Internet Draft
draft-ietf-idn-idnra-00.txt
August 17, 2000
Expires in six months

Paul Hoffman
IMC & VPNC
Patrik Faltstrom
Cisco

Internationalized Host Names Using Resolvers and Applications (IDNRA)

Status of this Memo

Abstract

The current DNS infrastructure does not provide a way to use internationalized host names (IDN). This document describes a mechanism that requires no changes to any DNS server that will allow internationalized host names to be used by end users with changes only to resolvers and applications. It allows flexibility for user input and display, and assures that host names that have non-ASCII characters are not sent to servers.

1. Introduction

In the discussion of IDN solutions, a great deal of discussion has focused on transition issues and how IDN will work in a world where not all of the components have been updated. Earlier proposed solutions require that user applications, resolvers, and DNS servers to be updated in order for a user to use an internationalized host name. Instead of this requirement for widespread updating of all components, the current proposal is that only user applications and the resolvers on user's systems be updated; no changes are needed to the DNS protocol or any DNS

servers. We also show that it is enough to update only the application, and at the same time an encoded version of the host name can be used even in current existing applications.

The proposal is called IDNRA because it only requires changes to resolvers and applications (the "R" and "A" in the name).

1.1 Design philosophy

To date, the proposals for IDN protocols have required that DNS servers be updated to handle internationalized host names. Because of this, the person who wanted to use an internationalized host name had to be sure that their request went to a DNS server that was updated for IDN. Further, that server could only send queries to other servers that had been updated for IDN because the queries contain new protocol elements to differentiate IDN name parts from current host parts. In addition, these proposals require that resolvers must be updated to use the new protocols, and in most cases the applications would need to be updated as well.

Updating all (or even a significant percentage) of the DNS servers in the world will be difficult, to say the least. Because of this, we have designed a protocol that requires no updating of any name servers. IDNRA still requires the updating of applications and resolvers, but once a user has updated these, she or he could immediately start using internationalized host names. The cost of implementing IDN would thus be much lower, and the speed of implementation will be much higher.

IDNRA also specifies how to use old applications and/or old resolvers in parallel with updated ones.

1.2 Terminology

The key words "MUST", "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.3 IDN summary

Using the terminology in [IDNCOMP], this protocol specifies an IDN architecture of arch-3 (just send ACE). The format is ace-1.2 (RACE), and the method for distinguishing ACE name parts from current name parts is ace-2.1.1 (add hopefully-unique legal tag). Because there is no changes needed to the DNS, the transition strategy is trans-1 (always do current plus new architecture).

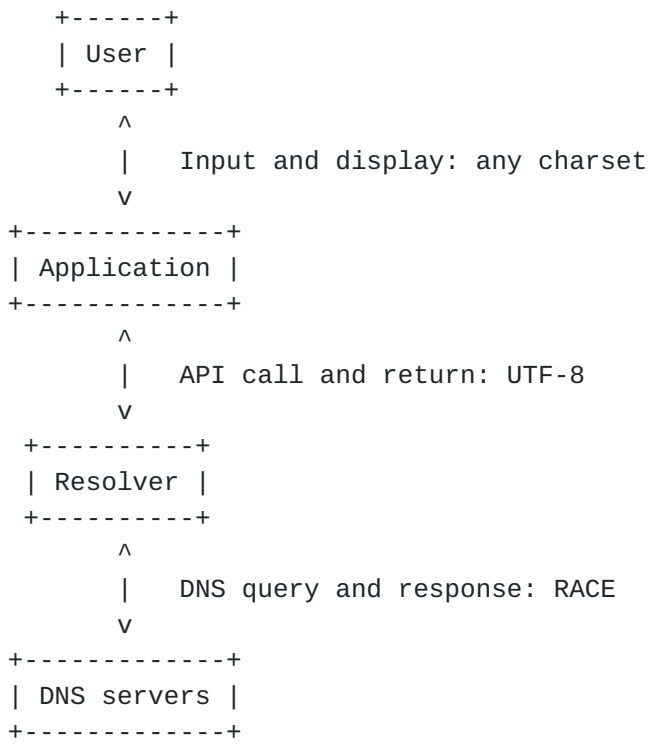
2. Structural Overview

In IDNRA, users' applications and resolvers are updated to perform the processing needed to input internationalized host names from users, display internationalized host names that are returned from the DNS to

users, and process the inputs and outputs from the DNS.

