Abstract

The EName workshop asks about the explicitness of the resolution context. There have been a number of innovations in the DNS that provide contextual information. This memo looks at some of them in order to spur thinking about what "explicit context" might be needed.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 12, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 2
2. The 'Missing' Control Plane ................................. 3
3. Network and Technology Context ............................ 3
   3.1. Protocol Switching by Using Label ...................... 3
1. Introduction

It may be that naming things is hard, but there is reason to suppose that the way we name things on the Internet is harder than it ought to be. While it might turn out that the facts of the world (i.e. the "installed base") make changing things impracticable, it is nevertheless worth wondering what current features might be better, if one had the opportunity to change things.

Domain names, and the Domain Name System (DNS, [RFC1034] [RFC1035]; see [I-D.lewis-domain-names] for additional discussion of this distinction) are inarguably important if not central to naming things on the Internet. Yet both domain names and the DNS seem to be fraying in the face of various needs of Internet users. If we are to understand whether that fraying is real, whether it can be contained, and whether it can be bound back together, we first must comprehend the source of the friction that causes the cords to fray.

One way of understanding the friction is that it comes from failing to know the desire of the user at resolution time. If we set aside Quine's observation that such explicit knowledge may not be determined even for the person holding the desire, then we might be tempted to think that adding some sort of resolution context will make this situation enough better that such an addition would help. In order to explore that possibility, it is useful to think of some of the contextual information that has either been added or inferred in domain name use in recent years, in order to be clear about the manifold contexts that may be relevant.

Even though it is possible to distinguish between "domain names" and "the Domain Name System", domain names inarguably make up the basic naming system for the Internet, if for no other reason than that many applications and protocols have rules about what may fit into a domain name slot. (Indeed, this basic fact is what makes any adjustment of naming on the Internet a thorny problem.)

This memo assumes familiarity with the DNS, and with contemporary discussions about deficiencies and possible improvements. An exhaustive list would be tedious here, but a reader unfamiliar with the references of [RFC7719] will probably find the rest of this memo totally opaque. Specific references are offered when they seem appropriate. The reader is also assumed to be familiar with controversies about naming that have appeared around the IETF in recent years, particularly but not only within the DNS Operations (DNSOP) Working Group. The reader may also find that familiarity
with the discussion in [I-D.lewis-domain-names],
[I-D.klensin-dns-function-considerations], and [I-D.stw-whatsinaname]
will be useful. Some of the what follows owes a debt to
[I-D.hardie-arc-pointers].

2. The 'Missing' Control Plane

When dealing with domain names, one must always attend to features of
the DNS, even if the DNS is not in use for that domain name. One of
the constraining facts about the DNS is that it does not distinguish
between control and data plane messages. It is a completely
symmetric protocol among both nodes in any exchange, with the
different roles signalled by one (or, with DNSSEC, as many as three)
bits.

The approach contains advantages and disadvantages. Without a
control plane, there is no need to negotiate versions for the
protocol. The protocol can be (and has been) extended simply by
adding new resource record types (notably, with EDNS and the OPT
RRTYPE: see [RFC2671] and [RFC6891]). In a hop-by-hop protocol
designed for extremely low latency, keeping handshaking and version
negotiation to a minimum seems well-advised. On the other hand,
every innovation in the DNS needs to consider the possible effects on
resolvers that are more than 20 years old, "just in case". Moreover,
the lack of signalling means that there is no way to ascertain the
context of resolution, and no way to fix certain "misfeatures" of the
original DNS specification.

3. Network and Technology Context

3.1. Protocol Switching by Using Label

The DNS emerged in an environment of heterogenous name-resolution
systems, and it was natural for users to enter a familiar name and
have it go to "the right place". The original DNS design appears to
have encouraged this assumption by users, in part by offering the

CLASS division of the name space. The division might have been
useful to distinguish among different network technologies (for some
tangential discussion, see [I-D.sullivan-dns-class-useless]). The
IAB made note of an underlying assumption of name system
heterogeneity in [RFC6055], and there is no reason to suppose that
heterogeneity has diminished since. Nevertheless, to the extent
CLASS was ever in wide use it appears to have declined to the point
of irrelevance. This means that many applications implicitly assume
the DNS IN CLASS when performing lookups (even if DNS is not actually
in use), and often rely on the underlying resolution infrastructure
to use a different resolution context if that is appropriate.

