Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 17, 2014

P. Vixie
Farsight Security, Inc.
A. Kato
Keio University/WIDE Project
J. Abley
Dyn, Inc.

February 13, 2014

DNS Referral Response Size Issues
draft-ietf-dnsop-respsize-15

Abstract

With a mandated default minimum maximum UDP message size of 512 octets, the DNS protocol presents some special problems for zones wishing to expose a moderate or high number of authority servers (NS resource records). This document explains the operational issues caused by, or related to this response size limit, and suggests ways to optimize the use of this limited space. Guidance is offered to DNS server implementors and to DNS zone administrators.

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 August 17, 2014.

Copyright Notice

Table of Contents

1. Terminology

This document uses terminology specific to the Domain Name System (DNS), including the following common abbreviations:

A: A resource record type used to specify an IPv4 address [RFC1034]

AAAA: A resource record type used to specify an IPv6 address [RFC3596]

CNAME: A resource record type used to define a canonical name [RFC1034]

DNAME: A resource record type used to map a DNS subtree onto another domain [RFC2672]

DNSSEC: DNS Security Extensions [RFC4033]

DO: "DNS OK" -- a flag in the EDNS header used to signal the ability to use DNSSEC [RFC4035]

EDNS: Extension mechanisms for DNS [RFC6891]

EDNS0: EDNS version 0 [RFC6891]

MTU: Maximum Transmission Unit, the maximum size for a datagram to be forwarded on an interface without needing fragmentation [RFC0791] [RFC2460]

NS: A resource record type used to specify a nameserver on either side of a zone cut [RFC1034]

RR: Resource Record [RFC1034]

RRSet: Resource Record Set [RFC1034]

TC: A bit in the DNS message header used to indicate that the message has been truncated [RFC1034]

In an exchange of DNS messages between two hosts, this document refers to the host sending a DNS request as the initiator, and the host sending a DNS response as the responder.

2. Introduction and Overview

The original DNS standard limited the UDP message size to 512 octets (see Section 4.2.1 of [RFC1035]). Even though this limitation was due to the required minimum IP reassembly limit for IPv4, it became a hard DNS protocol limit and is not implicitly relaxed by changes in a network layer protocol, e.g. by the larger minimum MTU specified in IPv6 [RFC2460] than in IPv4 [RFC0791].

The EDNS protocol extension starting with version 0 permits larger responses by mutual agreement of the initiator and responder (see Section 4.3 and Section 6.2 of [RFC6891]), and it is recommended to support EDNS. The 512 octets UDP message size limit will remain in practical effect until substantially all DNS servers and resolvers support EDNS.

Since DNS responses include a copy of the request, the space available for response data is somewhat less than the full 512 octets. Negative responses are quite small, but for positive and referral responses, every octet must be carefully and sparingly allocated. While the response size of positive responses is also a concern in [RFC3226], this document specifically addresses referral response size.

While more than fourteen years passed since the publication of the original EDNS0 document [RFC2671], measurements conducted at the M Root Server in May 2012 suggested that only around 65% of initiators support it. This fraction was consistent with similar measurements conducted in 2010 and 2011. The long tail of EDNS deployment may eventually be measured in decades.

DNS initiators and responders that support DNSSEC [RFC4033], and signal a desire to use it, can expect larger response sizes in the case where those responses contain DNSSEC RRSets. EDNS support in DNSSEC-aware initiators and responders can be assumed, since the desire to use DNSSEC is signalled using the DO flag in the EDNS0 header.

Even in scenarios where EDNS support in initiators and responders can be assumed, e.g. in the case of messages exchanged using DNSSEC, or at some future time where EDNS deployment can be considered ubiquitous, there will still be cases when MTU limitations or IP fragmentation/reassembly problems in firewalls and other middleboxes will cause EDNS failures which lead to non-extended DNS retries. A smaller referral response will always be better than a larger one if the same end result can be achieved either way. See [RFC5625], [SAC035], and Section 6.2.6 of [RFC6891] for further discussion.

3. Delegation Details

3.1. Relevant Protocol Elements

A positive delegation response will include the following elements:

Section	Description
Header Section	Fixed length (12 octets)
Question Section	Original query (name, class, type)
Answer Section	Empty, or a CNAME/DNAME chain
Authority Section	NS RRSet (name server names)
Additional Section	A and AAAA RRSets (name server addresses)

If the total size of the UDP response exceeds 512 octets or the size advertised in EDNS, and if the data that does not fit was "required", then the TC bit will be set to indicate truncation. This will usually cause the requester to retry using TCP, depending on what information was desired and what information was omitted. For example, truncation in the authority section is of no interest to a stub resolver who only plans to consume the answer section. If a retry using TCP is needed, the total cost of the transaction is much higher. See Section 6.1.3.2 of [RFC1123] for details on the requirement that UDP be attempted before falling back to TCP.

