IETF IDN Working Group
Internet Draft
12th May 2000

Editors Zita Wenzel, James Seng

draft-ietf-idn-requirements-02.txt
Expires 12rd Oct 2000

Requirements of Internationalized Domain Names

Status of this Memo

Abstract

This document describes the requirement for encoding international characters into DNS names and records. This document is guidance for developing protocols for internationalized domain names.

1. Introduction

At present, the encoding of Internet domain names is restricted to a subset of 7-bit ASCII (ISO/IEC 646). HTML, XML, IMAP, FTP, and many other text based items on the Internet have already been at least partially internationalized. It is important for domain names to be similarly internationalized or for an equivalent solution to be found. This document assumes that the most effective solution involves putting non-ASCII names inside some parts of the overall DNS system.

This document is being discussed on the "idn" mailing list. To join the list, send a message to <majordomo@ops.ietf.org> with the words "subscribe idn" in the body of the message. Archives of the mailing list can also be found at ftp://ops.ietf.org/pub/lists/idn*.

1.1 Definitions and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Characters mentioned in this document are identified by their position

in the Unicode [UNICODE] character set. The notation U+12AB, for example, indicates the character at position 12AB (hexadecimal) in the Unicode character set. Note that the use of this notation is not an indication of a requirement to use Unicode.

Examples quoted in this document should be considered as a method to further explain the meanings and principles adopted by the document. It is not a requirement for the protocol to satisfy the examples.

A character is a member of a set of elements used for organization, control, or representation of data.

A coded character is a character with its coded representation.

A coded character set ("CCS") is a set of unambiguous rules that establishes a character set and the relationship between the characters of the set and their coded representation.

A graphic character or glyph is a character, other than a control function, that has a visual representation normally handwritten, printed, or displayed.

A character encoding scheme or "CES" is a mapping from one or more coded character sets to a set of octets. Some CESs are associated with a single CCS; for example, UTF-8 [RFC2279] applies only to ISO 10646. Other CESs, such as ISO 2022, are associated with many CCSs.

A charset is a method of mapping a sequence of octets to a sequence of abstract characters. A charset is, in effect, a combination of one or more CCS with a CES. Charset names are registered by the IANA according to procedures documented in RFC 2278.

A language is a way that humans interact. In written form, a language is expressed in characters. The same set of characters can often be used in many languages, and many languages can be expressed using different scripts. A particular charset may have different glyphs (shapes) depending on the language being used.

1.2 Description of the Domain Name System

The Domain Name System is defined by [RFC1034] and [RFC1035], with clarifications, extensions and modifications given in [RFC1123], [RFC1996], [RFC2181] and others. Of special importance here is the security extensions described in [RFC2535] and companions.

Over the years, many different words have been used to describe the components of resource naming on the Internet [URI], [URN], ...; to make certain that the set of terms used in this document are well-defined and non-ambiguous, the definitions are given here.

A master server for a zone holds the main copy of that zone. This copy is sometimes stored in a zone file. A slave server for a zone holds a complete copy of the records for that zone. Slave servers may be either authorized by the zone owner (secondary servers) or unauthorized

(so-called "stealth secondaries"). Master and authorized slave servers are listed in the NS records for the zone, and are termed "authoritative" servers. In many contexts, outside this document the term "primary" is used interchangeably with "master" and "secondary" is used interchangeably with "slave".

A caching server holds temporary copies of DNS records; it uses records to answer queries about domain names. Further explanation of these terms can be found in [RFC1034] and [RFC1996].

DNS names can be represented in multiple forms, with different properties for internationalization. The most important ones are:

- Domain name: The binary representation of a name used internally in the DNS protocol. This consists of a series of components of 1-63 octets, with an overall length limited to 255 octets (including the length fields).
- Master file format domain name: This is a representation of the name as a sequence of characters in some character sets; the common convention (derived from [RFC1035] section 5.1) is to represent the octets of the name as ASCII characters where the octet is in the set corresponding to the ASCII values for [a-zA-Z0-9-], using an escape mechanism (\x or \NNN) where not, and separating the components of the name by the dot character (".").

The form specified for most protocols using the DNS is a limited form of the master file format domain name. This limited form is defined in [RFC1034] Section 3.5 and [RFC1123]. In most implementations of applications today, domain names in the Internet have been limited to the much more restricted forms used, e.g., in email.   Those names are limited to the ASCII upper and lower-case characters (interpreted in a case-independent fashion), the digits, and the hyphen, with the further restrictions that a name may not consist entirely of digits and that a hyphen cannot occur at the beginning or end of a component or following another hyphen.