While many naming systems (including the DNS) do not restrict their
respective network scopes, some other naming systems depend on scope
in order to resolve correctly. Examples include mDNS ([RFC6762]) and
the Tor network (for background, see [RFC6786]). Because of a desire
to use existing applications with these technologies, and the
effective death of CLASS, it is natural to use "in-band signalling"
-- effectively, a special domain name -- to tell applications and resolver libraries that there is something special to do. Therefore, some naming systems use a special domain name in order to signal that the correct resolution protocol has changed. Usually, the special domain name is at the top level, but that is not necessary. The key feature is that the domain name itself functions as an in-band protocol switch, to change the resolution context. Since there is no control plane to provide such switching, this technique is not surprising.

3.2. Network Names Related to a Network

People sometimes want certain resources to be available to nodes on a network without those resources being available to everyone connected to the Internet. The result is a desire to give different answers for the same query, depending on where that query came from. Sometimes, this is implemented with the unstandardized but widely-implemented "views" feature of nameservers such as BIND. In many cases, the different answers are generated simply based on the network address that originated the query. For authoritative servers, however, that technique may be inadequate because the (differential) answer is supposed to be tailored to the end point that originated the query (and an authoritative server is likely to have the IP address of an intermediate resolver, not the end point). To meet the latter need, the IETF created "Client Subnet in DNS Queries" [RFC7871].

3.3. DWIM and the Search Path

The DNS specification included a mechanism by which partial domains could be completed by the addition of a "local domain" [RFC1034] or some other domain known to the software in question. In master files, there is a directive to contain the relative domain names and thereby make the context explicit, but in a resolution context the search path is usually automatically configured and often not easily accessible to the user of the system. The relative context usually includes the root label, so most domain names that users enter are relative ones (if only relative to the root).

Relative names and search paths would probably do little harm if it were always clear which network context a given host or resolution is happening in. Unfortunately, as more systems are multi-homed, and with the widespread adoption of things like VPNs, the network context is often far from clear. There are DHCP options (see [RFC6731]), but that mechanism works by specifying target domains related to a particular interface and may be less flexible than required not to surprise users.

4. Data Subtyping

The DNS is, in its design, "strongly typed". There are several well-
know RRTYPEs, most of which date to the early specification of DNS. It is perhaps the greatest failing of the DNS that the RRTYPEs defined in [RFC1035] and perhaps a few others are the only ones that have been widely adopted: what was the point of an extensible type system, if many would reject the unknown ones? Regardless, applications all over the Internet assume a fixed set of RRTYPEs: even though it is not hard to perform DNS queries using an unknown RRTYPE (see [RFC3597]), many applications restrict the ability to use them. (The in-band signal RRTYPEs appear to be the exception to this observation, which is not too surprising: a DNS system that does not know about a change to the control plane will simply never use that RRTYPE in the first place, and may behave in surprising ways when queried for it.)

The "underscore convention" provides a way to address unreliable acceptance of new types, and also perhaps to provide metadata about a given name and RRTYPE. It emerged originally as part of the SRV specification ([RFC2782]), "to lower the probability of an accidental clash with a similar name used for unrelated purposes." But the underscore convention was then used to subtype lookups at a parent domain, using a TXT record. In effect, the underscore convention became a DNS RRTYPE subtyping convention.

5. Label Internationalization

In theory, DNS labels may be made up of any octet. Because of the reasons outlined in "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale" ([RFC5894]), there were unfortunate consequences of simply assuming that labels would always be interpretable as sequences of UTF-8 encoded Unicode code points. Therefore, IDNA offers an ASCII-compatible encoding for internationalized domain names.

