Misbehaving Name Servers and What They're Missing

by David Malone, Hamilton Institute, NUI Maynooth, Ireland

IPv6-capable hosts abound, and the number is growing. Evidence[1] shows that more than 2 million Windows XP machines are probing for 6to4[2] connectivity. When combined with deployments of Linux and BSD that have been shipping with IPv6 support enabled by default for some time, that is a sizable platform on which to build IPv6 applications. Most Web browsers (Internet Explorer, Mozilla, Opera) now support IPv6 if the underlying platform does, so that is a significant number of applications ready to start making IPv6 queries.

In fact, many of these applications are already looking for IPv6 addresses in the *Domain Name System* (DNS), even if IPv6 connectivity is not actually available. This usually does not result in a problem—the name server says there are no IPv6 records and the application falls back to IPv4. In a small number of cases, name servers running outdated or errant software are misbehaving when faced with a request for an IPv6 address.

The Problem

So, what problem are these name servers having with the request for IPv6 addresses? Well, the DNS stores different types of information, such as host names and addresses. Different types of data are stored using different record types. For example, IPv4 addresses are stored using a type "A" record and host names are stored using a type "PTR" record. Some new record types have been introduced for IPv6. The most important one is "AAAA," which is for storing IPv6 addresses. (Another type called "A6" was also introduced, but it is now consigned to experimental status because it proved too complicated in certain situations.)

When you issue a request to the DNS, you indicate the domain and type of record that you are interested in. If the server has records of that type for that domain, it replies, including those records. If the server has no records of that type, it should respond saying "there are no records of this type." If the domain does not exist, then the server should return a "no such domain" error.

However, the problems arise when the DNS server does something different, and some name servers behave badly when faced with a query for a type they do not explicitly know about. For the sake of simplicity, we will highlight three wrong reactions to an unknown query that have been observed. A more complete technical analysis of the problem can be found in[3].

The first reaction that people notice is that some name servers do not reply when faced with a query for an unknown type. In this case, the person who made the request waits a while before the request is reissued. Eventually the application falls back to IPv4. "Eventually" means anything from 10 seconds to 100 seconds, depending on the operating system and application—enough to irk the casual Web user.

The second reaction is more subtle. Here the name server returns a "no such domain" response. At first glance this may seem harmless enough—the query for an IPv4 address is issued quickly. However, DNS specifications say that the "no such domain" response may be cached. This means that the "A" query is never issued, and the system acts as if the domain does not exist.

The third reaction is that the server issues some other sort of incorrect response. Usually this is less serious than the two previous reactions, because other responses at worst result in a particular name server being considered "bad" and being avoided for future queries. This means that some better-behaved name server can answer the query.

The Extent of the Problem

Although sites with these problems are sometimes discussed on mailing lists, the extent of a problem is not always proportional to the coverage it receives. Historically, numerous online advertising companies have had load-balancing DNS servers that exhibit these symptoms. Because the content of an ad server is embedded in the Web pages of many organizations, this means a single errant DNS server can give the end user the impression that this problem is more widespread than it is.

To give some idea of the scale of the problem, Table 1 shows the results of querying the name servers for the names mentioned in a month's worth of Web proxy logs. The number of servers responding in each of the three ways mentioned (no reply, no such domain, or other error) is shown, along with a total. Also shown is the number of name servers that actually returned IPv6 addresses.

These results show that actually only a small number of name servers have this problem. Unfortunately, it also looks as if the number of name servers distributing IPv6 addresses is actually comparable. However, it does look like the proportion of problem name servers is decreasing over time.

Table 1: Responses to Name Queries

Nameservers that:	January 2004	April 2004	*August 2004*
Responded to type A	*16838*	*20631*	*17934*
Did not reply to type A	*64 (0.38%)*	*49 (0.24%)*	*36 (0.20%)*
Returned no such domain	*11 (0.07%)*	*19 (0.09%)*	*11 (0.06%)*
Returned other error	*22 (0.13%)*	*39 (0.19%)*	*11 (0.06%)*
Had any issue with AAAA	*97 (0.58%)*	*107 (0.52%)*	*58 (0.32%)*
Returned AAAA records	*105 (0.62%)*	*123 (0.60%)*	*18 (0.66%)*

Looking at Web logs to determine the size of the problem gives us a feeling for the number of name servers that need attention. Another interesting parameter to consider is the proportion of requests that might be subject to this problem. The answer would tell us how many queries might be mishandled if your name server cannot deal with new query types.