2.1 Interfaces between DNS components in IDNRA

The interfaces in IDNRA can be represented pictorially as:



2.1.1 Users and applications

Applications can accept host names using any character set or sets desired by the application developer, and can display host names in any charset. That is, this protocol does not affect the interface between users and applications.

An IDNRA-aware application can accept and display internationalized host names in two formats: the internationalized character set(s) supported by the application, and in RACE [RACE] ASCII-compatible encoding. Applications MAY allow RACE input and output, but are not encouraged to do so except as an interface for advanced users, possibly for debugging. RACE encoding is opaque and ugly, and should thus only be exposed to users who absolutely need it. The optional use, especially during a transition period, of RACE encodings in the user interface is described in section 3.

2.1.2 Applications and resolvers

Applications communicate with resolver libraries through a programming interface (API). Typically, the IETF does not standardize APIs, although it has for IPv6. This protocol does not specify a specific API, but instead specifies only the input and output formats of the host names to

the resolver library.

This protocol specifies that host names SHOULD be passed to resolvers using UTF-8 [RFC2279] because there are many libraries for converting between arbitrary charsets and UTF-8. However, because the API is not specified in this document, some resolvers may use different charsets for input and output, and applications must, of course, use the same charset as the resolver library they call.

IDNRA-aware applications MUST be able to work with both IDNRA-aware and non-aware resolvers. An IDNRA-aware application that is resolving a non-internationalized host name (one that conforms to RFC 1035[STD13]) MUST use non-aware APIs such as "gethostbyname" and "gethostbyaddr". An IDNRA-aware application that is resolving a internationalized host name (one that does not conform to RFC 1035) MUST use an API that is specific to IDNRA.

2.1.3 Resolvers and DNS servers

Before converting the name parts into RACE, the resolver MUST prepare each name part as specified in [NAMEPREP]. The resolver MUST use RACE ASCII-compatible encoding for the name parts that are sent in the DNS query, and will always get name parts encoded in RACE from the DNS service. DNS servers MUST use the RACE format for internationalized host name parts.

If a signalling system which makes negotiation possible between old and new DNS clients and servers is standardized in the future, the encoding of the query in the DNS protocol itself can be changed from RACE to something else, such as UTF-8. The question whether or not this should be used is, however, a separate problem and is not discussed in this memo.

3. Combinations of Resolvers and Applications

IDNRA allows non-IDNRA applications to coexist with IDNRA-aware resolvers, and non-IDNRA resolvers to coexist with IDNRA-aware applications. This section describes the interactions between applications and resolvers as users update each separately.

In this section, "old" means an application or resolver that has not bee upgraded to be IDNRA-aware, and "new" means an IDNRA-aware application or resolver. The two APIs are also called "old" and "new". "Binary" means any host name that is not compatible with current DNS character restrictions.

3.1 Old application, old resolver

Because it is an old resolver (and an old application), all host names MUST (and will) be resolved using the old API. A user cannot enter binary names in the application. A user MAY enter a name that uses RACE

encoding. Each RACE-encoded name part in such a name MUST already have had all name preparation done on it and be correctly converted to RACE encoding; otherwise, it will not be matched in the DNS.

When the resolver receives a RACE name in a response to a old API gethostbyaddr-type query, the resolver will not convert the host name to a binary form, and the application will thus display the name in RACE format. Showing the results of a gethostbyaddr-type queries is rare in typical Internet applications, so the display of RACE names is not likely in typical environments.

