Simplified DNS Query under IPv4/IPv6 Mixed Environment

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 2011.

Copyright Notice

Copyright (c) 2011 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 (http://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.
Abstract

This document discusses a simplified regular DNS query (resolving from a domain name to IP address(es)) method under IPv4/IPv6 mixed environment.

With current DNS query method under IPv4/IPv6 mixed environment, it requires for a client to issue two pairs of DNS query/response messages (one for A(IPv4) record, the other for AAAA(IPv6) record) to obtain all of IPv4 and IPv6 addresses of the node with its One domain name argument.

It is clear that the current method (that requires to issue two pairs of DNS query/response messages for One domain name resolving) is not sophisticated and not optimized for current IPv4/IPv6 mixed environment.

The method was designed at the initial stage of IPv6 deployment (when IPv6 was minority). At that time, it was a best practical solution. However, now IPv6 is widely spreading. The current DNS query method is getting problematic and will never last to the future IPv6 fully deployed environment.

Goals of this document are:
1. to clarify the problems of current regular DNS query method
2. to propose a simplified regular DNS query method:
   ("One pair of DNS query/response messages for One domain name resolving")
   that is suitable for current IPv4/IPv6 mixed environment.

1. Introduction

This document discusses a simplified regular DNS query (resolving from a domain name to IP address(es)) method under IPv4/IPv6 mixed environment.

It is becoming common to set both IPv4 and IPv6 addresses to one node in current IPv4/IPv6 mixed environment. In typical or normal situation, the both IPv4 and IPv6 addresses information of the node are registered to the DNS DB. In other words; for the node (one domain name), both A (IPv4) record entry and AAAA (IPv6) record entry exist in the DNS DB.

Under such IPv4/IPv6 mixed environment, in order to obtain IPv4 and IPv6 addresses of the node with its one domain name argument by the DNS query(ies), it requires for a client to issue two pairs of DNS query/response messages (one for A(IPv4) record, the other for AAAA(IPv6) record).
In shortly to say the current DNS query method: "Two pairs of DNS query/response messages are required for One domain name resolving."

There are several historical reasons why we use this Two pairs of DNS query/response messages method. However, it has proved until now that the Two pairs method is complicated, inefficient and problematic. It is clear that the Two pairs of messages method is not suitable and not optimized for current IPv4/IPv6 mixed environment, and this method will never last to the future IPv6 fully deployed environment.

Goals of this document are the followings:
1. to clarify the problems of current regular DNS query method
2. to propose a simplified regular DNS query method:
   ("One pair of DNS query/response messages for One domain name resolving")
   that is suitable for current IPv4/IPv6 mixed environment.

2. Analysis of current regular DNS query method

Hereafter, we will discuss the following typical example case on one node under IPv4/IPv6 mixed environment.

   Domain Name: "hostX"
   IPv4 addresses: p, q
   IPv6 addresses: s, t

All together four IP addresses are set to one node whose domain name is "hostX" and these addresses and domain name information is registered.

Fig. 1 shows how these IP addresses and domain name of this node are registered in the DNS DB (example of zone file configuration).

    hostX IN A  p(IPv4 address)
    hostX IN A  q(IPv4 address)
    hostX IN AAAA s(IPv6 address)
    hostX IN AAAA t(IPv6 address)

Fig. 1 DNS DB situation: both IPv4 and IPv6 addresses are registered as A and AAAA record entries respectively.
Fig. 2 shows typical current regular DNS query method. Two pairs of DNS query/response messages (one for A (IPv4) record entries, the other for AAAA (IPv6) record entries) are issued from a client in order to resolve One domain name "hostX".

An idea of the "Two pairs of DNS query/response messages" is invented from an idea (keep the existing DNS query for A record as it is, and add a new DNS query for AAAA record.).

Fig. 2 only shows one type of typical DNS query methods; order of DNS query/response messages: [1st for A record, 2nd for AAAA record], messages issued style: [serial]
With some OSes, DNS queries implementation methods are different. Table 1 shows the categorization of known implementation method types.

