Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2014

P. Wouters
Red Hat
J. Abley
Dyn Inc.
October 15, 2013

The edns-tcp-keepalive EDNS0 Option
draft-wouters-edns-tcp-keepalive-00

Abstract

DNS messages between clients and servers may be received over either UDP or TCP. UDP transport involves keeping less state on a busy server, but can cause truncation and retries over TCP. Additionally, UDP can be exploited for reflection attacks. Using TCP would reduce retransmits and amplification. However, clients are currently limited in their use of the TCP transport as most implementations limit the TCP session to a single DNS query and answer, making use of TCP only suitable as a fallback protocol for UDP.

This document defines an EDNS0 option ("edns-tcp-keepalive") that allows DNS clients and servers to signal their respective readiness to conduct multiple DNS transactions over individual TCP sessions. This signalling facilitates a better balance of UDP and TCP transport between individual clients and servers, reducing the impact of problems associated with UDP transport and allowing the state associated with TCP transport to be managed effectively with minimal impact on the DNS transaction time.

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 18, 2014.

Table of Contents

1. Introduction

DNS messages between clients and servers may be received over either UDP or TCP [RFC1035]. Generally, DNS clients prefer to send queries over UDP, and fall back to TCP only if a query over UDP resulted in a truncated response (see [RFC1035] Section 4.1.1). A client that has

resorted to TCP transport as a reaction to a truncated response from a server typically closes the session after exchanging a single (request, response) DNS message pair, and continues with UDP transport for subsequent queries. Although [RFC1035] specifies that a single TCP session may be used to exchange multiple DNS messages, in practice this is rarely seen.

UDP transport is stateless, and hence presents a much lower resource burden on a busy DNS server than TCP. An exchange of DNS messages over UDP can also be completed in a single round trip between communicating hosts, resulting in optimally-short transaction times. UDP transport is not without its risks, however.

A single-datagram exchange over UDP between two hosts can be exploited to enable a reflection attack on a third party. Mitigation of such attacks on authoritative-only servers is possible using an approach known as Response Rate-Limiting [RRL], an approach designed to minimise the frequency at which legitimate responses are discarded by truncating responses that appear to be motivated by an attacker, forcing legitimate clients to re-query using TCP transport.

[RFC1035] specified a maximum DNS message size over UDP transport of 512 bytes. Deployment of DNSSEC [RFC4033] and other protocols subsequently increased the observed frequency at which responses exceed this limit. EDNS0 [RFC6891] allows DNS messages larger than 512 bytes to be exchanged over UDP, with a corresponding increased incidence of fragmentation. Fragmentation is known to be problematic in general, and has also been implicated in increasing the risk of cache poisoning attacks.

The use of TCP transport does not suffer from the risks of fragmentation nor reflection attacks. However, TCP transport as currently deployed has expensive overhead.

The overhead of the three-way TCP handshake for a single DNS transaction is substantial, increasing the transaction time for a

single (request, response) pair of DNS messages from 1 x RTT to 2 x RTT. There is no such overhead for a session that is already established, however, and the overall impact of the TCP setup handshake when the resulting session is used to exchange N DNS message pairs over a single session, (1 + N)/N, approaches unity as N increases.

(It should perhaps be noted that the overhead for a DNS transaction over UDP truncacated due to RRL is 3x RTT, higher than the overhead imposed on the same transaction initiated over TCP.)

With increased deployment of DNSSEC and new RRtypes containing application specific cryptographic material, there is an increase in UDP truncation with fallback to TCP.

The use of TCP transport requires considerably more state to be retained on DNS servers. If a server is to perform adequately with a significant query load received over TCP, it must manage its available resources to ensure that all established TCP sessions are well-used, and that those which are unlikely to be used for the exchange of multiple DNS messages are closed promptly.

This document proposes a signalling mechanism between DNS clients and servers that provides a means for servers to better balance the use of UDP and TCP transport, reducing the impact of problems associated with UDP whilst constraining the impact of TCP on response times and server resources to a manageable level.