1.3 Definition of "hostname" and "Internationalized Domain Name"

In the DNS protocols, a name is referred to as a sequence of octets. However, when discussing requirements for internationalized domain names, what we are looking for is ways to represent characters, something meaningful for humans.

In this document, this is referred to as a "hostname". While this term has been used for many different purposes over the years, it is used here in the sense of "sequence of characters (not octets) representing a domain name conforming to the limited hostname syntax".

This document attempts to define the requirements for an "Internationalized Domain Name" (IDN). This is defined as a sequence of characters that can be used in the context of functions where a hostname is used today, but contains one or more characters that are outside the set of characters specified as legal characters for host names.

1.4 A multilayer model of the DNS function

The DNS can be seen as a multilayer function:

- The bottom layer is where the packets passed across the Net in a DNS query and a DNS response. At this level, what matters is the format and meaning of bits and octets in a DNS packet.
- Above that is the "DNS service", created by an infrastructure of DNS servers, NS records that point to those DNS servers, and is pointed to by the root servers (listed in the "root cache file" on each DNS server, often called "named.cache". It is at this level that the statement "the DNS has a single root" [UNIROOT] makes sense, but still, what are being transferred are octets, not characters.
- Interfacing to the user is a service layer, often called "the resolver library", and often embedded in the operating system or system libraries of the client machines. It is at the top of this layer that the API calls commonly known as "gethostbyname" and "gethostbyaddress" reside. These calls are modified to support IPv6 [RFC2553]. A conceptually similar layer exists in authoritative DNS servers, comprising the parts that generate "meaningful" strings in DNS files. Due to the popularity of the "master file" format, this layer often exists only in the administrative routines of the service maintainers.
- The user of this layer (resolver library) is the application programs that use the DNS, such as mailers, mail servers, Web clients, Web servers, Web caches, IRC clients, FTP clients, distributed file systems, distributed databases, and almost all other applications on TCP/IP. (preference not fact)

Graphically, one can illustrate it like this:

```
+---------------+                           +---------------------+
| Application   |                           | (Base data)         |
+---------------+                           +---------------------+
      |  Application service interface                |
      |  For ex. GethostbyXXXX interface              | (no standard)
+---------------+                           +---------------------+
| Resolver      |                           | Auth DNS server     |
+---------------+                           +---------------------+
      |     <-----   DNS service interface   ----->    |
+--------------------------------------------------------------------+
|  DNS service                                                       |
|  +-----------------------+        +--------------------+          |
|  | Forwarding DNS server |        | Caching DNS server |          |
|  +-----------------------+        +--------------------+          |
|                                                                    |
|                 +-------------------------+                        |
|                 | Parent-zone DNS servers |                        |
|                 +-------------------------+                        |
|                                                                    |
|                 +-------------------------+                        |
|                 | Root DNS servers        |                        |
|                 +-------------------------+                        |
|                                                                    |
+--------------------------------------------------------------------+
```


1.5 Service model of the DNS

The Domain Name Service is used for multiple purposes, each of which is characterized by what it puts into the system (the query) and what it expects as a result (the reply).
The most used ones in the current DNS are:

- Hostname-to-address service (A, AAAA, A6): Enter a hostname, and get back an IPv4 or IPv6 address.
  Hostname-to-Mail server service (MX): As above, but the expected return value is a hostname and a priority, for smtp servers.
- Address-to-hostname service (PTR): Enter an IPv4 or IPv6 address (in in-addr.arpa or ip6.int form respectively) and get back a hostname.
- Domain delegation service (NS). Enter a domain name and get back nameserver records (designated hosts who provides authoritive nameservice) for the domain.

New services are being defined, either as entirely new services (IPv6 to hostname mapping using binary labels) or as embellishments to other services (DNSSEC returning information about whether a given DNS service is performed securely or not).

These services exist, conceptually, at the Application/Resolver interface, NOT at the DNS-service interface. This document attempts to set requirements for an equivalent of the "used services" given above, where "hostname" is replaced by "Internationalized Domain Name". This doesn't preclude the fact that IDN should work will any kind of DNS queries. IDN is a new service, since existing protocols like SMTP or HTTP use the old service. it is a matter of great concern how the new and old services work together, and how other protocols can take advantage of the new service.

