#) - Incremental Zone Transfer in DNS

The [[RFC1995](#)] standards track document documents the use of TCP as the fallback transport when IXFR responses do not fit into a single UDP response. As with AXFR, IXFR messages are typically delivered over TCP by default in practice.

[A.4.](#) IETF [RFC 1996](#) - A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)

The [[RFC1996](#)] standards track document suggests a master server may decide to issue NOTIFY messages over TCP. In practice, NOTIFY messages are generally sent over UDP, but this specification leaves open the possibility that the choice of transport protocol is up to the master server, and therefore a slave server ought to be able to operate over both UDP and TCP.

[A.5.](#) IETF [RFC 2181](#) - Clarifications to the DNS Specification

The [[RFC2181](#)] standards track document includes clarifying text on how a client should react to the TC bit set on responses. It is advised that the response should be discarded and the query resent using TCP.

[A.6.](#) IETF [RFC 2694](#) - DNS extensions to Network Address Translators (DNS_ALG)

The informational document [[RFC2694](#)] enumerates considerations for network address translation (NAT) devices to properly handle DNS traffic. This document is noteworthy in its suggestion that "[t]ypically, TCP is used for AXFR requests," as further evidence that helps explain why DNS over TCP may often have been treated very differently than DNS over UDP in operational networks.

[A.7.](#) IETF [RFC 3225](#) - Indicating Resolver Support of DNSSEC

The [[RFC3225](#)] standards track document makes statements indicating DNS over TCP is "detrimental" as a result of increased traffic, latency, and server load. This document is a companion to the next

document in the RFC series expressing the requirement for EDNS0 support for DNSSEC.

A.8. IETF [RFC 3326](#) - DNSSEC and IPv6 A6 aware server/resolver message size requirements

Although updated by later DNSSEC RFCs, the standards track document [[RFC3226](#)] strongly argued in favor of UDP messages over TCP largely for performance reasons. The document declares EDNS0 a requirement for DNSSEC servers and advocated packet fragmentation may be preferable to TCP in certain situations.

A.9. IETF [RFC 4472](#) - Operational Considerations and Issues with IPv6 DNS

This informational document [[RFC4472](#)] notes that IPv6 data may increase DNS responses beyond what would fit in a UDP message. Particularly noteworthy, perhaps less common today than when this document was written, it refers to implementations that truncate data without setting the TC bit to encourage the client to resend the query using TCP.

A.10. IETF [RFC 5452](#) - Measures for Making DNS More Resilient against Forged Answers

This informational document [[RFC5452](#)] arose as public DNS systems began to experience widespread abuse from spoofed queries, resulting in amplification and reflection attacks against unwitting victims. One of the leading justifications for supporting DNS over TCP to thwart these attacks is briefly described in this document's 9.3 Spoof Detection and Countermeasure section.

A.11. IETF [RFC 5507](#) - Design Choices When Expanding the DNS

This informational document [[RFC5507](#)] was largely an attempt to dissuade new DNS data types from overloading the TXT resource record type. In so doing it summarizes the conventional wisdom of DNS design and implementation practices. The authors suggest TCP overhead and stateful properties pose challenges compared to UDP, and imply that UDP is generally preferred for performance and robustness.

A.12. IETF [RFC 5625](#) - DNS Proxy Implementation Guidelines

This best current practice document [[RFC5625](#)] provides DNS proxy implementation guidance including the mandate that a proxy "MUST [...] be prepared to receive and forward queries over TCP" even though it suggests historically TCP transport has not been strictly mandatory in stub resolvers or recursive servers.

[A.13.](#) IETF [RFC 5936](#) - DNS Zone Transfer Protocol (AXFR)

The [[RFC5936](#)] standards track document provides a detailed specification for the zone transfer protocol, as originally outlined in the early DNS standards. AXFR operation is limited to TCP and not specified for UDP. This document discusses TCP usage at length.