The reduced overhead of this extension adds up significantly when combined with other edns extensions, such as [CHAIN-QUERY]. The combination of these two EDNS extensions make it possible for hosts on high-latency mobile networks to natively perform DNSSEC validation.

2. Requirements Notation

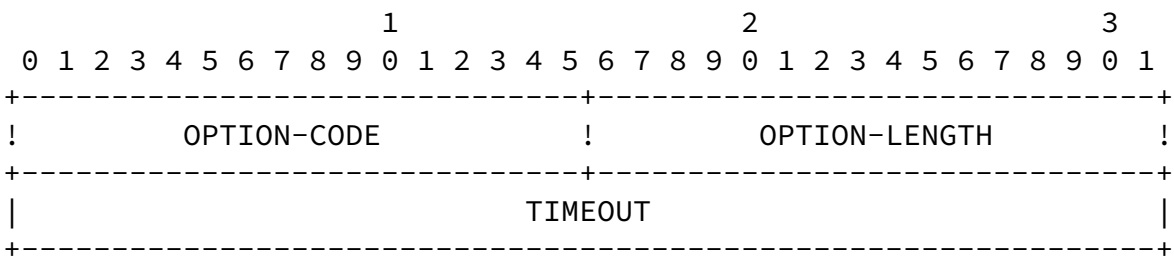
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. The edns-tcp-keepalive Option

This document specifies a new EDNS0 [RFC6891] option, edns-tcp-keepalive, which can be used by DNS clients and servers to signal a willingness to conduct multiple DNS transactions over a single TCP session. This specification does not distinguish between different types of DNS client and server in the use of this option.

3.1. Option Format

The edns-tcp-keepalive option is encoded as follows:



where:

OPTION-CODE: the EDNS0 option code assigned to edns-tcp-keepalive, [TBD]

OPTION-LENGTH: the value 2;

TIMEOUT: a timeout value for the TCP connection specified by DNS servers, specified in seconds, encoded in network byte order. DNS clients set this value to 0.

3.2. Use by DNS Clients

3.2.1. Sending Queries

DNS clients MAY include the edns-tcp-keepalive option in queries sent using UDP transport to signal their general ability to use individual TCP sessions for multiple DNS transactions with a particular server.

DNS clients MAY include the edns-tcp-keepalive option in the first query sent to a server using TCP transport to signal their desire that that specific TCP session be used for multiple DNS transactions.

DNS Clients MUST specify a TIMEOUT value of zero.

3.2.2. Receiving Responses

A DNS client that receives a response using UDP transport that includes the edns-tcp-keepalive option MAY record the presence of the option and the associated TIMEOUT value, and use that information as part of its server selection algorithm in the case where multiple candidate servers are available to service a particular query.

A DNS client that receives a response using TCP transport that includes the edns-tcp-keepalive option MAY keep the existing TCP session open.

A DNS client that receives a response that includes the edns-tcp-keepalive option with a TIMEOUT value of 0 is allowed to keep the TCP connection open indefinately.

3.3. Use by DNS Servers

3.3.1. Receiving Queries

A DNS server that receives a query using UDP transport that includes the edns-tcp-keepalive option MAY record the presence of the option for statistical purposes, but should not otherwise modify its usual behaviour in sending a response.

A DNS server that receives a query that includes the edns-tcp-keepalive option MUST ignore the TIMEOUT value

3.3.2. Sending Responses

DNS servers MAY include the edns-tcp-keepalive option in responses sent using UDP transport to signal their general ability to use individual TCP sessions for multiple DNS transactions with a particular server. The TIMEOUT value should be indicative of what a client might expect if it was to open a TCP session with the server and receive a response with the edns-tcp-keepalive option present. The DNS server MAY omit including the edns-tcp-keepalive option if it is running too low on resources to service more TCP keepalive sessions.