Looking at the queries for addresses at one authoritative name server shows that 65 percent of queries are for A records, 21 percent are for AAAA records, and 14 percent are for A6 records. Although this server is IPv6-capable and might attract more queries for AAAA records, even the root servers run by RIPE show that 10 percent of address queries are for IPv6 addresses.

The Solution

Some of the name servers that exhibit this problem are simply running old versions of DNS server software. If this is the case, then the fix is simple: *upgrade!*

A significant number of the remaining problem servers are running unusual name server software, and the only way to fix the problem is to have that software fixed. Where the name server software is maintained in house, there should be enough DNS expertise to resolve the issue when it is identified. Where DNS systems have been bought in, it can be difficult to get the relevant information to the developers who can make the necessary changes. Thus increasing awareness of the issue among DNS vendors and troubleshooters is important.

In some cases[5,6], discussions on Internet mailing lists has alerted those responsible for the server to the problem and the issue has been resolved. In other cases, feedback provided by users and customers has marked IPv6 conformance as an issue for future upgrades of a site’s DNS infrastructure. Unfortunately, on some occasions, feedback has been ignored and the problem has persisted. This is maybe not so surprising because it is a subtle problem. The fact that it is IPv6-related means it is sometimes dismissed because the organization thinks “we have not begun IPv6 deployment yet, so it cannot affect us.”

Where problems have persisted, people have resorted to various practical solutions (hacks?) to avoid the issue. Some people, who do not need IPv6 at this time, have just suppressed the AAAA queries. Others, when they discover a name server that times out, add it to a blacklist. This avoids any delays, but may make a site unavailable. Mozilla includes a more forgiving style of blacklisting, in the form of a “ipv4OnlyDomains” setting, that can be set to a list of domains known to have problems[7].

The long-term solution seems straightforward. As we have seen, the number of name servers exhibiting this problem is relatively small, though some do serve some often-queried domains. If we can ensure that no more servers with these problems get deployed, then as the existing servers are updated or retired the problem will be resolved.

To this end, it is worth testing new DNS deployments to make sure that they correctly respond to unusual query types[8]. This will smooth the path not just for IPv6, but also for other new technology such the *Domain Name System Security Extension* (DNSSEC)[9].

References

[1] "Observations of 6to4 Traffic on a 6to4 Router," Pekka Savola, preprint, October 2004.

[2] "Connecting IPv6 Routing Domains Over the IPv4 Internet," Carpenter, Moore, and Fink, *The Internet Protocol Journal*, Volume 3, No. 1, March 2000.

[3] "Common Misbehavior against DNS Queries for IPv6 Addresses," Y. Morishita and T. Jinmei, **draft-ietf-dnsop-misbehavior-against-aaaa-01.txt**, April 2004.

[4] K-Root information page, RIPE, **http://k.root-servers.org/**

[5] "**news.bbc.co.uk** NXDOMAIN problem fixed," itojun, Simon Lockhart, et al., 6bone mailing list, April 2002.

[6] "**ftp.perl.org** strangenes" thread, Mark Andrews, Ask Bjoern Hansen, et al., freebsd-stable mailing list, March 2004.

[7] "IPv6: Some IPv4 addresses won't resolve w/IPv6 OS," Mozilla bug 68796,
https://bugzilla.mozilla.org/show_bug.cgi?id=68796

[8] "AAAA lookup checker," David Malone,
http://www.cnri.dit.ie/cgi-bin/check_aaaa.pl

[9] "DNSSEC," Miek Gieben, *The Internet Protocol Journal*, Volume 7, No. 2, June 2004.

DAVID MALONE received B.A. (mod), M.Sc., and Ph.D. degrees from Trinity College Dublin. He has been involved with system administration since 1994 and has been slowly growing IPv6 networks since 1999, when he also became a FreeBSD committer. With Niall Murphy, he is the coatuthor of *IPv6 Network Administration*, ISBN 0-596-00934-8 published by O'Reilly and Associates, 2005. He is currently on secondment to the Hamilton Institute of NUI Maynooth. E-mail: **dwmalone@maths.tcd.ie**