Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: March 14, 2018

L. Song
Beijing Internet Institute
September 10, 2017

ATR: Additional Truncated Response for Large DNS Response
draft-song-atr-large-resp-00

Abstract

As the increasing use of DNSSEC and IPv6, there are more public evidence and concerns on IPv6 fragmentation issues due to larger DNS payloads over IPv6. This memo introduces an simple improvement on authoritative server by replying additional truncated response just after the normal large response.

REMOVE BEFORE PUBLICATION: The source of the document with test script is currently placed at GitHub [ATR-Github]. Comments and pull request are welcome.

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 March 14, 2018.

Copyright Notice

Copyright (c) 2017 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

Large DNS response is identified as a issue for a long time. It has been regarded mainly as a issue or limitation on authoritative server (delegation) as [I-D.ietf-dnsop-respsize] introduced. As the increasing use of DNSSEC and IPv6, there are more public evidence and concerns on resolver's suffering due to packets dropping caused by IPv6 fragmentation in DNS.

It is observed that some IPv6 network devices like firewalls intentionally choose to drop the IPv6 packets with fragmentation Headers[I-D.taylor-v6ops-fragdrop]. [RFC7872] reported more than 30% drop rates for sending fragmented packets. Regarding IPv6 fragmentation issue due to larger DNS payloads in response, one measurement [IPv6-frag-DNS] reported 37% of endpoints using IPv6-capable DNS resolver can not receive a fragmented IPv6 response over UDP.

Some workarounds and short-term solutions are proposed. One is to continue to keep the response within a safe boundary, 512 octets for IPv4 and 1232 octets for IPv6 (IPv6 MTU minus IPv6 header and UDP header). It avoids fragmentation, but it requires TCP and UDP applications to fit this limitation explicitly. Currently coordination between IP layer and upper layer still do not go well. For example the draft [I-D.andrews-tcp-and-ipv6-use-minmtu] viewed it as a problem that TCP fails to respect IPV6_USE_MIN_MTU.

Still, some cases are hard to avoid, for example the coming KSK rollover which will produce 1424 octets DNS response containing the new key and signature. To encounter this problem, some root servers (A, B, G and J) implemented countermeasures by truncating the

response once the large IPv6 packet surpasses 1280 octets [root-stars]. But it is reported that 17% resolvers is not capable to send query via TCP [IPv6-frag-DNS] (It is also possbile that the middle boxes drop the tcp queries). It becomes a dilemma to choose hurting the users who can not receive fragmentation or users without TCP capacity.

To relieve the dilemma in short term, this memo introduces an small improvement on DNS responding process by replying Additional Truncated Response (ATR) just after the normal response. The original design of ENDS0 and Truncation mechanism for Large response are orthogonal. ATR intends to decouple the two. In ATR EDNS0 and TCP fall-back can work independently according to Authoritative server's requirement.

ATR targets to relieve the hurt of resolver (both stub and recursive resolver) from the position of server (both authoritative and recursive server). It does not require any changes on resolver and has a deploy-and-gain feature to encourage operators to implement it to benefit their resolvers.

ATR can be also used as a measurement tool for those operators who would like to know how much and which resolvers can not receive IPv6 fragmented response. They can turn on the ATR function occasionally and record the TCP connection it received during the period. The data may be helpful to do some fine-grained analysis between different NS servers and provide ATR to specific group of resolvers.

Note that the methodology of ATR can be extended to support other transport protocol like DNS over HTTP(s), DNS over QUIC, if they become one optional transport for DNS.

2. EDNS0 and DNS TCP

DNS has an inherent mechanism defined in [RFC1035] to handle large DNS response by indicating (set TrunCation bit) the resolver to fall back to query via TCP. However, due to the fear of cost of TCP, TCP fall-back in DNS was in negative position from the very beginning of DNS. people had to seek another way to handle large DNS response.

EDNS(0) [RFC2671] was introduced as a cure for the issue of large DNS response and TCP fall back firstly in 1999 and obsoleted by [RFC6891] in 2013. The basic idea of EDNS(0) is to introduce a channel for resolver and authoritative server to negotiate an appropriate DNS payload size in end-to-end approach.

The intention of EDNS(0) is to avoid TCP fall back. So the use of EDNS(0) make TCP fall-back rare, which in turn gives people a wrong

implication that EDNS(0) is more advanced than DNS TCP and DNS TCP is not necessary if EDNS(0) is already supported for both resolver and authoritative server. Plus the fear of "poor" TCP performance, DNS TCP function is stripped even for modern DNS implementations. An measurement study [Not-speak-TCP]showed that about 17% of resolvers in the samples can not ask a query in TCP when they receive truncated response.

Ironically today when TCP is recalled as a solutions to large DNS response, the installed base of resolver without TCP function (or the middle box stops DNS TCP connections) become a real issue which should be consider.

