DNSEXT	R. Bellis	
Internet-Draft	Nominet UK	
Updates: 1035, 1123	March 22, 2010	
(if approved)		
Intended status: Standards Track		
Expires: September 23, 2010		

TOC

DNS Transport over TCP - Implementation Requirements
draft-ietf-dnsext-dns-tcp-requirements-03

Abstract

This document updates the requirements for the support of TCP as a transport protocol for DNS implementations.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.
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.”
The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.
This Internet-Draft will expire on September 23, 2010.

Table of Contents

---

1. Introduction

TOC

Most DNS (Mockapetris, P., “Domain names - implementation and specification,” November 1987.) [RFC1035] transactions take place over UDP (Postel, J., “User Datagram Protocol,” August 1980.) [RFC0768]. TCP (Postel, J., “Transmission Control Protocol,” September 1981.) [RFC0793] is always used for zone transfers and is often used for messages whose sizes exceed the DNS protocol's original 512 byte limit. Section 6.1.3.2 of [RFC1123] (Braden, R., “Requirements for Internet Hosts - Application and Support,” October 1989.) states:

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

However, some implementors have taken the text quoted above to mean that TCP support is an optional feature of the DNS protocol.
The majority of DNS server operators already support TCP and the default configuration for most software implementations is to support TCP. The primary audience for this document is those implementors whose failure to support TCP restricts interoperability and limits deployment of new DNS features.
This document therefore updates the core DNS protocol specifications such that support for TCP is henceforth a REQUIRED part of a full DNS protocol implementation.
Whilst this document makes no specific recommendations to operators of DNS servers, it should be noted that failure to support TCP (or blocking of DNS over TCP at the network layer) may result in resolution failure and/or application-level timeouts.

---

2. Terminology used in this document

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

---

3. Discussion

TOC

In the absence of EDNS0 (see below) the normal behaviour of any DNS server needing to send a UDP response that would exceed the 512 byte limit is for the server to truncate the response so that it fits within that limit and then set the TC flag in the response header. When the client receives such a response it takes the TC flag as an indication that it should retry over TCP instead.
RFC 1123 also says:

> ... 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. Thus, resolvers and name servers should implement TCP services as a backup to UDP today, with the knowledge that they will require the TCP service in the future.

Existing deployments of DNSSEC (Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, “DNS Security Introduction and Requirements,” March 2005.) [RFC4033] have shown that truncation at the 512 byte boundary is now commonplace. For example an NXDOMAIN (RCODE == 3)

response from a DNSSEC signed zone using NSEC3 (Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence," March 2008.) [RFC5155] is almost invariably larger than 512 bytes.

Since the original core specifications for DNS were written, the Extension Mechanisms for DNS (EDNS0 (Vixie, P., "Extension Mechanisms for DNS (EDNS0)," August 1999.) [RFC2671]) have been introduced. These extensions can be used to indicate that the client is prepared to receive UDP responses larger than 512 bytes. An EDNS0 compatible server receiving a request from an EDNS0 compatible client may send UDP packets up to that client's announced buffer size without truncation. However, transport of UDP packets that exceed the size of the path MTU causes IP packet fragmentation, which has been found to be unreliable in some circumstances. Many firewalls routinely block fragmented IP packets, and some do not implement the algorithms necessary to reassemble fragmented packets. Worse still, some network devices deliberately refuse to handle DNS packets containing EDNS0 options. Other issues relating to UDP transport and packet size are discussed in [RFC5625] (Bellis, R., "DNS Proxy Implementation Guidelines," August 2009.).

The MTU most commonly found in the core of the Internet is around 1500 bytes, and even that limit is routinely exceeded by DNSSEC signed responses.

The future that was anticipated in RFC 1123 has arrived, and the only standardised UDP-based mechanism which may have resolved the packet size issue has been found inadequate.

---

4. Transport Protocol Selection

TOC

All general purpose DNS implementations MUST support both UDP and TCP transport.

- Authoritative server implementations MUST support TCP so that they do not limit the size of responses.
- Recursive resolver (or forwarder) implementations MUST support TCP so that the do not prevent large responses from a TCP-capable server from reaching its TCP-capable clients.
- Stub resolver implementations (e.g. an operating system's DNS resolution library) MUST support TCP since to do otherwise would limit their interoperability with their own clients and with upstream servers.

An exception may be made for proprietary stub resolver implementations. These MAY omit support for TCP if operating in an environment where

truncation can never occur, or where DNS lookup failure is acceptable should truncation occur.
Regarding the choice of when to use UDP or TCP, RFC 1123 says:

> ... a DNS resolver or server that is sending a non-zone-transfer query MUST send a UDP query first.

That requirement is hereby relaxed. A resolver SHOULD send a UDP query first, but MAY elect to send a TCP query instead if it has good reason to expect the response would be truncated if it were sent over UDP (with or without EDNS0) or for other operational reasons, in particular if it already has an open TCP connection to the server.

---

5. Connection Handling

TOC

Section 4.2.2 of [RFC1035] (Mockapetris, P., “Domain names - implementation and specification,” November 1987.) says:

> If the server needs to close a dormant connection to reclaim resources, it should wait until the connection has been idle for a period on the order of two minutes. In particular, the server should allow the SOA and AXFR request sequence (which begins a refresh operation) to be made on a single connection. Since the server would be unable to answer queries anyway, a unilateral close or reset may be used instead of a graceful close.

