IDN Deployment Test Results – Public Root

Lars-Johan Liman
Autonomica AB

Abstract

Autonomica AB has, under a contract with ICANN, investigated whether the addition of top level domains containing encoded internationalized characters (so called IDNs) to the public root zone for testing purposes has any impact on the iterative mode resolvers used to look up the information. No impact at all could be detected. All involved systems behaved exactly as expected.

$Id: idn-report-2.tex,v 1.4 2008/01/07 12:56:00 liman Exp $

Contents

1 Background

ICANN is responsible for the management of the name space in the highest level of the domain name system. ICANN wants to deploy a new type of top level domain in the public DNS system – domain names that contain encoded versions of names expressed with other characters than those in the English alphabet, so called "internationalized domain names" (IDNs).

ICANN has requested that tests external to ICANN be performed, to investigate what the technical impact on resolver (the DNS client side) of deploying such IDNs as top level domains in the public root would be. The scope of the test is limited to the addition of these domain *names*. There are no new record types or DNS "classes" involved, only the classical NS (name server) records for delegations, and the connected A (IPv4 address) records.

With IDNs, the domain names stored in the DNS servers are ordinary domain names just like before. The names stored have no special properties that makes it possible for the DNS servers to single out the IDN domains. There was no reason to believe that IDNs would make the DNS system as a whole behave in a different way than it normally does. Nevertheless, for prudence ICANN asked that it be tested that this assumption was true.

It should also be noted, that the root name servers play a very small but crucial role in the DNS system. In principle the root name servers only return two types of valid answers to valid DNS queries:

To queries about domain names in existing top level domains, the root name server will return a "referral" to a top level domain server ("go and talk to him over there"). For a query regarding a domain name in a non-existing top level domain, it will return a message indicating "non-existing domain" (NXDOMAIN). This is the expected behaviour. There are of course a number of cases where the query is totally invalid (broken), and in these cases other return messages may appear, such as refusal to handle the query (REFUSED), query format error (FORMERR), or "not implemented" (NOTIMP, which is used when the server *does* recognize the query, but is not equipped with the necessary software to handle it).

In the tests we have not generated any queries with format errors, as they have no special relation to IDNs.

The tests have been carried out on the public Internet and covered various implementations of client software to a degree that hopefully makes ICANN comfortable when making the decision whether to permanently add these IDN TLDs to the public DNS or not.

Autonomica AB was contracted to undertake such tests, and this is a presentation of the results of the tests. In the course of the tests we also measured the time it took to obtain the final response. Unexpected delays would have been noted.

2 System Setup

The test setup is described in a separate document: "IDN Deployment Test – Test Setup – Public Root".

The actual address plan used in the test is appended in Appendix B.

The tests were carried out in Autonomica's test facilities in Stockholm. The test system consisted of the following blocks

1. A number of different iterative mode resolvers (IMRs).
2. Query generator (client).
3. Connecting network.

The test system utilized the normal public root name servers for DNS, and the existing public TLD servers for the IDN test domains, provided by ICANN.

Only the qualitative performance of the root name servers and the iterative mode resolvers were evaluated in this test, according to the demarcations in the contract.

2.1 Root Name Servers – root

There was no special configuration with regards to root name service. The ordinary public root name servers were used, obtained from the public list at

`http://rs.internic.net/domain/named.cache`

The root zone published by the public root name servers is, temporarily, augmented with delegations for IDN test domains supplied by ICANN (see Appendix A). The domains are delegated to servers operated by ICANN on the public Internet.

2.2 Top Level Domain Name Server – TLD

The TLD servers for the IDN domains are operated by ICANN.

The TLD zones contain a DNS address record that can be queried for, to make the query process reach a final answer and come to full completion. All the TLD zones are stored on one and the same server – a name server serving multiple TLDs, as is commonly the case on the Internet.