2. General Requirements

These requirements address two concerns: The service offered to the users (the application service), and the protocol extensions, if needed, added to support this service.

In the requirements, we attempt to use the term "service" whenever a requirement concerns the service, and "protocol" whenever a requirement is believed to constrain the possible implementation.

2.1 Compatibility and Interoperability

[1] The DNS is essential to the entire Internet. Therefore, the service must not damage present DNS protocol interoperability. It must make the minimum number of changes to existing protocols on all layers of the stack. It must continue to allow any system anywhere to resolve any internationalized domain name.

[2] The service must preserve the basic concept and facilities of domain names as described in [RFC1034]. It must maintain a single, global, universal and consistent hierarchical namespace.

[3] The same name resolution request must generate the same response, regardless of the location or localization settings in the resolver, in

the master server, and in any slave servers involved in the resolution process.

[4] If the service allows more than one charset, the protocol should also allow creation of caching servers that do not understand the charset in which a request or response is encoded. Such caching servers should work as well for IDNs as they do for current domain names. The caching server performs correctly if it gives essentially the same answer (without the authoritative bit) as the master server would have if presented with the same request.

[5] A caching server must not return data in response to a query that would not have been returned if the same query had been presented to an authoritative server. This applies fully for the cases when:

- The caching server does not know about IDN
- The caching server implements the whole specification
- The caching server implements a valid subset of the specification

[6] The service should be able to be upgraded at any time with new features and retain backwards compatibility with the current specification.

[7] The service may modify the DNS protocol [RFC1035] and other related work undertaken by the DNSEXT WG. However, these changes should be as small as possible and any changes must be approved by the DNSEXT WG.

[8] The protocol supporting the service should be as simple as possible from the user's perspective. Ideally, users should not realize that IDN was added on to the existing DNS.

[9] A fall-back strategy or mechanism based upon ASCII may be needed during a transition period during deployment and adoption of IDN. Therefore, if an encoding is not mapped into ASCII, then there might be an ASCII-only representation compatible with the current DNS and there should be a way for a program to find the ASCII-only representation for IDN. This is depending on how the protocol will handle exceptions.

[10] The best solution is one that maintains maximum feasible compatibility with current DNS standards as long as it meets the other requirements in this document.

2.2 Internationalization

[11] Internationalized characters must be allowed to be represented and used in DNS names and records. The protocol must specify what charset is used when resolving domain names and how characters are encoded in DNS records.

[12] This document does not recommend any charset for IDN. If more than one charset is used, or might be used in future, in the protocol, then the protocol must specify all the charsets being used and for what purpose. It must also conform to [RFC1766] by tagging the charset. No implicit rules should be allowed for multiple charsets. A CCS(s) chosen

must at least cover the range of characters as currently defined (and as being added) by ISO 10646/Unicode.

[13] CES(s) chosen should not encode ASCII characters differently depending on the other characters in the string. In other words, unless IDN names are identified and coded differently from ASCII-only ones, characters in the ASCII set should remain as specified in [US-ASCII].

[14] The protocol should not invent a new CCS for the purpose of IDN only and should use existing CES. The charset(s) chosen should also be non-ambiguous.

[15] The protocol should not make any assumptions about the location in a domain name where internationalization might appear. In other words, it should not differentiate between any part of a domain name because this may impose restrictions on future internationalization efforts.

[16] The protocol should also not make any localized restrictions in the protocol. For example, an IDN implementation which only allows domain names to use a single local script would immediately restrict multinational organization.

[17] Because of the wide range of devices that use the DNS and the wide range of characteristics of international scripts, the service might need to allow more than one method of domain name input and display. However, there must be a single way of encoding an internationalized domain name within the core of the DNS.

2.3 Localization

[18] The service should be able to handle localized requirements of different languages. For example, IDN must be able to handle bi-directional writing for scripts such as Arabic.

[19] Historically, "." has been the separator of labels in the host names. The service should not use different separators for different languages.

[20] Most of the localization work could be handled by the user interface. It should not matter how the domain names are input or presented, such as in a reverse order or bi-directional, or with the introduction of a new separator. However, the final wire format must be in canonical order.

2.4 Canonicalization

[21] Matching rules are a complicated process for IDN. Canonicalization of characters must follow precise and predictable rules to ensure consistency. [CHARREQ] is a recommended as a guide on canonicalization.