RRSets are are never sent partially unless the TC bit is set to indicate truncation. When the TC bit is set, the final apparent RRSet in the final non-empty section must be considered "possibly damaged" (see Section 6.2 of [RFC1035] and Section 9 of [RFC2181]).

With or without truncation, the glue present in the additional data section should be considered "possibly incomplete", and requesters should be prepared to re-query for any damaged or missing RRSets. Note that truncation of the additional data section might not be signaled via the TC bit since additional data is often optional (see discussion in Appendix B of [RFC4472]).

DNS label compression allows the component labels of a domain name to be instantiated exactly once per DNS message, and then referenced with a two-octet "pointer" from other locations in that same DNS message (see Section 4.1.4 of [RFC1035]). If all name server names in a message share a common parent domain (for example, all of them

are in the "ROOT-SERVERS.NET" domain), then more space will be available for incompressible data (such as name server addresses).

The query name can be as long as 255 octets of network data. In this worst case scenario, the question section will be 259 octets in size, which would leave only 240 octets for the authority and additional sections (after deducting 12 octets for the fixed length header) in a referral.

3.2. Advice to Zone Administrators

Average and maximum question section sizes can be predicted by the zone administrator, since they will know what names actually exist and can measure which ones are queried for most often. Note that if the zone contains any wildcards, it is possible for maximum length queries to require positive responses, but that it is reasonable to expect truncation and TCP retry in that case. For cost and performance reasons, the majority of requests should be satisfied without truncation or TCP retry.

Some queries for non-existant names can be large. If DNSSEC is not being used this is unlikely to pose a problem since unsigned negative responses need not contain any answer, authority or additional records. See Section 2.1 of [RFC2308] for more information about the format of negative responses without DNSSEC. Negative responses from DNSSEC-signed zones can be much larger, however, due to the need to provide authenticated denial of existance [RFC7129].

The minimum useful number of name servers is two, for redundancy (see Section 4.1 of [RFC1034]). A zone's name servers should be reachable by all IP protocols versions (e.g., IPv4 and IPv6) in common use. As long as the servers are well managed, the server serving IPv6 might be different from the server serving IPv4 sharing the same server name.

The best case is no truncation at all. This is because many requesters will retry using TCP immediately, or will automatically requery for RRSets that are possibly truncated, without considering whether the omitted data was actually necessary.

Anycast [RFC3258] [RFC4786] is a useful technique for improving performance and below the zone cut being described by a delegation is responses.

While it is irrelevant to the response size issue, all zones have to be served via IPv4 as well as IPv6 to avoid name space fragmentation [RFC3901].

3.3. Advice to Server Implementors

Each NS RR for a zone will add 12 fixed octets (name, type, class, ttl, and rdlen) plus 2 to 255 variable octets (for the NSDNAME). Each A RR will require 16 octets, and each AAAA RR will require 28 octets.

While DNS distinguishes between necessary and optional resource records, this distinction is according to protocol elements necessary to signify facts, and takes no official notice of protocol content necessary to ensure correct operation. For example, a name server name that is in or below the zone cut being described by a delegation is "necessary content", since there is no way to reach that zone unless the parent zone's delegation includes "glue records" describing that name server's addresses.

Recall that the TC bit is only set when a required RRSet can not be included in its entirety (see Section 9 of [RFC2181]). Even when some of the RRSets to be included in the additional section don't fit in the response size, the TC bit isn't set. These RRSets may be important for a referral. Some DNS implementations try to resolve these missing glue records separately which will introduce extra queries and extra time to resolve a given name.

A delegation response should prioritize glue records as follows.

first: All glue RRSets for one name server whose name is in or below the zone being delegated, or which has multiple address RRSets (currently A and AAAA), or preferably both;

second: Alternate between adding all glue RRSets for any name servers whose names are in or below the zone being delegated, and all glue RRSets for any name servers who have multiple address RRSets (currently A and AAAA);

thence: All other glue RRSets, in any order.

Whenever there are multiple candidates for a position in this priority scheme, one should be chosen on a round-robin or fully random basis. The goal of this priority scheme is to offer "necessary" glue first to fill into the response if possible.