[A.14.](#) IETF [RFC 5966](#) - DNS Transport over TCP - Implementation Requirements

This standards track document [[RFC5966](#)] instructs DNS implementers to provide support for carrying DNS over TCP messages in their software. The authors explicitly make no recommendations to operators, which we seek to address here.

[A.15.](#) IETF [RFC 6304](#) - AS112 Nameserver Operations

[[RFC6304](#)] is an informational document enumerating the requirements for operation of AS112 project DNS servers. New AS112 nodes are tested for their ability to provide service on both UDP and TCP transports, with the implication that TCP service is an expected part of normal operations.

[A.16.](#) IETF [RFC 6762](#) - Multicast DNS

In this standards track document [[RFC6762](#)], the TC bit is deemed to have essentially the same meaning as described in the original DNS specifications. That is, if a response with the TC bit set is received, "[...] the querier SHOULD reissue its query using TCP in order to receive the larger response."

[A.17.](#) IETF [RFC 6891](#) - Extension Mechanisms for DNS (EDNS(0))

This standards track document [[RFC6891](#)] helped slow the use of and need for DNS over TCP messages. This document highlights concerns over server load and scalability in widespread use of DNS over TCP.

[A.18.](#) IETF [RFC 6950](#) - Architectural Considerations on Application Features in the DNS

An informational document [[RFC6950](#)] that draws attention to large data in the DNS. TCP is referenced in the context as a common fallback mechanism and counter to some spoofing attacks.

A.19. IETF [RFC 7477](#) - Child-to-Parent Synchronization in DNS

This standards track document [[RFC7477](#)] specifies a RRType and protocol to signal and synchronize NS, A, and AAAA resource record changes from a child to parent zone. Since this protocol may require multiple requests and responses, it recommends utilizing DNS over TCP to ensure the conversation takes place between a consistent pair of end nodes.

A.20. IETF [RFC 7720](#) - DNS Root Name Service Protocol and Deployment Requirements

This best current practice [[RFC7720](#)] declares root name service "MUST support UDP [[RFC768](#)] and TCP [[RFC793](#)] transport of DNS queries and responses."

A.21. IETF [RFC 7766](#) - DNS Transport over TCP - Implementation Requirements

The standards track document [[RFC7766](#)] might be considered the direct ancestor of this operational requirements document. The implementation requirements document codifies mandatory support for DNS over TCP in compliant DNS software.

A.22. IETF [RFC 7828](#) - The edns-tcp-keepalive EDNS0 Option

This standards track document [[RFC7828](#)] defines an EDNS0 option to negotiate an idle timeout value for long-lived DNS over TCP connections. Consequently, this document is only applicable and relevant to DNS over TCP sessions and between implementations that support this option.

A.23. IETF [RFC 7858](#) - Specification for DNS over Transport Layer Security (TLS)

This standards track document [[RFC7858](#)] defines a method for putting DNS messages into a TCP-based encrypted channel using TLS. This specification is noteworthy for explicitly targeting the stub-to-recursive traffic, but does not preclude its application from recursive-to-authoritative traffic.

A.24. IETF [RFC 7873](#) - Domain Name System (DNS) Cookies

This standards track document [[RFC7873](#)] describes an EDNS0 option to provide additional protection against query and answer forgery. This specification mentions DNS over TCP as a reasonable fallback mechanism when DNS Cookies are not available. The specification does make mention of DNS over TCP processing in two specific situations.

In one, when a server receives only a client cookie in a request, the server should consider whether the request arrived over TCP and if so, it should consider accepting TCP as sufficient to authenticate the request and respond accordingly. In another, when a client receives a BADCOOKIE reply using a fresh server cookie, the client should retry using TCP as the transport.

[A.25.](#) IETF [RFC 7901](#) - CHAIN Query Requests in DNS

This experimental specification [[RFC7901](#)] describes an EDNS0 option that can be used by a security-aware validating resolver to request and obtain a complete DNSSEC validation path for any single query. This document requires the use of DNS over TCP or a source IP address verified transport mechanism such as EDNS-COOKIE [[RFC7873](#)].