[22] The DNS has to match a host name in a request with a host name held in one or more zones. It also needs to sort names into order. It is expected that some sort of canonicalization algorithm will be used as the first step of this process. This section discusses some of the

properties which will be required of that algorithm.

[23] The canonicalization algorithm might specify operations for case, ligature, and punctuation folding.

[24] In order to retain backwards compatibility with the current DNS, the service must retain the case-insensitive comparison for US-ASCII as specified in [RFC1035]. For example, Latin capital letter A (U+0041) must match Latin small letter a (U+0061). [UTR-21] describes some of the issues with case mapping. Case-insensitivity for non US-ASCII has to be discussed in the protocol proposal.

[25] Case folding must be locale independent. For example, Latin capital letter I (U+0049) case folded to lower case in the Turkish context will become Latin small letter dotless i (U+0131). But in the English context, it will become Latin small letter i (U+0069).

[26] If other canonicalization is done, then it must be done before the domain name is resolved. Further, the canonicalization must be easily upgradable as new languages and writing systems are added.

[27] Any conversion (case, ligature folding, punctuation folding, ...) from what the user enters into a client to what the client asks for resolution must be done identically on any request from any client.

[28] If the protocol specifies a canonicalization algorithm, a caching server should perform correctly regardless of how much (or how little) of that algorithm it has implemented. [1 request to remove]

[29] If the protocol requires a canonicalization algorithm, all requests sent to a caching server must already be in the canonical form.

[30] If the charset can be normalized, then it should be normalized before it is used in IDN. (conflict)

[31] The protocol should avoid inventing a new normalization form provided a technically sufficient one is available (such as in an ISO standard).

2.5 Operational Issues

[32] Zone files should remain easily editable.

[33] An IDN-capable resolver or server shall not generate more traffic than a non-IDN-capable resolver or server would when resolving an ASCII-only domain name. The amount of traffic generated when resolving an IDN shall be similar to that generated when resolving an ASCII-only name.

[34] The service should not add new centralized administration for the DNS. A domain administrator should be able to create internationalized names as easily as adding current domain names.

[35] Within a single zone, the zone manager must be able to define

equivalence rules that suit the purpose of the zone, such as, but not limited to, and not necessarily, non-ASCII case folding, Unicode normalizations (if Unicode is chosen), Cyrillic/Greek/Latin folding, or traditional/simplified Chinese equivalence. Such defined equivalences must not remove equivalences that are assumed by (old or local-rule-ignorant) caches.

[36] The character set of a signed zone file should be the same as the character set of the unsigned zone file. The protocol must allow offline DNSSEC signing. It should be possible to look at the signed file and see that it is the same as the unsigned one.

2.6 Others

[37] The service may provide the same DNS resources using internationalized text as it currently provides using ASCII text.

[38] To get full semantics for IDN, an upgrade of the DNS and related software may be needed.

[39] The protocol should consider new features of DNS such as DNSSEC and DNAME. For example, DNAME might be useful to simplify canonicalization for IDN.

[40] The protocol must work for IPv4 and IPv6.

3. Technical Analysis

There are many standard protocols and RFCs which depend on domain names and have make various assumptions about the characters in them always conforming to [RFC1034] and the other restriction discussed above (see [IABIDN]). We expect that the protocols listed below to be affected:

```
I RFC2813 Internet Relay Chat : Server Protocol
I RFC2805 Media Gateway Control Protocol Architecture and Requirements
S RFC2789 Mail Monitoring MIB
S RFC2782 A DNS RR for specifying the location of services (DNS SRV)
I RFC2775 Internet Transparency
I RFC2772 6Bone Backbone Routing Guidelines
I RFC2768 Network Policy and Services: A Report of a Workshop on
          Middleware
I RFC2767 Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)
S RFC2766 Network Address Translation - Protocol Translation (NAT-PT)
S RFC2765 Stateless IP/ICMP Translation Algorithm (SIIT)
```

I RFC2763 Dynamic Hostname Exchange Mechanism for IS-IS
E RFC2756 Hyper Text Caching Protocol (HTCP/0.0)
S RFC2748 The COPS (Common Open Policy Service) Protocol
S RFC2744 Generic Security Service API Version 2 : C-bindings
S RFC2743 Generic Security Service Application Program Interface
I RFC2705 Media Gateway Control Protocol (MGCP) Version 1.0
I RFC2694 DNS extensions to Network Address Translators (DNS_ALG)
E RFC2693 SPKI Certificate Theory
S RFC2673 Binary Labels in the Domain Name System

