DNS transport issues

draft-fujiwara-dnsop-dns-transport-issue-00.txt

Status of this Memo

Copyright Notice

Abstract

This memo describes DNS transport issues in DNS shared unicast environment. Recently, many root DNS servers and some TLD servers

have introduced DNS shared unicast technique for DNS authoritative services, this may cause some problems.

1. Introduction

In DNS, There are roughly three kinds of DNS communications, recursive query resolution from stub resolver to iterative server, iterative queries from iterative server to authoritative server, and zone transfer between authoritative servers. This document mainly describes iterative queries from iterative server to authoritative server.

DNS uses three types of transports, basic UDP transport (limited to 512 octets) [RFC1035], TCP transport (over 512 and lower than 65536 octets) [RFC1035] and EDNS0 UDP transport[RFC2671].

Recently, many root DNS servers and some TLD servers use DNS shared unicast techniche[RFC3258] for DNS service. Shared unicast technique influences IP packet reachability between authoritative server and iterative server. this is described in section 2.

TCP transport needs high cost and inquiry failure brings an awful load to the iterative servers. It is necessary to think the ISP iterative servers should resolve the name and how much cost can be paid.

2. DNS transports under DNS shared unicast

Suppose communication between an iterative server which has one unique IP address and multiple shared unicast authoritative DNS servers which shares one IP address. To which real authoritative server reach the DNS query from the iterative server is selected by the routing protocol and may sometimes change. On the other hand, replies from the all authoritative servers which share one IP address allways reach the iterative server.

2.1 Basic UDP transport case

As described in RFC3258 section 2.5, this UDP transport has no problem.

2.2 EDNS0 UDP transport case

Any DNS query packet is smaller than 512 octets and fit in one UDP packet because DNS domainname is smaller than 256 octets. DNS response packet may be larger than path MTU, then DNS response packet mey be fragmented to multiple fragment packets.

A DNS query packet reaches one of shared authoritative servers and fragmented response packets returns to the iterative server. It works fine even if route flaps.

2.3 TCP transport case

As described in RFC3258 section 2.5, TCP transport may have problems. Without per packet load sharing, most queries over TCP session may sucess because DNS query session is short time and routes may be stable during DNS query session in most cases. With per packet load sharing, special cosideration is needed. But some transit ISPs use per packet load sharing in BGP4 routing. It is prohibited in RFC1771 BGP4 protocol. Transit ISPs is not under shared unicast DNS service provider.

As a result, TCP connection to shared unicast DNS server may fail frequently.

3. DNS packet size

As described in RFC3226 "DNSSEC and IPv6 A6 aware server/resolver message size requirements", DNSSEC compliant servers and resolvers MUST support EDNS0 and SHOULD advertise message size of 4000.

Recently, without DNSSEC, As a result of adding IPv6 AAAA glue RRs in the root zone and TLD zones, EDNS0 necessity has risen. EDNS0 message size of 4000 is enough in many cases.

But as described in [draft-fujiwara-bad-dns-auth], some people writes very large RRset which cannot be carried by 4000-octet-EDNS0, it is necessary to use TCP transport as last resort.

4. Other requirements

4.1 IPv6 fragmentation issue

As described in RFC2460 "IPv6 Specification" section 5, "the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU."

But EDNS0 needs to use IP fragmentation to avoid TCP.

4.2 pMTU discovery

Especially in IPv6 environment, it is necessary to consider pMTU discovery setting to pass larger data which need to be fragmented.

EDNS0 with fragmentation does not work well without pMTU discovery.

5. Iterative server cost-effectiveness

TCP transport needs high cost for both authoritative servers and

iterative servers. Iterative servers case, inquiry failure brings an awful load. It is necessary to consider the ISP iterative servers should resolve the name by TCP and how much cost can be paid.

As described in section 2.3, TCP queries may fail, it is necessary to consider frequent TCP failure to implement iterative server.

TBD

6. Future proposal

In the future, any DNS server MUST support EDNS0. Furthermore, it is not necessary to consider EDNS0 unaware iterative servers.

In the case, if any response from root/TLD zone is smaller than 4000 octets, the root/TLD authoritative servers need not answer TCP query.

TBD

7. Security considerations

TBD

References

[I-D. fujiwara-dnsop-bad-dns-auth] K. Fujiwara, K.Toyama, and K.Ishibashi, "DNS authoritative server misconfiguration and a countermeasure in resolver" draft-fujiwara-dnsop-bad-dns-auth-01 (work in progress), Oct. 2004.

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

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

[RFC1035] P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION, " RFC 1035, November 1987.

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

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

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

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

Authors' Addresses

Kazunori Fujiwara
Japan Registry Service Co.,Ltd.
Chiyoda First Bldg. East 13F,
3-8-1 Nishi-Kanda Chiyoda-ku,
Tokyo 101-0065, JAPAN
Phone: +81-3-5215-8451
E-Mail: fujiwara@jprs.co.jp

Acknowledgment

We would like to thank Ichiro Mizukoshi, Haruhiko Ohshima, Masahiro Ishino, Chika Yoshimura, Tsuyoshi Toyono, Hirotaka Matsuoka, Yasuhiro Morisita, and Bill Manning.

Funding for the RFC Editor function is currently provided by the Internet Society.