IDN Deployment Test
Test Setup

Lars-Johan Liman
Autonomica AB

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 domains 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 two external parties test what the technical impact, on the DNS client side, of deploying such IDNs as top level domains in the public root would be. The tests are to be carried out in a closed lab environment, and expected to cover various implementations of server and client software, to a degree that makes ICANN comfortable when making the decision whether to add these IDN TLDs to the public DNS or not.

Autonomica has been contracted to undertake such testing, and this is a description of the system that will be used for these tests.

2 System Setup

The tests will be carried out in Autonomica's test facilities in Stockholm. The test system will consist of the following blocks

1. Two root name servers (root).

2. Top level domain authoritative server (TLD).

3. iterative mode resolver (IMR).

4. Query generator (client).

5. Connecting network.

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

2.1 Root Name Servers – root

The root name servers will run on a Unix platform. Two major DNS server implementations will be tested, known to be used by root server operators. They are
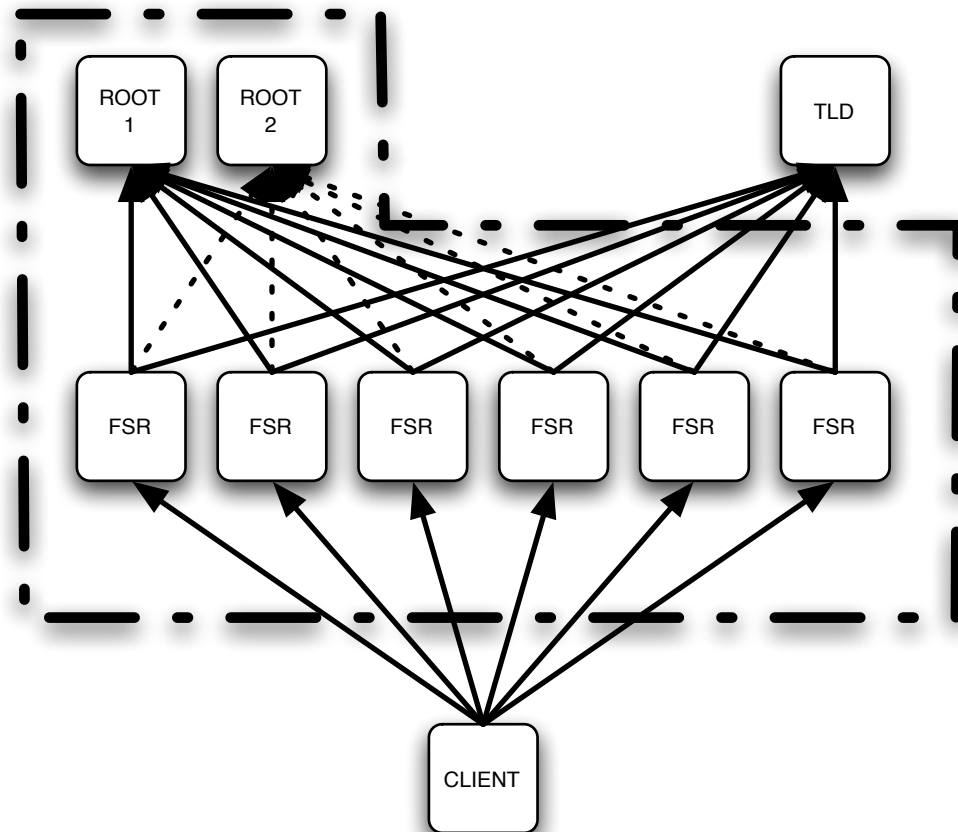
Figure 1: Service topology for the test

1. ISC BIND

2. NLNet Labs nsd

The reason for testing only these two, and only recent versions, is that the root name servers are few and operated by groups that have DNS as their main focus. Upgrading the root name servers to capable code is less overwhelming than upgrading the iterative mode resolvers. We will try to mimick the existing root name servers to the best of our abilities. Two servers will be operated to ensure that the IMR software doesn't have problems based on the fact that only a single server is reachable.

The root servers will contain a copy of the active root zone on the Internet, augmented with the IDN test domains supplied by ICANN, see Appendix A.

2.2 Top Level Domain Name Server – TLD

The Top Level Domain Name Server is not part of the test as such. It is involved only because the normal run of a DNS query (when there is no cached data to rely on) is to go to the root and follow the referral to the TLD. If the TLD server isn't in place, this top step cannot be completed, and the client will not return a complete answer. To make easy use of existing tools, and to make the testing smooth, the TLD will be set up with top level domain zones for each of the test domains provided by ICANN.