S RFC2672 Non-Terminal DNS Name Redirection
S RFC2671 Extension Mechanisms for DNS (EDNS0)
I RFC2663 IP Network Address Translator (NAT) Terminology and Considerations
S RFC2661 Layer Two Tunneling Protocol "L2TP"
E RFC2654 A Tagged Index Object for use in the Common Indexing Protocol
I RFC2637 Point-to-Point Tunneling Protocol (PPTP)
I RFC2636 Wireless Device Configuration (OTASP/OTAPA) via ACAP
S RFC2632 S/MIME Version 3 Certificate Handling
S RFC2622 Routing Policy Specification Language (RPSL)
S RFC1616 Hypertext Transfer Protocol -- HTTP/1.1
I RFC2614 An API for Service Location
S RFC2609 Service Templates and Service: Schemes
B RFC2606 Reserved Top Level DNS Names
I RFC2604 Wireless Device Configuration (OTASP/OTAPA) via ACAP
S RFC2600 Internet Official Protocol Standards
S RFC2595 Using TLS with IMAP, POP3 and ACAP
I RFC2553 Basic Socket Interface Extensions for IPv6
I RFC2546 6Bone Routing Practice
S RFC2543 SIP: Session Initiation Protocol
I RFC2541 DNS Security Operational Considerations
E RFC2540 Detached Domain Name System (DNS) Information
S RFC2539 Storage of Diffie-Hellman Keys in the Domain Name System (DNS)
S RFC2538 Storing Certificates in the Domain Name System (DNS)
S RFC2537 RSA/MD5 KEYs and SIGs in the Domain Name System (DNS)
S RFC2546 DSA KEYs and SIGs in the Domain Name System (DNS)
S RFC2535 Domain Name System Security Extensions
I RFC2517 Building Directories from DNS: Experiences from WWWSeeker
S RFC2511 Internet X.509 Certificate Request Message Format
B RFC2505 Anti-Spam Recommendations for SMTP MTAs
S RFC2500 Internet Official Protocol Standards
S RFC2486 The Network Access Identifier
S RFC2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile
S RFC2421 Voice Profile for Internet Mail - version 2
I RFC2412 The OAKLEY Key Determination Protocol
S RFC2408 Internet Security Association and Key Management Protocol

(ISAKMP)
S RFC2407 The Internet IP Security Domain of Interpretation for ISAKMP
S RFC2401 Security Architecture for the Internet Protocol
S RFC2400 INTERNET OFFICIAL PROTOCOL STANDARDS
S RFC2396 Uniform Resource Identifiers (URI): Generic Syntax
I RFC2377 Naming Plan for Internet Directory-Enabled Applications
I RFC2367 "PF_KEY Key Management API, Version 2"
I RFC2353 APPN/HPR in IP Networks APPN Implementers' Workshop Closed
Pages Document
E RFC2345 Domain Names and Company Name Retrieval
S RFC2326 Real Time Streaming Protocol (RTSP)
B RFC2317 Classless IN-ADDR.ARPA delegation
S RFC2308 Negative Caching of DNS Queries (DNS NCACHE)
S RFC2300 INTERNET OFFICIAL PROTOCOL STANDARDS
S RFC2298 An Extensible Message Format for Message Disposition
Notifications
S RFC2280 Routing Policy Specification Language (RPSL)

```
S RFC2249 Mail Monitoring MIB
S RFC2247 Using Domains in LDAP/X.500 Distinguished Names
I RFC2230 Key Exchange Delegation Record for the DNS
B RFC2219 Use of DNS Aliases for Network Services
S RFC2200 INTERNET OFFICIAL PROTOCOL STANDARDS
I RFC2187 "Application of Internet Cache Protocol (ICP), version 2"
B RFC2182 Selection and Operation of Secondary DNS Servers
S RFC2181 Clarifications to the DNS Specification
E RFC2168 Resolution of Uniform Resource Identifiers using the Domain
          Name System
I RFC2167 Referral Whois (RWhois) Protocol V1.5
S RFC2163 Using the Internet DNS to Distribute MIXER Conformant Global
          Address Mapping (MCGAM)
S RFC2156 MIXER (Mime Internet X.400 Enhanced Relay): Mapping between
          X.400 and RFC 822/MIME
I RFC2151 A Primer On Internet and TCP/IP Tools and Utilities
I RFC2146 U.S. Government Internet Domain Names
S RFC2142 MAILBOX NAMES FOR "COMMON SERVICES, ROLES AND FUNCTIONS"
S RFC2137 Secure Domain Name System Dynamic Update
S RFC2136 Dynamic Updates in the Domain Name System (DNS UPDATE)
I RFC2133 Basic Socket Interface Extensions for IPv6
S RFC2131 Dynamic Host Configuration Protocol
I RFC2130 The Report of the IAB Character Set Workshop
I RFC2101 IPv4 Address Behaviour Today
S RFC2078 "Generic Security Service Application Program Interface,
          Version 2"
S RFC2074 Remote Network Monitoring MIB Protocol Identifiers
I RFC2072 Router Renumbering Guide
S RFC2068 Hypertext Transfer Protocol -- HTTP/1.1
S RFC2065 Domain Name System Security Extensions
E RFC2052 A DNS RR for specifying the location of services (DNS SRV)
S RFC2034 SMTP Service Extension for Returning Enhanced Error Codes
I RFC2010 Operational Criteria for Root Name Servers
E RFC2009 GPS-Based Addressing and Routing
S RFC2000 INTERNET OFFICIAL PROTOCOL STANDARDS
S RFC1996 A Mechanism for Prompt Notification of Zone Changes (DNS
          NOTIFY)
S RFC1995 Incremental Zone Transfer in DNS
S RFC1985 SMTP Service Extension for Remote Message Queue Starting
I RFC1983 Internet Users' Glossary
S RFC1982 Serial Number Arithmetic
S RFC1964 The Kerberos Version 5 GSS-API Mechanism
I RFC1958 Architectural Principles of the Internet
I RFC1955 New Scheme for Internet Routing and Addressing (ENCAPS) for
          IPNG
S RFC1933 Transition Mechanisms for IPv6 Hosts and Routers
S RFC1920 INTERNET OFFICIAL PROTOCOL STANDARDS
I RFC1919 Classical versus Transparent IP Proxies
I RFC1912 Common DNS Operational and Configuration Errors
```