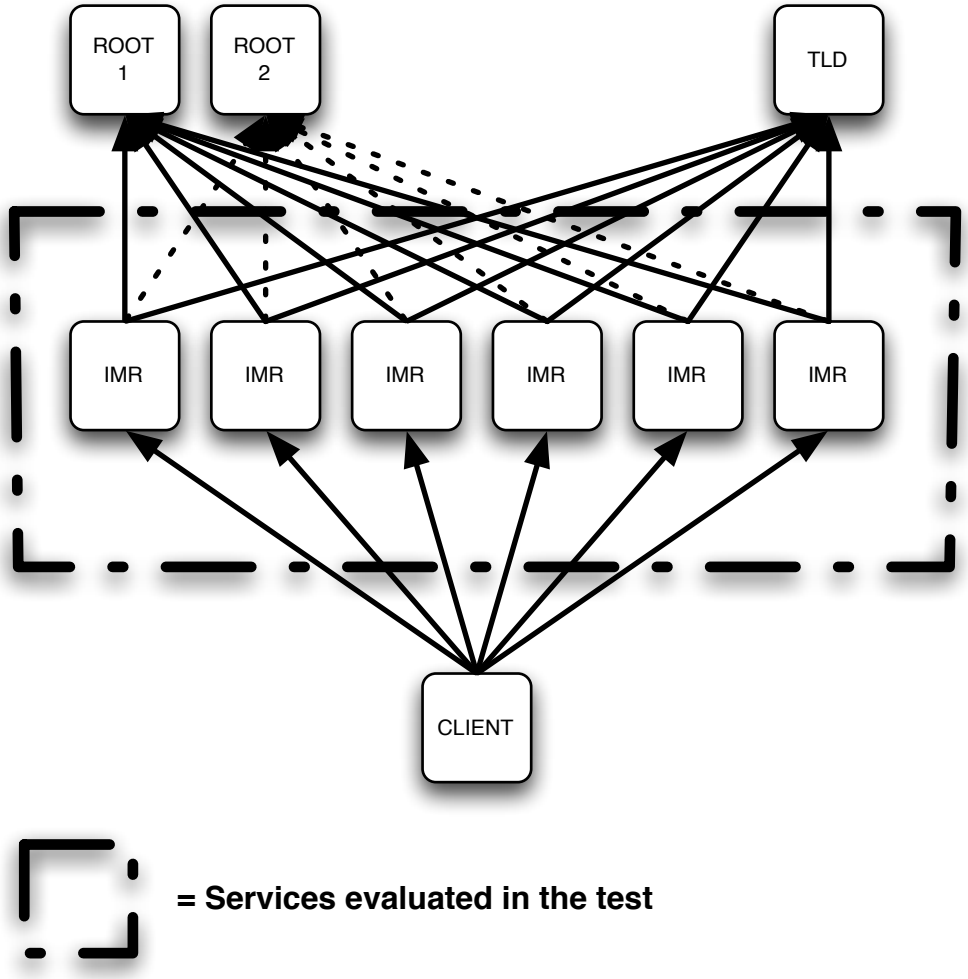

Figure 1: Service topology for the test

2.3 Iterative Mode Resolver – IMR

The IMRs were set up side by side on parallel servers. They were set up as "empty" iterative mode resolvers, where the only substantial configuration was which root name servers to use.

The IMRs were queried for the existing IDN address records, and the results were recorded for verification and comparison.

For comparison, the IMRs were also queried for non-existing domain names that look like IDNs. There were both domain names with non-existing terminal labels (the "host part" of the name), and with non-existing TLD labels.

The IMR software versions tested were:

1. ISC BIND version 8.3
2. ISC BIND version 8.4
3. ISC BIND version 9.0
4. ISC BIND version 9.1
5. ISC BIND version 9.2
6. ISC BIND version 9.3
7. Microsoft DNS Server as shipped in Windows 2000 Server
8. Microsoft DNS Server as shipped in Windows 2003 Server

The number of implementations of IMR software is vast. There was no reasonable way one could test all versions of all software on all platforms. To make this at all feasible, we had to limit ourselves to the most common platforms, which are various versions of ISC BIND, and various versions of Microsoft DNS servers. Apple Macintosh uses BIND, and most, if not all, Unix vendors ship BIND as their primary DNS server. There is a plethora of alternative server platforms, but they are counted in far smaller numbers than those above above.

Since we were looking for possibly broken software, we chose not to test the most recent versions of the software, but the most *ancient* versions of the most common minor versions of BIND, and the basic installations – without any service packs, upgrades, or patches – of the Windows 2000 Server and the Windows 2003 Server in the belief that the service packs improve the software, and we really want to test "worst case".