If any "necessary" content cannot be fit in the response, then it is advisable that the TC bit be set in order to force a TCP retry, rather than have the zone be unreachable. Note that a parent server's proper response to a query for in-child glue or below-child glue is a referral rather than an answer, and that this referral must be able to contain the in-child or below-child glue, and that in

outlying cases, only EDNS or TCP will be large enough to contain that data.

The glue record order should be independent of the version of IP used in the query because the DNS server might just see a query from an intermediate server rather than the query from the original client.

4. Analysis

An instrumented protocol trace of a best case delegation response is shown in Figure 1. Note that 13 servers are named, and 13 addresses are given. This query was artificially designed to exactly reach the 512 octets limit.

```
;; flags: qr rd; QUERY: 1, ANS: 0, AUTH: 13, ADDIT: 13
;; QUERY SECTION:
;;  [23456789.123456789.123456789.\
     123456789.123456789.123456789.com A IN]        ;; @80

;; AUTHORITY SECTION:
com.                172800 NS  E.GTLD-SERVERS.NET.  ;; @112
com.                172800 NS  F.GTLD-SERVERS.NET.  ;; @128
com.                172800 NS  G.GTLD-SERVERS.NET.  ;; @144
com.                172800 NS  H.GTLD-SERVERS.NET.  ;; @160
com.                172800 NS  I.GTLD-SERVERS.NET.  ;; @176
com.                172800 NS  J.GTLD-SERVERS.NET.  ;; @192
com.                172800 NS  K.GTLD-SERVERS.NET.  ;; @208
com.                172800 NS  L.GTLD-SERVERS.NET.  ;; @224
com.                172800 NS  M.GTLD-SERVERS.NET.  ;; @240
com.                172800 NS  A.GTLD-SERVERS.NET.  ;; @256
com.                172800 NS  B.GTLD-SERVERS.NET.  ;; @272
com.                172800 NS  C.GTLD-SERVERS.NET.  ;; @288
com.                172800 NS  D.GTLD-SERVERS.NET.  ;; @304


;; ADDITIONAL SECTION:
A.GTLD-SERVERS.NET.  172800 A   192.5.6.30          ;; @320
B.GTLD-SERVERS.NET.  172800 A   192.33.14.30        ;; @336
C.GTLD-SERVERS.NET.  172800 A   192.26.92.30        ;; @352
D.GTLD-SERVERS.NET.  172800 A   192.31.80.30        ;; @368
E.GTLD-SERVERS.NET.  172800 A   192.12.94.30        ;; @384
F.GTLD-SERVERS.NET.  172800 A   192.35.51.30        ;; @400
G.GTLD-SERVERS.NET.  172800 A   192.42.93.30        ;; @416
H.GTLD-SERVERS.NET.  172800 A   192.54.112.30       ;; @432
I.GTLD-SERVERS.NET.  172800 A   192.43.172.30       ;; @448
J.GTLD-SERVERS.NET.  172800 A   192.48.79.30        ;; @464
K.GTLD-SERVERS.NET.  172800 A   192.52.178.30       ;; @480
L.GTLD-SERVERS.NET.  172800 A   192.41.162.30       ;; @496
M.GTLD-SERVERS.NET.  172800 A   192.55.83.30        ;; @512

;; MSG SIZE  sent: 80  rcvd: 512
```

Figure 1

For longer query names, the number of address records supplied will be lower. Furthermore, it is only by using a common parent name (which is "GTLD-SERVERS.NET." in this example) that all 13 addresses are able to fit, due to the use of label compression pointers in the last 12 occurrences of the parent domain name. The outputs from the response simulator in Appendix A (written in perl [PERL]) shown in Figure 2 and Figure 3 demonstrate these properties.

```
% perl respsize.pl a.dns.br b.dns.br c.dns.br d.dns.br
a.dns.br requires 10 bytes
b.dns.br requires 4 bytes
c.dns.br requires 4 bytes
d.dns.br requires 4 bytes
# of NS: 4
For maximum size query (255 byte):
    only A is considered:        # of A is 4 (green)
    A and AAAA are considered:   # of A+AAAA is 3 (yellow)
    preferred-glue A is assumed: # of A is 4, # of AAAA is 3 (yellow)
For average size query (64 byte):
    only A is considered:        # of A is 4 (green)
    A and AAAA are considered:   # of A+AAAA is 4 (green)
    preferred-glue A is assumed: # of A is 4, # of AAAA is 4 (green)
```

