

Network Working Group  
Request for Comments: 4074  
Category: Informational

Y. Morishita  
JPRS  
T. Jinmei  
Toshiba  
May 2005

## Common Misbehavior Against DNS Queries for IPv6 Addresses

### Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2005).

### Abstract

There is some known misbehavior of DNS authoritative servers when they are queried for AAAA resource records. Such behavior can block IPv4 communication that should actually be available, cause a significant delay in name resolution, or even make a denial of service attack. This memo describes details of known cases and discusses their effects.

### 1. Introduction

Many existing DNS clients (resolvers) that support IPv6 first search for AAAA Resource Records (RRs) of a target host name, and then for A RRs of the same name. This fallback mechanism is based on the DNS specifications, which if not obeyed by authoritative servers, can produce unpleasant results. In some cases, for example, a web browser fails to connect to a web server it could otherwise reach. In the following sections, this memo describes some typical cases of such misbehavior and its (bad) effects.

Note that the misbehavior is not specific to AAAA RRs. In fact, all known examples also apply to the cases of queries for MX, NS, and SOA RRs. The authors believe this can be generalized for all types of queries other than those for A RRs. In this memo, however, we concentrate on the case for AAAA queries, since the problem is particularly severe for resolvers that support IPv6, which thus affects many end users. Resolvers at end users normally send A and/or AAAA queries only, so the problem for the other cases is relatively minor.

## 2. Network Model

In this memo, we assume a typical network model of name resolution environment using DNS. It consists of three components: stub resolvers, caching servers, and authoritative servers. A stub resolver issues a recursive query to a caching server, which then handles the entire name resolution procedure recursively. The caching server caches the result of the query and sends the result to the stub resolver. The authoritative servers respond to queries for names for which they have the authority, normally in a non-recursive manner.

## 3. Expected Behavior

Suppose that an authoritative server has an A RR but has no AAAA RR for a host name. Then, the server should return a response to a query for an AAAA RR of the name with the response code (RCODE) being 0 (indicating no error) and with an empty answer section (see Sections 4.3.2 and 6.2.4 of [1]). Such a response indicates that there is at least one RR of a different type than AAAA for the queried name, and the stub resolver can then look for A RRs.

This way, the caching server can cache the fact that the queried name has no AAAA RR (but may have other types of RRs), and thus improve the response time to further queries for an AAAA RR of the name.

## 4. Problematic Behaviors

There are some known cases at authoritative servers that do not conform to the expected behavior. This section describes those problematic cases.

### 4.1. Ignore Queries for AAAA

Some authoritative servers seem to ignore queries for an AAAA RR, causing a delay at the stub resolver to fall back to a query for an A RR. This behavior may cause a fatal timeout at the resolver or at the application that calls the resolver. Even if the resolver eventually falls back, the result can be an unacceptable delay for the application user, especially with interactive applications like web browsing.

### 4.2. Return "Name Error"

This type of server returns a response with RCODE 3 ("Name Error") to a query for an AAAA RR, indicating that it does not have any RRs of any type for the queried name.

With this response, the stub resolver may immediately give up and never fall back. Even if the resolver retries with a query for an A RR, the negative response for the name has been cached in the caching server, and the caching server will simply return the negative response. As a result, the stub resolver considers this to be a fatal error in name resolution.

Several examples of this behavior are known to the authors. As of this writing, all have been fixed.

#### 4.3. Return Other Erroneous Codes

Other authoritative servers return a response with erroneous response codes other than RCODE 3 ("Name Error"). One such RCODE is 4 ("Not Implemented"), indicating that the servers do not support the requested type of query.

These cases are less harmful than the previous one; if the stub resolver falls back to querying for an A RR, the caching server will process the query correctly and return an appropriate response.

However, these can still cause a serious effect. There was an authoritative server implementation that returned RCODE 2 ("Server failure") to queries for AAAA RRs. One widely deployed mail server implementation with a certain type of resolver library interpreted this result as an indication of retry and did not fall back to queries for A RRs, causing message delivery failure.