Even if IDNA were adequate to cover the internationalization needs of domain names (and it isn't -- see below), it would not cover all the problems, because not every application uses IDNA. As the IAB points out in [RFC6055], some non-DNS resolution mechanisms that nevertheless use domain names have specified what kind of input they would get. So, for instance, mDNS [RFC6762] specifies that labels are interpreted as UTF-8-encoded strings. In addition, some uses of the DNS itself can depend on UTF-8-encoded strings. For example, DNS-SD [RFC6763] specifies UTF-8 strings for its labels, and permits a character repertoire in some labels (most notably, the <Instance> portion of the domain name) beyond what is permitted by IDNA2008 [RFC5890], [RFC5891], [RFC5892], [RFC5893], and [RFC5894].

At the same time, it is manifest that IDNA is inadequate to solve all internationalization issues anyway. The original version (IDNA2003,
limited itself to a particular version of Unicode and turned out to use the wrong Unicode normalization form. IDNA2008, by contrast, was supposed to be Unicode version-neutral and to avoid normalization problems. IDNA2008 always recognized that it could not be used without informed policies imposed by zone operators, but it turned out that even with such policies in place it might not be as safe as intended. See [I-D.sullivan-lucid-prob-stmt] and [I-D.freytag-troublesome-characters] for some discussion. Those documents proceed from the assumption that DNS lookup contexts are inherently language- or writing-system-context-free. It may be, however, that finding some way to signal some kind of linguistic context would be a better answer than trying to make internationalization work in the DNS without a language context.

In any case, internationalization of domain names automatically creates at least one context, because the rules for the presentation form of the domain name (i.e. the U-label form in IDNA) are different. Because the DNS treats upper and lower case ASCII characters as matching (even while it preserves the case), traditional LDH labels have case-insensitive matching while U-labels do not (because upper case characters are not permitted in U-labels). Unfortunately, however, if the application does some sort of mapping (such as described in [RFC5895]), that context is lost to the name lookup system. Similarly, for the purposes of users, it is not plain why some labels permit things like spaces and other labels do not (see, e.g. [I-D.ietf-dnssd-mdns-dns-interop]).

It is also worth noting that IDNA used yet another in-band signal mechanism to differentiate processing of its labels: a two-ASCII-letter prefix followed by two hyphen-minus (-) characters.

6. Practical Problems for Applications

Just because some kind of communication of context is possible in the naming system, that does not mean that applications actually have access to such context. The historic interface for naming on many systems did not provide the facilities to communicate context even if an application was prepared to send or receive such information. This limitation is gradually being reduced by projects such as <https://getnsapi.net>, but such libraries are not universally available yet (and anyway, do not provide an interface to all naming systems).

7. What Is To Be Done?

The foregoing suggests a bleak outlook even for understanding what "context" is at the time of resolution, never mind what to do with it. But there are some lessons. The IAB observed, in [RFC5218], that incremental deployability is a major consideration in whether a protocol is successful. It seems a natural inference that incremental deployability is also likely to be important for extensions of a protocol's capability (and indeed, extensibility seems to be important for wild success). This suggests that the strategy of using in-band signals may be a way forward. It is not
plain whether the signals are best carried in underscore labels or by subtyping lookups using the "[character][character]--" pattern IDNA adopted. But without an incremental deployment strategy, a new naming system is likely doomed; and with such a strategy, the reasons to abandon DNS itself seem weaker, since fallback to DNS will always be necessary.

8. Informative References

[I-D.freytag-troublesome-characters]

[I-D.hardie-arc-pointers]

[I-D.ietf-dnssd-mdns-dns-interop]

[I-D.klensin-dns-function-considerations]

[I-D.lewis-domain-names]

[I-D.stw-whatsinaname]

[I-D.sullivan-dns-class-useless]

[I-D.sullivan-lucid-prob-stmt]
Sullivan, A. and A. Freytag, "A Problem Statement to Motivate Work on Locale-free Unicode Identifiers", draft-sullivan-lucid-prob-stmt-00 (work in progress), March