Figure 2

```
% perl respsize.pl ns-ext.isc.org ns.psg.com ns.ripe.net ns.eu.int
ns-ext.isc.org requires 16 bytes
ns.psg.com requires 12 bytes
ns.ripe.net requires 13 bytes
ns.eu.int requires 11 bytes
# of NS: 4
For maximum size query (255 byte):
    only A is considered:        # of A is 4 (green)
    A and AAAA are considered:   # of A+AAAA is 3 (yellow)
    preferred-glue A is assumed: # of A is 4, # of AAAA is 2 (yellow)
For average size query (64 byte):
    only A is considered:        # of A is 4 (green)
    A and AAAA are considered:   # of A+AAAA is 4 (green)
    preferred-glue A is assumed: # of A is 4, # of AAAA is 4 (green)
```

Figure 3

Here we use the term "green" if all address records could fit, or "yellow" if two or more could fit, or "orange" if only one could fit, or "red" if no address record could fit. It's clear that without a

common parent for name server names, much space would be lost. For these examples we use an average/common name size of 15 octets, befitting our assumption of "GTLD-SERVERS.NET." as our common parent name.

We assume a medium query name size of 64 since that is the typical size seen in trace data at the time of this writing. If Internationalized Domain Name (IDN) or any other technology that results in larger query names be deployed significantly in advance of EDNS, then new measurements and new estimates will have to be made.

5. Conclusions

The current practice of giving all name server names a common parent (such as "GTLD-SERVERS.NET." or "ROOT-SERVERS.NET.") saves space in DNS responses and allows for more name servers to be enumerated than would otherwise be possible, since the common parent domain name only appears once in a DNS message and is referred to via "compression pointers" thereafter.

If all name server names for a zone share a common parent, then it is operationally advisable to make all servers for the zone thus served also be authoritative for the zone of that common parent. For example, the root name servers (?.ROOT-SERVERS.NET.) can answer authoritatively for the ROOT-SERVERS.NET. zone. This is to ensure that the zone's servers always have the zone's name servers' glue available when delegating, and will be able to respond with answers rather than referrals if a requester who wants that glue comes back asking for it. In this case the name server will likely be a "stealth master" -- authoritative but not advertised in the glue zone's NS RRSet. See Section 2 of [RFC1996] for more information about stealth masters.

Thirteen (13) is the effective maximum number of name server names usable with traditional (non-extended) DNS, assuming a common parent domain name, and given that implicit referral response truncation is undesirable in the average case.

More than one address record in a protocol family per server is inadvisable since the necessary glue RRSets (A or AAAA) are atomically indivisible, and will be larger than a single resource record. Larger RRSets are more likely to lead to or encounter truncation.

More than one address record across protocol families is less likely to lead to or encounter truncation, partly because multiprotocol clients, which are required to handle larger RRSets such as AAAA RRs, are more likely to speak EDNS, which can use a larger UDP response size limit, and partly because the resource records (A and AAAA) are in different RRSets and are therefore divisible from each other.

Name server names that are at or below the zone they serve are more sensitive to referral response truncation, and glue records for them should be considered "more important" than other glue records, in the assembly of referral responses.

6. Security Considerations

The recommendations contained in this document have no known security implications.

7. IANA Considerations

This document has no IANA actions.

8. Acknowledgements

The authors thank Peter Koch, Rob Austein, Mark Andrews, Kenji Rikitake, Stephane Bortzmeyer, Olafur Gudmundsson, Alfred Hoenes, Alexander Mayrhofer, and Ray Bellis for their valuable comments and suggestions.

This work was supported by the US National Science Foundation (research grant SCI-0427144) and DNS-OARC.

9. References

9.1. Normative References

[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.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

9.2. Informative References

[PERL] Wall, L., Christiansen, T., and J. Orwant, "Programming Perl, 3rd ed.", ISBN 0-596-00027-8, July 2000.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

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

[RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, August 1996.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

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

[RFC2672] Crawford, M., "Non-Terminal DNS Name Redirection", RFC 2672, August 1999.

[RFC3226] Gudmundsson, O., "DNSSEC and IPv6 A6 aware server/resolver message size requirements", RFC 3226, December 2001.

[RFC3258] Hardie, T., "Distributing Authoritative Name Servers via Shared Unicast Addresses", RFC 3258, April 2002.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC3901] Durand, A. and J. Ihren, "DNS IPv6 Transport Operational Guidelines", BCP 91, RFC 3901, September 2004.

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

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", RFC 4472, April 2006.