I RFC1900 Renumbering Needs Work
S RFC1891 SMTP Service Extension for Delivery Status Notifications
I RFC1887 An Architecture for IPv6 Unicast Address Allocation
S RFC1886 DNS Extensions to support IP version 6
S RFC1880 INTERNET OFFICIAL PROTOCOL STANDARDS

```
I RFC1877 PPP Internet Protocol Control Protocol Extensions for Name
          Server Addresses
E RFC1876 A Means for Expressing Location Information in the Domain Name
          System
E RFC1845 SMTP Service Extension for Checkpoint/Restart
I RFC1816 U.S. Government Internet Domain Names
S RFC1800 INTERNET OFFICIAL PROTOCOL STANDARDS
I RFC1794 DNS Support for Load Balancing
E RFC1788 ICMP Domain Name Messages
S RFC1780 INTERNET OFFICIAL PROTOCOL STANDARDS
I RFC1739 A Primer On Internet and TCP/IP Tools
S RFC1720 INTERNET OFFICIAL PROTOCOL STANDARDS
I RFC1713 Tools for DNS debugging
E RFC1712 DNS Encoding of Geographical Location
I RFC1711 Classifications in E-mail Routing
I RFC1709 K-12 Internetworking Guidelines
I RFC1707 CATNIP: Common Architecture for the Internet
I RFC1706 DNS NSAP Resource Records
I RFC1705 Six Virtual Inches to the Left: The Problem with IPng
I RFC1703 Principles of Operation for the TPC.INT Subdomain: Radio
          Paging -- Technical Procedures
I RFC1671 IPng White Paper on Transition and Other Considerations
E RFC1664 Using the Internet DNS to Distribute
  RFC1327 Mail Address Mapping Tables
E RFC1637 DNS NSAP Resource Records
I RFC1636 Report of IAB Workshop on Security in the Internet
          Architecture "February 8-10, 1994"
I RFC1630 Universal Resource Identifiers in WWW
I RFC1621 Pip Near-term Architecture
I RFC1616 X.400(1988) for the Academic and Research Community in Europe
S RFC1612 DNS Resolver MIB Extensions
S RFC1611 DNS Server MIB Extensions
S RFC1610 INTERNET OFFICIAL PROTOCOL STANDARDS
E RFC1608 Representing IP Information in the X.500 Directory
S RFC1600 INTERNET OFFICIAL PROTOCOL STANDARDS
I RFC1597 Address Allocation for Private Internets
I RFC1594 FYI on Questions and Answers "Answers to Commonly asked ""New
          Internet User"" Questions"
I RFC1591 Domain Name System Structure and Delegation
I RFC1588 WHITE PAGES MEETING REPORT
I RFC1569 Principles of Operation for the TPC.INT Subdomain: Radio
          Paging -- Technical Procedures
I RFC1546 Host Anycasting Service
S RFC1540 INTERNET OFFICIAL PROTOCOL STANDARDS
I RFC1537 Common DNS Data File Configuration Errors
I RFC1536 Common DNS Implementation Errors and Suggested Fixes
I RFC1535 A Security Problem and Proposed Correction With Widely
          Deployed DNS Software
I RFC1530 Principles of Operation for the TPC.INT Subdomain: General
```

```
          Principles and Policy
E RFC1528 Principles of Operation for the TPC.INT Subdomain: Remote
          Printing -- Technical Procedures
S RFC1519 Classless Inter-Domain Routing (CIDR): an Address Assignment
          and Aggregation Strategy
```

