Internet Engineering Task Force (IETF)
Request for Comments: 5966
Updates: 1035, 1123
Category: Standards Track
ISSN: 2070-1721

R. Bellis
Nominet UK

August 2010

DNS Transport over TCP - Implementation Requirements

Abstract

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5966.

Table of Contents

1. Introduction

Most DNS [RFC1034] transactions take place over UDP [RFC0768]. TCP [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] 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 the 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

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. Discussion

In the absence of EDNS0 (Extension Mechanisms for DNS 0) (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 DNS Security (DNSSEC) [RFC4033] have shown that truncation at the 512-byte boundary is now commonplace. For example, a Non-Existent Domain (NXDOMAIN) (RCODE == 3) response from a DNSSEC-signed zone using NextSECure 3 (NSEC3) [RFC5155] is almost invariably larger than 512 bytes.

Since the original core specifications for DNS were written, the Extension Mechanisms for DNS (EDNS0 [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].

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 that may have resolved the packet size issue has been found inadequate.

4. Transport Protocol Selection

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 to what fits in a single UDP packet.

- Recursive server (or forwarder) implementations MUST support TCP so that they 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.

Stub resolver implementations MAY omit support for TCP when specifically designed for deployment in restricted environments where truncation can never occur or where truncated DNS responses are acceptable.

Regarding the choice of when to use UDP or TCP, Section 6.1.3.2 of RFC 1123 also 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

Section 4.2.2 of [RFC1035] 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 [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.

It is therefore RECOMMENDED that the default application-level idle period should be of the order of seconds, but no particular value is specified. 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 Reordering

RFC 1035 is ambiguous on the question of whether TCP queries may be reordered -- 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 that arrive in a different order to that in which the requests were sent, regardless of the transport protocol in use.

7. Security Considerations

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 Top Level Domain (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].

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 [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. Acknowledgements

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, Olafur Gudmondsson, Jim Reid, Paul Vixie, and Nicholas Weaver.

9. References

9.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

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

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

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

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

9.2. Informative References

[CPNI-TCP] CPNI, "Security Assessment of the Transmission Control Protocol (TCP)", 2009, <http://www.cpni.gov.uk/Docs/tn-03-09-security-assessment-TCP.pdf>.

[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.

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

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC5358] Damas, J. and F. Neves, "Preventing Use of Recursive Nameservers in Reflector Attacks", BCP 140, RFC 5358, October 2008.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, August 2009.

Author’s Address

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/