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

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

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

[RFC7129] Gieben, R. and W. Mekking, "Authenticated Denial of Existence in the DNS", RFC 7129, February 2014.

[SAC035] Bellis, R. and L. Phifer, "Test Report: DNSSEC Impact on Broadband Routers and Firewalls", SAC 035, September 2008.

Appendix A. The response simulator program

```
#!/usr/bin/perl
#
# SYNOPSIS
#    respsize.pl [ -z zone ] fqdn_ns1 fqdn_ns2 ...
#        if all queries are assumed to have a same zone suffix,
#     such as "jp" in JP TLD servers, specify it in -z option
#
use strict;
use Getopt::Std;

my ($sz_msg) = (512);
my ($sz_header, $sz_ptr, $sz_rr_a, $sz_rr_aaaa) = (12, 2, 16, 28);
my ($sz_type, $sz_class, $sz_ttl, $sz_rdlen) = (2, 2, 4, 2);
my (%namedb, $name, $nssect, %opts, $optz);
my $n_ns = 0;

getopt('z', %opts);
if (defined($opts{'z'})) {
    server_name_len($opts{'z'}); # just register it
}

foreach $name (@ARGV) {
    my $len;
    $n_ns++;
    $len = server_name_len($name);
    print "$name requires $len bytes\n";
    $nssect += $sz_ptr + $sz_type + $sz_class + $sz_ttl
            +  $sz_rdlen + $len;
}
print "# of NS: $n_ns\n";
arsect(255, $nssect, $n_ns, "maximum");
arsect(64, $nssect, $n_ns, "average");

sub server_name_len {
    my ($name) = @_;
    my (@labels, $len, $n, $suffix);

    $name =~ tr/A-Z/a-z/;
    @labels = split(/\./, $name);
    $len = length(join('.', @labels)) + 2;
    for ($n = 0; $#labels >= 0; $n++, shift @labels) {
        $suffix = join('.', @labels);
        return length($name) - length($suffix) + $sz_ptr
            if (defined($namedb{$suffix}));
        $namedb{$suffix} = 1;
    }
```

```
        return $len;
    }

    sub arsect {
        my ($sz_query, $nssect, $n_ns, $cond) = @_;
        my ($space, $n_a, $n_a_aaaa, $n_p_aaaa, $ansect);
        $ansect = $sz_query + $sz_type + $sz_class;
        $space = $sz_msg - $sz_header - $ansect - $nssect;
        $n_a = atmost(int($space / $sz_rr_a), $n_ns);
        $n_a_aaaa = atmost(int($space
                               / ($sz_rr_a + $sz_rr_aaaa)), $n_ns);
        $n_p_aaaa = atmost(int(($space - $sz_rr_a * $n_ns)
                               / $sz_rr_aaaa), $n_ns);
        printf "For %s size query (%d byte):\n", $cond, $sz_query;
        printf "    only A is considered:        ";
        printf "# of A is %d (%s)\n", $n_a, &judge($n_a, $n_ns);
        printf "    A and AAAA are considered:   ";
        printf "# of A+AAAA is %d (%s)\n",
               $n_a_aaaa, &judge($n_a_aaaa, $n_ns);
        printf "    preferred-glue A is assumed: ";
        printf "# of A is %d, # of AAAA is %d (%s)\n",
            $n_a, $n_p_aaaa, &judge($n_p_aaaa, $n_ns);
    }

    sub judge {
        my ($n, $n_ns) = @_;
        return "green" if ($n >= $n_ns);
        return "yellow" if ($n >= 2);
        return "orange" if ($n == 1);
        return "red";
    }

    sub atmost {
        my ($a, $b) = @_;
        return 0 if ($a < 0);
        return $b if ($a > $b);
        return $a;
    }
```

Appendix B. Editorial Notes

This section (and sub-sections) to be removed prior to publication.

B.1. Change History

15 Draft resurrected; Joe added as co-author; changed Paul's affiliation. Minor wordsmithing to account for the passage of time. Terminology section added. Added commentary on DNSSEC impact on response sizes and EDNS support.

Authors' Addresses

Paul Vixie
Farsight Security, Inc.
155 Bovet Road, #476
San Mateo, CA 94402
USA

Phone: +1 650 489 7919
Email: vixie@farsightsecurity.com

Akira Kato
Keio University/WIDE Project
Graduate School of Media Design
4-1-1 Hiyoshi
Kohoku, Yokohama 223-8526
Japan

Phone: +81 45 564 2490
Email: kato@wide.ad.jp

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

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