S RFC1500 INTERNET OFFICIAL PROTOCOL STANDARDS
- RFC1486 An Experiment in Remote Printing
- RFC1480 The US Domain
- RFC1470 FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices
- RFC1464 Using the Domain Name System To Store Arbitrary String Attributes
- RFC1459 Internet Relay Chat Protocol
- RFC1454 Comparison of Proposals for Next Version of IP
- RFC1430 A Strategic Plan for Deploying an Internet X.500 Directory Service
- RFC1415 FTP-FTAM Gateway Specification
- RFC1410 IAB OFFICIAL PROTOCOL STANDARDS
- RFC1401 Correspondence between the IAB and DISA on the use of DNS throughout the Internet
- RFC1395 BOOTP Vendor Information Extensions
- RFC1392 Internet Users' Glossary
- RFC1386 The US Domain
- RFC1385 EIP: The Extended Internet Protocol A Framework for Maintaining Backward Compatibility
- RFC1383 An Experiment in DNS Based IP Routing
- RFC1360 IAB OFFICIAL PROTOCOL STANDARDS
- RFC1348 DNS NSAP RRs
- RFC1347 "TCP and UDP with Bigger Addresses (TUBA)," A Simple Proposal for Internet Addressing and Routing
- RFC1335 A Two-Tier Address Structure for the Internet: A Solution to the Problem of Address Space Exhaustion
- RFC1325 FYI on Questions and Answers "Answers to Commonly asked ""New Internet User"" Questions"
- RFC1309 Technical Overview of Directory Services Using the X.500 Protocol
- RFC1308 Executive Introduction to Directory Services Using the X.500 Protocol
- RFC1291 Mid-Level Networks Potential Technical Services
- RFC1280 IAB OFFICIAL PROTOCOL STANDARDS
- RFC1279 X.500 and Domains
- RFC1274 The COSINE and Internet X.500 Schema
- RFC1250 IAB OFFICIAL PROTOCOL STANDARDS
- RFC1207 FYI on Questions and Answers "Answers to Commonly asked ""Experienced Internet User"" Questions"
- RFC1206 FYI on Questions and Answers "Answers to Commonly asked ""New Internet User"" Questions"
- RFC1200 IAB OFFICIAL PROTOCOL STANDARDS
- RFC1183 New DNS RR Definitions
- RFC1177 FYI on Questions and Answers "Answers to Commonly asked ""New Internet User"" Questions"
- RFC1175 FYI on Where to Start - A Bibliography of Internetworking Information
- RFC1174 IAB Recommended Policy on Distributing Internet Identifier

Assignment And "IAB Recommended Policy Change to Internet ""Connected"" Status"

- RFC1168 INTERMAIL AND COMMERCIAL MAIL RELAY SERVICES
- RFC1147 FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices

- RFC1123 Requirements for Internet Hosts -- Application and Support
- RFC1101 DNS Encoding of Network Names and Other Types
- RFC1100 IAB OFFICIAL PROTOCOL STANDARDS
- RFC1085 ISO Presentation Services on top of TCP/IP-based internets
- RFC1083 IAB OFFICIAL PROTOCOL STANDARDS
- RFC1035 DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION
- RFC1034 DOMAIN NAMES - CONCEPTS AND FACILITIES
- RFC0830 A Distributed System for Internet Name Service

S - Standards Track I - Informational
E - Experimental B - Best Current Practice

All idn protocol proposal documents must fully detail the expected effects of leaking of the specified encoding to protocols other than the DNS resolution protocol.