These zones will contain a few DNS records that can be queried for, for which the query process will complete, so that it can be studied in whole. All zones will be stored on one and the same server, as this server rôle will not be evaluated.

2.3 Iterative Mode Resolver – IMR

The rôle of the Iterative Mode Resolver is the most important one to test. The number of IMRs on the Internet is vast, and it is not easy to have them all upgraded. It is therefore important to understand what impact the addition of IDN top level domains may have on these servers.

The test system will have several IMRs which will all be configured to start with empty caches, and they will then be queried by the client for a number of existing IDN TLDs (as provided by ICANN) to test the exptected sucessful query process, and a number of non-existing ones (as invented by Autonomica staff) to verify that the IMRs handle the unsucessful queries gracefully.

The number of implementations of IMR software i vast. There is no reasonable way one can test all versions of all software on all platforms. To make this at all feasible, we have 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 are looking for possibly broken software, we have chosen 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 – 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".

Windows Vista server is not yet released, and Microsoft is undoubtedly aware of the IDN concept, so in case there are any minor problems, they can be fixed before the release.

Thus, the versions of software that will be tested are:

1. ISC BIND version 8.2 (hopefully)[1]

2. ISC BIND version 8.3

3. ISC BIND version 8.4

4. ISC BIND version 9.0

5. ISC BIND version 9.1

6. ISC BIND version 9.2

7. ISC BIND version 9.3

8. Microsoft DNS Server as shipped in Windows 2000 Server

9. Microsoft DNS Server as shipped in Windows 2003 Server

3 The Test Procedure

3.1 Installation

The first step will be to install operating systems on a number of machines, followed by installation of the DNS software for each machine. Standard installations, using default alternatives as far as possible, will be used.

---

[1]This software is so old that there are problems to make it work on the more recent versions of operating systems available today. We hope to succeed in our efforts to make it work.

Step two is to configure the root name servers with the root zone and the IDN delegation data.

The authoritative server will have the most complicated configuration, due to the fact that it will carry all the zones of the IDN test.

Setting up the IMRs is rather trivial, since they typically only need an "empty" configuration, with the one minor tweak that the root name server addresses have to be changed to the ones used in the testbed.

The final step is to set up the client. It will have a list of queries, which it will sent to each IMR. No nameserver software is neede on this machine. The software needed will be a DNS query generator, which will be a script that uses the commonly available tool "dig", which gives very good diagnostics containing every part of the received DSN answers.

3.2 Queries

As mentioned the list of domain names to query for will contain names for existing terminal nodes in the delegated TLDs, to ensure that the entire process of following the referral from the root to the TLD, and actually aqcuiring the final data, works as intended. To be able to identify possible failures in the process, the list will also contain queries for non-exising terminal nodes (to follow ther referral, but be unable to acquire the final data) and queries for non-existing IDN top level domain names in the root (to fail at an early stage).

The results will be checked for any signals of bad process or bad data. Response times will noted in the process, to look for unexpected tardiness.

The tests will be run several times, using different values on the TTL (Time To Live) values to investigate how the IMRs behave when records time out and are cleared from the cache.

4 Report

The results will be compiled in a report delivered to the IDN project at ICANN.

It is our goal to make the output from the test runs available on the Internet.

APPENDICES

A   IDN test strings

The list of TLD test strings to be used in the test, as provided by ICANN, is:

```
xn--18-7g4a9f
hippo18potamus
xn--18-xf0jl42g
xn--18-h31ew85n
xn--flod18hst-12a
xn--18-xsdrfd6ex1e
xn--18-dtd1bdi0h3ask
xn--18-28gg3ad5hl2fzb
xn--18-hmf0e1bza7dh8ioagd6n
xn--18-rjdbcud0neb9a8ce1ezef
xn--1818-63dcpd5be6bfqcecfadfad3dl
xn--1818-1goc0bacbac7eg2kh6ci9cj9bk4yla7ablb
xn--181818-qxecc5edd8aee8aebebecadeadead0fkkill5ymam
xn--flod18hstflod18hstflod18hstflod18hstflod18hstflod18-1iejjjj
hippo18potamushippo18potamushippo18potamushippo18potamus18hippo
```