If the caching server receives a response with these response codes, it does not cache the fact that the queried name has no AAAA RR, resulting in redundant queries for AAAA RRs in the future. The behavior will waste network bandwidth and increase the load of the authoritative server.

Using RCODE 1 ("Format error") would cause a similar effect, though the authors have not seen such implementations yet.

#### 4.4. Return a Broken Response

Another type of authoritative servers returns broken responses to AAAA queries. Returning a response whose RR type is AAAA with the length of the RDATA being 4 bytes is a known behavior of this category. The 4-byte data looks like the IPv4 address of the queried host name.

That is, the RR in the answer section would be described as follows:

```
www.bad.example. 600 IN AAAA 192.0.2.1
```

which is, of course, bogus (or at least meaningless).

A widely deployed caching server implementation transparently returns the broken response (and caches it) to the stub resolver. Another known server implementation parses the response by itself, and sends a separate response with RCODE 2 ("Server failure").

In either case, the broken response does not affect queries for an A RR of the same name. If the stub resolver falls back to A queries, it will get an appropriate response.

The latter case, however, causes the same bad effect as that described in the previous section: redundant queries for AAAA RRs.

#### 4.5. Make Lame Delegation

Some authoritative servers respond to AAAA queries in a way that causes lame delegation. In this case, the parent zone specifies that the authoritative server should have the authority of a zone, but the server should not return an authoritative response for AAAA queries within the zone (i.e., the AA bit in the response is not set). On the other hand, the authoritative server returns an authoritative response for A queries.

When a caching server asks the server for AAAA RRs in the zone, it recognizes the delegation is lame, and returns a response with RCODE 2 ("Server failure") to the stub resolver.

Furthermore, some caching servers record the authoritative server as lame for the zone and will not use it for a certain period of time. With this type of caching server, even if the stub resolver falls back to querying for an A RR, the caching server will simply return a response with RCODE 2, since all the servers are known to be "lame."

There is also an implementation that relaxes the behavior a little bit. It tries to avoid using the lame server, but continues to try it as a last resort. With this type of caching server, the stub resolver will get a correct response if it falls back after Server failure. However, this still causes redundant AAAA queries, as explained in the previous sections.

## 5. Security Considerations

The CERT/CC pointed out that the response with RCODE 3 ("Name Error"), described in Section 4.2, can be used for a denial of service attack [2]. The same argument applies to the case of "lame delegation", described in Section 4.5, with a certain type of caching server.

## 6. Acknowledgements

Erik Nordmark encouraged the authors to publish this document as an RFC. Akira Kato and Paul Vixie reviewed a preliminary version of this document. Pekka Savola carefully reviewed a previous version and provided detailed comments. Bill Fenner, Scott Hollenbeck, Thomas Narten, and Alex Zinin reviewed and helped improve the document at the last stage for publication.

## 7. Informative References

- [1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [2] The CERT Coordination Center, "Incorrect NXDOMAIN responses from AAAA queries could cause denial-of-service conditions", March 2003, <<http://www.kb.cert.org/vuls/id/714121>>.

## Authors' Addresses

MORISHITA Orange Yasuhiro  
Research and Development Department, Japan Registry Services Co., Ltd.  
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda  
Chiyoda-ku, Tokyo 101-0065  
Japan

EMail: [yasuhiro@jprs.co.jp](mailto:yasuhiro@jprs.co.jp)

JINMEI Tatuya  
Corporate Research & Development Center, Toshiba Corporation  
1 Komukai Toshiba-cho, Saiwai-ku  
Kawasaki-shi, Kanagawa 212-8582  
Japan

EMail: [jinmei@isl.rdc.toshiba.co.jp](mailto:jinmei@isl.rdc.toshiba.co.jp)

## Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

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