Table 1 current "Two pairs of DNS query/response messages" types

<table>
<thead>
<tr>
<th>Type</th>
<th>1st query</th>
<th>2nd query</th>
<th>style</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-AAAA Serial</td>
<td>for A record</td>
<td>for AAAA record</td>
<td>Serial</td>
</tr>
<tr>
<td>A-AAAA Parallel</td>
<td>for A record</td>
<td>for AAAA record</td>
<td>Parallel</td>
</tr>
<tr>
<td>AAAAA-A Serial</td>
<td>for AAAA record</td>
<td>for A record</td>
<td>Serial</td>
</tr>
<tr>
<td>AAAAA-A Parallel</td>
<td>for AAAA record</td>
<td>for A record</td>
<td>Parallel</td>
</tr>
</tbody>
</table>

Some OSes choose more complicated or advanced method (whether 2nd query is issued or not depends on the result of 1st query or configuration status of the node.)

3. Problems of the current Two pairs of DNS query/response messages method

Compared to the One pair of DNS query/response messages method, Two pairs of DNS query/response messages method has the following problems.

1. Complicated (not Simple enough)

   If either of Two pairs of messages is lost at somewhere, too complex message recover procedures are required because there are no acknowledgment-based confirmation mechanism in DNS query method and messages losing patterns are various.

2. Less efficient

   It is obvious that One pair of messages method is minimum and most efficient method. So, Two pairs of messages method becomes less efficient.

3. Takes long time to obtain the result

   After operation of 1st pair of messages is finished, operation for 2nd pair of messages is started. So, it takes long time to obtain the result.
4. Issues Twice much traffics

   Compared to one pair of messages, One pair of messages issues
twice much number of traffics.

As above shows, it is clear that Two pairs of DNS query/response
messages method is problematic and not recommendable. This method
will never last to the future IPv6 fully deployed environment.
Current Two pairs of DNS query/response messages method should be
improved.

One pair of DNS query/response messages for One domain name
resolving method solves all of above problems and we propose it in
the next section.

4. Proposed three types of "One pair of DNS query/response messages
   for One domain name resolving" methods.

    We can design the following three types of DNS query methods to
    achieve the goal "One pair of DNS query/response messages for One
domain name resolving".

    1. Two Existing Records (AAAA and A) Together type
    2. One New Special Record (e.g., AAAA+A) type
    3. One Existing Record (AAAA) w/ Mapped Transformation type
4.1 Two Existing Records (AAAA and A) Together type

Fig. 3 shows Two Existing Records (AAAA and A) Together type.

<table>
<thead>
<tr>
<th>DNS Server</th>
<th>Client (w/ PF_UNSPEC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;--------Queries 1-----</td>
</tr>
<tr>
<td>p(IPv4)</td>
<td>Name: hostX</td>
</tr>
<tr>
<td>q(IPv4)</td>
<td>Type: A</td>
</tr>
<tr>
<td>s(IPv6)</td>
<td>Name: hostX</td>
</tr>
<tr>
<td>t(IPv6)</td>
<td>Type: AAAA</td>
</tr>
<tr>
<td></td>
<td>== Answers 1===========&gt;</td>
</tr>
<tr>
<td></td>
<td>Name: hostX</td>
</tr>
<tr>
<td></td>
<td>Type: A</td>
</tr>
<tr>
<td></td>
<td>Addr: p(IPv4 addr)</td>
</tr>
<tr>
<td></td>
<td>Name: hostX</td>
</tr>
<tr>
<td></td>
<td>Type: A</td>
</tr>
<tr>
<td></td>
<td>Addr: q(IPv4 addr)</td>
</tr>
<tr>
<td></td>
<td>Name: hostX</td>
</tr>
<tr>
<td></td>
<td>Type: AAAA</td>
</tr>
<tr>
<td></td>
<td>Addr: s(IPv6 addr)</td>
</tr>
<tr>
<td></td>
<td>Name: hostX</td>
</tr>
<tr>
<td></td>
<td>Type: AAAA</td>
</tr>
<tr>
<td></td>
<td>Addr: t(IPv6 addr)</td>
</tr>
</tbody>
</table>

Fig. 3 Two Existing Records (AAAA and A) Together type

In this type, two existing records (AAAA and A) are set together to Queries.