Other more modern protocols (e.g. HTTP (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.) [RFC2616]) have support for persistent TCP connections and operational experience has shown that long timeouts can easily cause resource exhaustion and poor response under heavy load. Intentionally opening many connections and leaving them dormant can trivially create a "denial of service" attack. This document therefore RECOMMENDS that the default application-level idle period should be of the order of seconds, but does not specify any particular value. In practise the idle period may vary dynamically, and servers MAY allow dormant connections to remain open for longer periods as resources permit.
To mitigate the risk of unintentional server overload, DNS clients MUST take care to minimize the number of concurrent TCP connections made to any individual server. Similarly servers MAY impose limits on the number of concurrent TCP connections being handled for any particular client.

Further recommendations for the tuning of TCP stacks to allow higher throughput or improved resiliency against denial of service attacks are outside the scope of this document.

---

6. Response re-ordering

TOC

RFC 1035 is ambiguous on the question of whether TCP queries may be re-ordered - the only relevant text is in Section 4.2.1 which relates to UDP:

> Queries or their responses may be reordered by the network, or by processing in name servers, so resolvers should not depend on them being returned in order.

For the avoidance of future doubt, this requirement is clarified. Client resolvers MUST be able to process responses which arrive in a different order to that in which the requests were sent, regardless of the transport protocol in use.

---

7. Security Considerations

TOC

Some DNS server operators have expressed concern that wider use of DNS over TCP will expose them to a higher risk of denial of service (DoS) attacks.
Although there is a higher risk of such attacks against TCP-enabled servers, techniques for the mitigation of DoS attacks at the network level have improved substantially since DNS was first designed.
At the time of writing the vast majority of TLD authority servers and all of the root name servers support TCP and the author knows of no evidence to suggest that TCP-based DoS attacks against existing DNS infrastructure are commonplace.
That notwithstanding, readers are advised to familiarise themselves with [CPNI-TCP] (CPNI, “Security Assessment of the Transmission Control Protocol (TCP),” 2009.).
Operators of recursive servers should ensure that they only accept connections from expected clients, and do not accept them from unknown sources. In the case of UDP traffic this will help protect against reflector attacks (Damas, J. and F. Neves, “Preventing Use of Recursive Nameservers in Reflector Attacks,” October 2008.) [RFC5358] and in the case of TCP traffic it will prevent an unknown client from exhausting the server's limits on the number of concurrent connections.

---

8. IANA Considerations

TOC

This document requests no IANA actions.

---

9. Acknowledgements

TOC

The author would like to thank the document reviewers from the DNSEXT Working Group, and in particular George Barwood, Alex Bligh, Alfred Hoenes, Fernando Gont, Jim Reid, Paul Vixie and Nicholas Weaver.

---

10. References

TOC

10.1. Normative References

TOC

[RFC0768]	Postel, J., “User Datagram Protocol,” STD 6, RFC 768, August 1980 (TXT).
[RFC0793]	Postel, J., “Transmission Control Protocol,” STD 7, RFC 793, September 1981 (TXT).
[RFC1035]	Mockapetris, P., “Domain names - implementation and specification,” STD 13, RFC 1035, November 1987 (TXT).
[RFC1123]	Braden, R., “Requirements for Internet Hosts - Application and Support,” STD 3, RFC 1123, October 1989 (TXT).
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2671]	Vixie, P., “Extension Mechanisms for DNS (EDNS0),” RFC 2671, August 1999 (TXT).

10.2. Informative References

TOC

[CPNI-TCP]	CPNI, “Security Assessment of the Transmission Control Protocol (TCP),” 2009.
[RFC2616]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” RFC 2616, June 1999 (TXT, PS, PDF, HTML, XML).
[RFC4033]	

	Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements," RFC 4033, March 2005 (TXT).
[RFC5155]	Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence," RFC 5155, March 2008 (TXT).
[RFC5358]	Damas, J. and F. Neves, "Preventing Use of Recursive Nameservers in Reflector Attacks," BCP 140, RFC 5358, October 2008 (TXT).
[RFC5625]	Bellis, R., "DNS Proxy Implementation Guidelines," BCP 152, RFC 5625, August 2009 (TXT).

Appendix A. Change Log

TOC

NB: to be removed by the RFC Editor before publication.

draft-ietf-dnsext-dns-tcp-requirements-03

> Editorial nits from WGLC
>
> Clarification on "general purpose"
>
> Fixed ref to UDP (RFC 768)
>
> Included more §4.2.2 text from RFC 1035 and removed some from this draft relating to connection resets.
>
> s/long/large/ for packet sizes

draft-ietf-dnsext-dns-tcp-requirements-02

> Change of title - more focus on implementation and not operation
>
> Re-write of some of the security section
>
> Added recommendation for minimal concurrent connections
>
> Minor editorial nits from Alfred Hoenes

draft-ietf-dnsext-dns-tcp-requirements-01

> Addition of response ordering section
>
> Various minor editorial changes from WG reviewers

draft-ietf-dnsext-dns-tcp-requirements-00

> Initial draft

Author's Address

TOC

	Ray Bellis
	Nominet UK
	Edmund Halley Road
	Oxford OX4 4DQ
	United Kingdom
Phone:	+44 1865 332211
Email:	ray.bellis@nominet.org.uk
URI:	http://www.nominet.org.uk/