3 The Test

3.1 Installation

The software platforms were standard installations of operating systems without any special configuration. In the Microsoft cases, plain installa-

tions from distribution CDs using default configurations were used, except that we checked the checkbox for "install DNS server software", an obvious prerequisite for the test.

3.1.1 Root Servers

The normal root name servers on the public Internet were used.

3.1.2 The TLD Server

The test TLD servers operated by ICANN on the public Internet were used.

3.1.3 The Iterative Mode Resolvers

The various BIND versions were installed by compiling from source code (program text) according to the default installation procedure.

In a few cases, minor tweaks were necessary to make old versions of the code work on the more recent version of operating system of our choice. E.g., the names of some properties of the operating system have been changed, which made it necessary to make the same name changes to the program code. No functional changes were made.

The Windows nameserver code was included in the Windows system installation and required no extra steps.

3.1.4 The Client

The client used was an ordinary workstation connected to the same network as all the servers. The DNS debug tool "dig" (version 9.5.0a7) was used, as it provides a plethora of information about the DNS traffic while it acts as any other application when sending a DNS query.

4 The Process

4.1 Iterative Mode Resolvers

All IMRs where restarted before the first query set to ensure that their caches were empty and pristine.

4.2 Queries

Each of the IMRs was then sent several series of queries. The lists of domain names used for querying the IMRs were generated based on the list of IDN TLDs provided by ICANN. See below for more specific examples of queries.

4.3 Answer Processing

In all cases, the results were recorded on file and checked for any signals of bad process or bad data. Response times were noted in the process, to look for unexpected tardiness.

The tests were run twice – first with empty caches in the IMRs, then a short period of time later to investigate the responses from IMRs that are expected to find the answers in their caches.

5 Results

5.1 Phase One: Empty Cache

The following results were noted during the first phase of the test, using IMRs with empty caches:

5.1.1 Existing Terminal Nodes

The first series of queries was made up by looping through the existing IDN top level domain names, as provided by ICANN, querying for the IPv4 address record type (A) in the Internet class (IN), thus making a list like:

```
xn--mgbh0fb.xn--kgbechtv.                IN      A
xn--fsqu00a.xn--0zwm56d.                 IN      A
xn--fsqu00a.xn--g6w251d.                 IN      A
xn--hxajbheg2az3al.xn--jxalpdlp.         IN      A
xn--p1b6ci4b4b3a.xn--11b5bs3a9aj6g.      IN      A
xn--r8jz45g.xn--zckzah.                  IN      A
xn--9n2bp8q.xn--9t4b11yi5a.              IN      A
xn--mgbh0fb.xn--hgbk6aj7f53bba.          IN      A
xn--e1afmkfd.xn--80akhbyknj4f.           IN      A
xn--zkc6cc5bi7f6e.xn--hlcj6aya9esc7a.    IN      A
xn--fdbk5d8ap9b8a8d.xn--deba0ad.         IN      A
```

This creates a list of existing terminal nodes in the delegated TLDs in the public DNS where the both labels are encoded according to IDN. The intent was to ensure that the entire process of following the referral from the root to the TLD, and actually retrieving the final data, works as intended.

The expected result was that the IMR would be able to find and return the terminal A records in reasonable time.

Results: *All answers were consistent with expected behaviour, and no unexpected delays were discovered.*

5.1.2 Non-existing Terminal Nodes

The second series of queries was made up by trying three domain names where characters in the first label (the "host part") had been garbled, to force the TLD servers to respond that the TLD does indeed exists, but the host within the TLD does not.

The queries issued were:

```
xn--n92bp8q.xn--9t4b11yi5a.         IN      A
xn-m-gbh0fb.xn--hgbk6aj7f53bba.     IN      A
xn--e1af.xn--80akhbyknj4f.          IN      A
```

The expected result was that the IMR would be able to find the TLD server(s) and return the status code `NXDOMAIN` (Non-existing Domain), an empty "answer" section, and an "SOA" record (Start of Authority) indicating that the answer had been obtained from one of ICANN's TLD servers.

Results: *All answers were consistent with expected behaviour, and no unexpected delays were discovered.*

5.1.3 Non-existing Terminal Top Level Domains

The last series of queries was made up by trying three domain names where characters in the last label (the TLD) had been garbled, to force the root name servers to respond that the TLD does not exist.