3. The ATR mechanism

The ATR mechanism is very simple that it involves a ATR module in the responding process of current DNS implementation . As show in the following diagram the ATR module is right after truncation loop if the packet is not going to be fragmented.

```
A DNS query +-------------+        +-------------+
            |             | No     |             |  Normal response
     +------>  Truncation +-------->     ATR     +------------->
            |    loop     |        |    Module   |
            | truncation? |        | truncation? |
            +-------------+        +-------------+
                yes|                   yes|         +-----+
                   |                      +---------+timer+---->
                   |                                +-----+
                   |                          Truncated Response
                   +----------------------------->
                    Truncated Response
```

Figure 1: High-Level Testbed Components

The ATR responding process goes as follows:

o 1) When an authoritative server receives a query and enters the responding process, it first go through the normal truncation loop to see whether the size of response surpasses the EDNS0 payload size. If yes, it ends up with responding a truncated packets. If no, it enters the ATR module.

- 2) In ATR module, similar like truncation loop, the size of response is compared with a fixed size. If the response of a query is larger than a certain value, 1220 octets for example, the server firstly sends the normal response and then coin a truncated response with the same ID of the query.

- 3) The server can send the coined truncated response in not time. But considering the possibility of network reordering, it is suggested a timer to delay the second truncated response to around 10 millisecond which can be configured by local operation.

There are three cases when ATR are deployed in the authoritative sever:

- Case 1: A resolver (or sub-resolver) will receive both the large response and a very small truncated response in sequence. It will happily accepts the first response and drop the second one because the transaction is over.

- Case 2: In case a fragment is dropped in the middle, the resolver will end up with only receiving the small truncated response. It will retry using TCP in no time.

- Case 3: For those (probably 30%*17% of them) who can not speak TCP and sitting behind a firewall stubbornly dropping fragments. Just say good luck to them!

Especially regarding the coming KSK rollover, if the root server implements ATR rather than setting IPv6-edns-size to 1220 octets, it would be helpful for resolver without TCP capacity, because it still has a fair chance to receive the large response.

As to case 2, there is one performance consideration on resolver side. It is about how resolver react to ATR when it receives only the truncated response. They can choose TCP right away or wait other NS servers to respond. Normally the fragments are dropped in the ASes along the path. A different NS server with different path may avoid "bad" ASes. But in the extreme case, implementation may first try UDP queries with all NS servers, but all fail due to the dropped fragments. It may end up with "no servers could be reached" or revert automatically to TCP which also introduce delay. So if allowed by local policy, a diligent resolver can also emit queries via both channels.

4. Security Considerations

There may be concerns on DDoS attack problem due to the fact that the ATR introduces multiple responses from authoritative server. DNS cookies [RFC7873] and RRL on authoritative may be possible solutions

5. Author's Commnets

REMOVE BEFORE PUBLICATION:

When drafting this proposal,there is a question in author's mind about the benefit of ATR which may be too trivial to implement. Resolver can retire many times(12 times for root) to other NS servers if one path to particular server failed. The performance comparison between retries with other NS server and ATR is hard to measure. But it is still valuable in two cases:

1) For those server (like root) implemented or plan to implement "always-truncation" for large packets, they can benefit from not doing unnecessary TCP fall back.

2) For those area or countries where only one or two NS servers instance are deployed (root in China for example), stick to the local root server (with around 10ms latency for UDP and roughly around 30ms for TCP) is better than select another NS server far away (with around 200ms latency)

6. IANA considerations

No IANA registration work is required for the time being

7. Acknowledgments

8. References

[ATR-Github]
"XML source file and test script of DNS ATR", September 2017, <https://github.com/songlinjian/DNS_ATR>.

[I-D.andrews-tcp-and-ipv6-use-minmtu]
Andrews, M., "TCP Fails To Respect IPV6_USE_MIN_MTU", draft-andrews-tcp-and-ipv6-use-minmtu-04 (work in progress), October 2015.

[I-D.ietf-dnsop-respsize]
Vixie, P., Kato, A., and J. Abley, "DNS Referral Response Size Issues", draft-ietf-dnsop-respsize-15 (work in progress), February 2014.

[I-D.taylor-v6ops-fragdrop]
Jaeggli, J., Colitti, L., Kumari, W., Vyncke, E., Kaeo, M., and T. Taylor, "Why Operators Filter Fragments and What It Implies", draft-taylor-v6ops-fragdrop-02 (work in progress), December 2013.

[IPv6-frag-DNS]
"Dealing with IPv6 fragmentation in the DNS", August 2017, <https://blog.apnic.net/2017/08/22/dealing-ipv6-fragmentation-dns>.

[Not-speak-TCP]
"A Question of DNS Protocols", August 2013, <https://labs.ripe.net/Members/gih/a-question-of-dns-protocols>.

[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>.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, DOI 10.17487/RFC2671, August 1999, <https://www.rfc-editor.org/info/rfc2671>.

[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>.

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <https://www.rfc-editor.org/info/rfc7872>.

[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>.

[root-stars]
"Scoring the DNS Root Server System", November 2016, <https://blog.apnic.net/2016/11/15/scoring-dns-root-server-system/>.

[SAC016] ICANN Security and Stability Advisory Committee, "Testing Firewalls for IPv6 and EDNS0 Support", 2007.

[SAC035] ICANN Security and Stability Advisory Committee, "DNSSEC Impact on Broadband Routers and Firewalls", 2008.

Author's Address

Linjian Song
Beijing Internet Institute
Floor-2, Building-5, Digital Planet, Courtyard-58, Jing Hai Wu Lu, BDA
Beijing 101111
P. R. China

Email: songlinjian@gmail.com
URI: http://www.biigroup.com/