4. Security Considerations

Any solution that meets the requirements in this document must not be less secure than the current DNS. Specifically, the mapping of internationalized host names to and from IP addresses must have the same characteristics as the mapping of today's host names.

Specifying requirements for internationalized domain names does not itself raise any new security issues. However, any change to the DNS may affect the security of any protocol that relies on the DNS or on DNS names. A thorough evaluation of those protocols for security concerns will be needed when they are developed. In particular, IDNs must be compatible with DNSSEC and, if multiple charsets or representation forms are permitted, the implications of this name-spoof must be throughly understood.

5. References

[CHARREQ] "Requirements for string identity matching and String Indexing", http://www.w3.org/TR/WD-charreq, July 1998, World Wide Web Consortium.

[DNSEXT] "IETF DNS Extensions Working Group", namedroppers@internic.net, Olafur Gudmundson, Randy Bush.

[RFC1034] "Domain Names - Concepts and Facilities", rfc1034.txt, November 1987, P. Mockapetris.

[RFC1035] "Domain Names - Implementation and Specification", rfc1035.txt, November 1987, P. Mockapetris.

[RFC1123] "Requirements for Internet Hosts -- Application and Support", rfc1123.txt, October 1989, R. Braden.

[RFC1766] Tags for the Identification of Languages, rfc1766.txt, March 1995, H. Alvestrand.

[RFC1996] "A Mechanism for Prompt Notification of Zone Changes

(DNS NOTIFY)", rfc1996.txt, August 1996, P. Vixie.

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", rfc2119.txt, March 1997, S. Bradner.

[RFC2181] "Clarifications to the DNS Specification", rfc2181.txt, July 1997, R. Elz, R. Bush.

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

[RFC2535] "Domain Name System Security Extensions", rfc2535.txt, March 1999, D. Eastlake.

[RFC2553] "Basic Socket Interface Extensions for IPv6", rfc2553.txt, March 1999, R. Gilligan and al.

[UNIROOT] "IAB Technical Comment on the Unique DNS Root", draft-iab-unique-dns-root-00.txt, iab@iab.org

[IABIDN] "A Tangled Web:issues of I18N domain names, and the other Internet protocols", rfc2825.txt iab@iab.org

[UNICODE] The Unicode Consortium, "The Unicode Standard -- Version 3.0", ISBN 0-201-61633-5. Described at http://www.unicode.org/unicode/standard/versions/Unicode3.0.html

[US-ASCII] Coded Character Set -- 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986.

[UTR15] "Unicode Normalization Forms", Unicode Technical Report #15, http://www.unicode.org/unicode/reports/tr15/, Nov 1999, M. Davis & M. Duerst, Unicode Consortium.

[UTR21] "Case Mappings", Unicode Technical Report #21, http://www.unicode.org/unicode/reports/tr21/, Dec 1999, M. Davis, Unicode Consortium.

6. Editors' Contact

Zita Wenzel
[...]

James Seng
8 Temesek Boulevand
#24-02 Suntec Tower 3
Singapore 038988
Tel: +65 248-6208

Fax: +65 248-6198
Email: jseng@pobox.org.sg

7. Acknowledgements

The editor gratefully acknowledges the contributions of:

Harald Tveit Alvestrand <Harald@Alvestrand.no>
Martin Duerst <duerst@w3.org>
Patrik Faltstrom <paf@swip.net>
Andrew Draper <ADRAPER@altera.com>
Bill Manning <bmanning@ISI.EDU>
Paul Hoffman <phoffman@imc.org>
James Seng <jseng@pobox.org.sg>
Randy Bush <randy@psg.com>
Alan Barret <apb@cequrux.com>
Olafur Gudmundsson <ogud@tislabs.com>
Karlsson Kent <keka@im.se>
Dan Oscarsson <Dan.Oscarsson@trab.se>
J. William Semich <bill@mail.nic.nu>
RJ Atkinson <request not to have email>
Simon Josefsson <jas+idn@pdc.kth.se>
Ned Freed <ned.freed@innosoft.com>
Dongman Lee <dlee@icu.ac.kr>
Mark Andrews <Mark.Andrews@nominum.com>
John Klensin <klensin+idn@jck.com>
Tan Juay Kwang <tanjk@i-dns.net>