The queries issued were:

```
xn--9n2bp8q.xn--9tb411yi5a.          IN      A
xn--mgbh0fb.xn-h-gbk6aj7f53bba.      IN      A
xn--e1afmkfd.xn--80akhj4f.           IN      A
```

The expected result was that the IMR would have to go to the root name servers to try to find the information regarding the domain names in question and return the status code `NXDOMAIN` (Non-existing Domain), an empty "answer" section, and an "SOA" record (Start of Authority) indicating that the answer had been obtained from a root name server.

Results: *All answers were consistent with expected behaviour, and no unexpected delays were discovered.*

5.2 Phase Two: Cache Already Populated

The IMRs were left untouched for a short period of time – short enough for all DNS records to remain in the caches of the respective servers. The queries of phase one were then identically repeated, to investigate how the IMRs would behave when the sought data was already in their caches.

5.2.1 Existing Terminal Nodes

The same series of queries as in section 5.1.1 were sent.

The expected result was the same as in section 5.1.1.

Results: *All answers were consistent with expected behaviour, and no unexpected delays were discovered.*

5.2.2 Non-existing Terminal Nodes

The same series of queries as in section 5.1.2 were sent.

The expected result was the same as in section 5.1.2.

Results: *All answers were consistent with expected behaviour, and no unexpected delays were discovered.*

5.2.3 Non-existing Terminal Top Level Domains

The same series of queries as in section 5.1.3 were sent.

The expected result was the same as in section 5.1.3.

Results: *All answers were consistent with expected behaviour, and no unexpected delays were discovered.*

5.3 Timing

Unexpected delay in the handling of the queries would have been perceived as a quality degrading property.

During the first phase of the test – when caches were empty, and much data had to be fetched from the public Internet – the longest response time noted was 378 ms (0.378 seconds), which is reasonably well within the expected limits, noting that the packet return time to the furthermost of ICANN's servers from Autonomica, is measured at 180 ms (0.180 seconds).

During the second phase of the test – when data was already present in the caches of the IMRs – the longest response time noted was 1 ms (0.001 second), which simply has to be regarded as excellent.

5.4 Availability of Technical Details

The resulting output files from most of the test runs have been retained, and will be made available upon request. Please contact the author for access to these files. Also, if the reader has more specific technical questions, the author is happy to answer these to the best of his ability. Contact information below.

6 Conclusions

During these tests, we were unable to detect any deviation from normal behaviour at all in any part of the DNS system under test.

The addition of IDN test strings seems to have had no measurable effect at all on the qualitative performance of the test systems.

7 Author's Contact Information

The author of this document is Lars-Johan Liman, Senior Systems Specialist at Autonomica AB, Stockholm, Sweden. He can be reached at:

Autonomica AB
Att: Lars-Johan Liman
Bellmansgatan 30
SE-118 47 Stockholm
Sweden

Tel: +46–8–6158572
Fax: +46–8–4420967

E-mail: liman@autonomica.se

APPENDICES

A IDN Test Strings

The valid TLD test strings used in the test, as provided by ICANN and deployed in the public root zone:

```
xn--mgbh0fb.xn--kgbechtv.
xn--fsqu00a.xn--0zwm56d.
xn--fsqu00a.xn--g6w251d.
xn--hxajbheg2az3al.xn--jxalpdlp.
xn--p1b6ci4b4b3a.xn--11b5bs3a9aj6g.
xn--r8jz45g.xn--zckzah.
xn--9n2bp8q.xn--9t4b11yi5a.
xn--mgbh0fb.xn--hgbk6aj7f53bba.
xn--e1afmkfd.xn--80akhbyknj4f.
xn--zkc6cc5bi7f6e.xn--hlcj6aya9esc7a.
xn--fdbk5d8ap9b8a8d.xn--deba0ad.
```

B Address Plan

The IP addresses used in the test were the following:

IP address	Service function
Resolvers:	
192.71.80.243	IMR BIND version 8.3
192.71.80.244	IMR BIND version 8.4
192.71.80.248	IMR Microsoft Windows 2000 Server
192.71.80.249	IMR Microsoft Windows 2003 Server
192.71.80.250	IMR BIND version 9.0
192.71.80.251	IMR BIND version 9.1
192.71.80.252	IMR BIND version 9.2
192.71.80.253	IMR BIND version 9.3