3.2 Old application, new resolver

Because it is an old application, all host names MUST (and will) be resolved using the old API. A user cannot enter binary names in the application. A user MAY enter a name that uses RACE encoding. Each RACE-encoded name part in such a name MUST already have had all name preparation done on it and be correctly converted to RACE encoding; otherwise, it will not be matched in the DNS. Note that, even though the resolver is new, the resolver MUST NOT do further name preparation on RACE-encoded name parts because the call was using the old API, which tells the resolver that the resolver is dealing with an old application.

If the resolver receives a RACE name in a response to a old API gethostbyaddr-type query, the resolver MUST NOT convert the host name to a binary form, and the application will thus display the name in RACE format. Showing the results of a gethostbyaddr-type queries is rare in typical Internet applications, so the display of RACE names is not likely in typical environments.

3.3 New application, old resolver

Because it is an old resolver, all host names MUST (and will) be resolved using the old API. If the user enters a binary host name, the application SHOULD reject the name as illegal. This is due to the fact that, if the application did not reject the name as illegal, the application would have to contain all of the name preparation logic and RACE-encoding logic, but that logic would only be used in the rare case where a user had updated applications but not the resolver. It is likely that applications would not fully implement and rigorously test the name preparation logic, and it is therefore likely that some applications in this scenario would give incorrect information to the user, and would possibly be susceptible to spoofing attacks. If an application is going allow the input of binary names and convert them to their RACE-encoded form for use on the old API, the application MUST do full name preparation exactly as it would have been done in a new resolver.

If the application receives a RACE-encoded name part in a response to a old API gethostbyaddr-type query, the application SHOULD convert the host name to a binary form for display. However, the application MAY have an interface that allows the display of RACE names that are

returned by gethostbyaddr-type queries, but the default setting of such an interface SHOULD be to show the binary form, not the RACE form.

3.4 New application, new resolver

All host names MUST be resolved using the new API. A user MAY enter a name that uses RACE encoding. Each RACE-encoded name part in such a name MUST already have had all name preparation done on it and be correctly converted to RACE encoding; otherwise, it will not be matched in the DNS.

When the resolver receives a RACE name in a response to a gethostbyaddr-type query, if the query was to the old API, the resolver MUST NOT convert the host name and MUST pass the RACE-formatted name to the application. If the query was to the new API, the resolver MUST convert the host name part to the binary form. The application MAY have an interface that allows the user to decide whether to use the old or new API, and therefore to show the results in RACE or binary format, but the default setting of such an interface SHOULD be to use the new API.

4. Root Server Considerations

Because there are no changes to the DNS protocols, adopting this protocol has no effect on the root servers.

5. Security Considerations

Much of the security of the Internet relies on the DNS. Thus, any change to the characteristics of the DNS can change the security of much of the Internet.

Host names are used by users to connect to Internet servers. The security of the Internet would be compromised if a user entering a single internationalized name could be connected to different servers based on different interpretations of the internationalized host name.

Because this document normatively refers to [NAMEPREP], it includes the security considerations from that document as well.

6. References

[IDNCOMP] Paul Hoffman, "Comparison of Internationalized Domain Name Proposals", draft-ietf-idn-compare.

[NAMEPREP] Paul Hoffman & Marc Blanchet, "Preparation of Internationalized Host Names", draft-ietf-idn-nameprep.

[RACE] RACE: Row-based ASCII Compatible Encoding for IDN, draft-ietf-idn-race.

[RFC2119] Scott Bradner, "Key words for use in RFCs to Indicate Requirement Levels", March 1997, RFC 2119.

[RFC2279] Francois Yergeau, "UTF-8, a transformation format of ISO 10646", January 1998, RFC 2279.

[STD13] Paul Mockapetris, "Domain names - implementation and specification", November 1987, STD 13 (RFC 1035).

A. Authors' Addresses

Paul Hoffman
Internet Mail Consortium and VPN Consortium
127 Segre Place
Santa Cruz, CA 95060 USA
phoffman@imc.org

Patrik Faltstrom
Cisco Systems
170 W Tasman Drive SJ-13/2
San Jose, CA 9 5134 USA
paf@cisco.com