DNS servers MAY include the edns-tcp-keepalive option in responses sent using TCP transport to signal their ability to use that specific session to exchange multiple DNS transactions. Servers MUST specify the TIMEOUT value that is currently associated with the TCP session. It is reasonable for this value to change according to local resource constraints. The DNS server MAY omit including the edns-tcp-keepalive option if it deems its local resources are too low to service more TCP keepalive sessions.

3.4. TCP Session Management

Both DNS clients and servers are subject to resource constraints which will limit the extent to which TCP sessions can persist. Effective limits for the number of active sessions that can be maintained on individual clients and servers should be established, either as configuration options or by interrogation of process limits imposed by the operating system.

In the event that there is greater demand for TCP sessions than can be accommodated, servers may reduce the TIMEOUT value signalled in

successive DNS messages to avoid abrupt termination of a session. This allows, for example, clients with other candidate servers to query to establish new TCP sessions with different servers in expectation that an existing session is likely to be closed, or to fall back to UDP.

DNS clients and servers MAY close a TCP session at any time in order to manage local resource constraints. The algorithm by which clients

and servers rank active TCP sessions in order to determine which to close is not specified in this document.

3.5. Non-Clean Paths

Many paths between DNS clients and servers suffer from poor hygiene, limiting the free flow of DNS messages that include particular EDNS0 options, or messages that exceed a particular size. A fallback strategy similar to that described in [RFC6891] section 6.2.2 SHOULD be employed to avoid persistent interference due to non-clean paths.

3.6. Anycast Considerations

DNS servers of various types are commonly deployed using anycast [RFC4786].

Successive DNS transactions between a client and server using UDP transport may involve responses generated by different anycast nodes, and the use of anycast in the implementation of a DNS server is effectively undetectable by the client. The edns-tcp-keepalive option SHOULD NOT be included in responses using UDP transport from servers provisioned using anycast unless all anycast server nodes are capable of processing the edns-tcp-keepalive option. The TIMEOUT values in UDP responses from anycast servers MUST be zero to indicate that there is no useful value that can be specified.

Changes in network topology between clients and anycast servers may cause disruption to TCP sessions making use of edns-tcp-keepalive more often than with TCP sessions that omit it, since the TCP sessions are expected to be longer-lived. Anycast servers MAY make use of TCP multipath [RFC6824] to anchor the server side of the TCP connection to an unambiguously-unicast address in order to avoid disruption due to topology changes.

4. Security Considerations

The edns-tcp-keep-alive option can potentially be abused to request large numbers of sessions in a quick burst. When a Nameserver detects abusive behaviour, it SHOULD immediately close the TCP connection and free all buffers used.

This section needs more work. As usual.

5. IANA Considerations

The IANA is directed to assign an EDNS0 option code for the edns-tcp-keepalive option from the DNS EDNS0 Option Codes (OPT) registry as follows:

Value	Name	Status	Reference
[TBA]	edns-tcp-keepalive	Optional	[This document]

6. Acknowledgements

empty for now

7. References

7.1. Normative References

[CHAIN-QUERY] Wouters, P., "TCP chain query requests in DNS", draft-wouters-edns-tcp-chain-query (work in progress), October 2013.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, April 2013.

7.2. Informative References

[RRL] Vixie, P. and V. Schryver, "DNS Response Rate Limiting (DNS RRL)", ISC-TN 2012-1-Draft1, April 2012.

Appendix A. Editors' Notes

A.1. Venue

An appropriate venue for discussion of this document is dnsext@ietf.org.

A.2. Abridged Change History

A.2.1. draft-wouters-edns-tcp-keepalive-00

Initial draft.

Authors' Addresses

Paul Wouters
Red Hat

Email: pwouters@redhat.com

Joe Abley
Dyn Inc.
470 Moore Street
London, ON N6C 2C2
Canada

Phone: +1 519 670 9327
Email: jabley@dyn.com