Advantages:

Since existing records are used, it is not necessary to introduce new record type. This is naturally designed type.

Disadvantages:

Client side implementation modification and update are necessary. This method does not follow DNS query usage. (Two record types set together Queries DNS query examples are not known till now.)
4.2 One New Special Record (e.g., AAAA+A) type

Fig. 4 shows One New Special Record (e.g., AAAA+A) type

In this type, one new special record (e.g., AAAA+A) is introduced and set to Queries. (This function may became possible by introducing new "pseudo" Resource Record type instead of new record type.)

Advantages:

This method follows standard DNS query usage (Only One record type set Queries).

Disadvantages:

Special record type (e.g., AAAA+A) should be newly introduced. Client side implementation modifications are necessary.
4.3 One Existing Record (AAAA) w/ Mapped Transformation type

Fig. 5 shows One Existing Record (AAAA) w/ Mapped Transformation of A (IPv4) record entries type. This idea is completely new.

All of existing A (IPv4) record entries are assumed to be AAAA (IPv6) entries by using "IPv4 mapped IPv6 address" transformation.

If "IPv4 mapped IPv6 address" function is installed and deployed on Client, it is not necessary to install additional reverse transformation function on the Client side.

```
Fig. 5 One Existing Record (AAAA) w/ Mapped Transformation type

This method type can be called "IPv6(AAAA) all-round" type. All IPv4(A) record entries are assumed as "IPv4 mapped IPv6(AAAA)" record entries. All record entries are acquired by a client with One pair of AAAA query/response messages. Mapped address transformation from A to AAAA is happened on DNS server side (not on client side).
```
Advantages:

NO fatal problems are found.
NO client side implementation modifications are necessary.
Since existing record (AAAA) is used, no new record is not introduced.
This method follows standard DNS query usage (Only One record type set Queries).

Disadvantages:

Since non-popular function "IPv4 mapped IPv6 address" is used, this method may be understood as a rather tricky method.
In order to use this method, "IPv4 mapped IPv6 address" function should be fully deployed on clients’ side.

4.4 Summary of "One pair of DNS query/response messages for One domain name resolving" methods.

A: If you stand on long-term or ideal design viewpoint

"Two Existing Records (AAAA and A) Together" type or "One New Special Record (e.g., AAAA+A)" type will become a good method. However, there is a high wall to overcome. Various and huge number of existing clients’ resolver implementations are should be updated with this new specification.

B: If you stand on practical deployment viewpoint

"One Existing Record (AAAA) w/ Mapped Transformation" type will become a good method. Since this method type utilize IPv4 mapped IPv6 address transformation technique, this method is very unique and it may takes time to understand this method. However, we can not find fatal problems in this method. It is a very strong good feature that client side implementation modifications are not needed.

If we would like to deploy the "One pair of DNS query/response messages for One domain name resolving" function as soon as possible, "One Existing Record (AAAA) w/ Mapped Transformation" type method will become a best practical solution.
5. Security Considerations

The proposed DNS query methods in this document is based on simplification or natural extension of the current DNS query method. Security considerations of the proposed methods are as same as those of the current normal DNS query method. There are no newly occurred security considerations in this document.

6. IANA Considerations

This document describes a new special DNS record type in one of the proposed DNS query methods. So, the value of this new special DNS record type should be managed by the IANA.

References

Normative References


Authors’ Addresses

Hiroshi Kitamura
Service Platform Research Laboratories, NEC Corporation
(SC building 12F)1753, Shimonumabe, Nakahara-Ku, Kawasaki,
Kanagawa 211-8666, JAPAN
Graduate School of Information Systems,
University of Electro-Communications
5-1 Chofugaoka 1-Chome, Chofu-shi, Tokyo 182-8585, JAPAN
Phone: +81 44 431 7686
Fax: +81 44 431 7680
Email: kitamura@da.jp.nec.com

Shingo Ata
Graduate School of Engineering, Osaka City University
3-3-138, Sugimoto, Sumiyoshi-Ku, Osaka 558-8585, JAPAN
Phone: +81 6 6605 2191
Fax: +81 6 6605 2191
Email: ata@info.eng.osaka-cu.ac.jp