[A.26.](#) IETF [RFC 8027](#) - DNSSEC Roadblock Avoidance

This document [[RFC8027](#)] details observed problems with DNSSEC deployment and mitigation techniques. Network traffic blocking and restrictions, including DNS over TCP messages, are highlighted as one reason for DNSSEC deployment issues. While this document suggests these sorts of problems are due to "non-compliant infrastructure" and is of type BCP, the scope of the document is limited to detection and mitigation techniques to avoid so-called DNSSEC roadblocks.

[A.27.](#) IETF [RFC 8094](#) - DNS over Datagram Transport Layer Security (DTLS)

This experimental specification [[RFC8094](#)] details a protocol that uses a datagram transport (UDP), but stipulates that "DNS clients and servers that implement DNS over DTLS MUST also implement DNS over TLS in order to provide privacy for clients that desire Strict Privacy [...]." This requirement implies DNS over TCP must be supported in case the message size is larger than the path MTU.

[A.28.](#) IETF [RFC 8162](#) - Using Secure DNS to Associate Certificates with Domain Names for S/MIME

This experimental specification [[RFC8162](#)] describes a technique to authenticate user X.509 certificates in an S/MIME system via the DNS. The document points out that the new experimental resource record types are expected to carry large payloads, resulting in the suggestion that "applications SHOULD use TCP -- not UDP -- to perform queries for the SMIMEA resource record."

- [A.29.](#) IETF [RFC 8324](#) - DNS Privacy, Authorization, Special Uses, Encoding, Characters, Matching, and Root Structure: Time for Another Look?**

An informational document [[RFC8324](#)] that briefly discusses the common role and challenges of DNS over TCP throughout the history of DNS.

- [A.30.](#) IETF [RFC 8467](#) - Padding Policies for Extension Mechanisms for DNS (EDNS(0))**

An experimental document [[RFC8467](#)] reminds implementers to consider the underlying transport protocol (e.g. TCP) when calculating the padding length when artificially increasing the DNS message size with an EDNS(0) padding option.

- [A.31.](#) IETF [RFC 8483](#) - Yeti DNS Testbed**

This informational document [[RFC8483](#)] describes a testbed environment that highlights some DNS over TCP behaviors, including issues involving packet fragmentation and operational requirements for TCP stream assembly in order to conduct DNS measurement and analysis.

- [A.32.](#) IETF [RFC 8484](#) - DNS Queries over HTTPS (DoH)**

This standards track document [[RFC8484](#)] defines a protocol for sending DNS queries and responses over HTTPS. This specification assumes TLS and TCP for the underlying security and transport layers, respectively. Self-described as a technique that more closely resembles a tunneling mechanism, DoH nevertheless likely implies DNS over TCP in some sense, if not directly.

- [A.33.](#) IETF [RFC 8490](#) - DNS Stateful Operations**

This standards track document [[RFC8490](#)] updates the base protocol specification with a new OPCODE to help manage stateful operations in persistent sessions, such as those that might be used by DNS over TCP.

- [A.34.](#) IETF [RFC 8501](#) - Reverse DNS in IPv6 for Internet Service Providers**

This informational document [[RFC8501](#)] identifies potential operational challenges with Dynamic DNS including denial-of-service threats. The document suggests TCP may provide some advantages, but that updating hosts would need to be explicitly configured to use TCP instead of UDP.

Authors' Addresses

John Kristoff
DataPlane.org
Chicago, IL 60605
US

Phone: +1 312 493 0305
Email: jtk@dataplane.org
URI: <https://dataplane.org/jtk/>

Duane Wessels
Verisign
12061 Bluemont Way
Reston, VA 20190
US

Phone: +1 703 948 3200
Email: dwessels@verisign.com
URI: <http://verisigninc.com>

