Internet Engineering Task Force (IETF)
Request for Comments: 8222
Category: Informational
ISSN: 2070-1721

A. Sullivan
Oracle

September 2017

Selecting Labels for Use with Conventional DNS and Other Resolution Systems in DNS-Based Service Discovery

Abstract

Despite its name, DNS-Based Service Discovery (DNS-SD) can use naming systems other than DNS when looking for services. Moreover, when it uses DNS, DNS-SD uses the full capability of DNS, rather than using a subset of available octets. This is of particular relevance where some environments use DNS labels that conform to Internationalized Domain Names for Applications (IDNA), and other environments use labels containing Unicode characters (such as containing octets corresponding to characters encoded as UTF-8). In order for DNS-SD to be used effectively in environments where multiple different name systems and conventions for their operation are in use, it is important to attend to differences in the underlying technology and operational environment. This memo presents an outline of the requirements for the selection of labels for conventional DNS and other resolution systems when they are expected to interoperate in this manner.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

DNS-Based Service Discovery (DNS-SD, [RFC6763]) specifies a mechanism for discovering services using queries to DNS ([RFC1034] and [RFC1035]) and to any other system that uses domain names, such as Multicast DNS (mDNS, [RFC6762]). Many applications that use DNS follow "Internet hostname" syntax [RFC952] for labels -- the so-called LDH (letters, digits, and hyphen) rule. That convention is the reason behind the development of Internationalized Domain Names for Applications (IDNA2008, [RFC5890], [RFC5891], [RFC5892], [RFC5893], [RFC5894], and [RFC5895]). It is worth noting that the LDH rule is a convention, and not a rule of the DNS; this is made entirely plain by Section 11 of [RFC2181], and discussed further in Section 3 of [RFC6055]. Nevertheless, there is a widespread belief that in many circumstances domain names cannot be used in the DNS unless they follow the LDH rule.

At the same time, mDNS requires that labels be encoded in UTF-8 and permits a range of characters in labels that are not permitted by IDNA2008 or the LDH rule. For example, mDNS encourages the use of spaces and punctuation in mDNS names (see Section 4.2.3 of [RFC6763]). It does not restrict which Unicode code points may be used in those labels, so long as the code points are UTF-8 in Net-Unicode [RFC5198] format.

Users and developers of applications are, of course, frequently unconcerned with (or oblivious to) the name-resolution system(s) in service at any given moment; they are inclined simply to use the same domain names in different contexts. As a result, names entered into the same domain name slot might be resolved using different name resolution technologies. If a given name will not work across the various environments, then user expectations are likely to be best satisfied when at least some parts of the domain names to be queried are compatible with the rules and conventions for all the relevant technologies. Given the uses of DNS-SD, a choice for such compatibility likely lies with the application designer or service operator.

One approach to interoperability under these circumstances is to use a single operational convention (a "profile") for domain names under the different naming systems. This memo assumes such a use profile, and attempts to outline what is necessary to make it work without specifying any particular technology. It does assume, however, that the global DNS is likely to be implicated. Given the general tendency of all resolution eventually to fall through to the DNS, that assumption does not seem controversial.

It is worth noting that users of DNS-SD do not use the service discovery names in the same way that users of other domain names might. In many cases, domain names can be entered as direct user input. But the service discovery context generally assumes that users are picking a service from a list. As a result, the sorts of application considerations that are appropriate to the general-purpose DNS name, and that resulted in the A-label/U-label split (see below) in IDNA2008, are not entirely the right approach for DNS-SD.

1.1. Conventions and Terms Used in This Document

Wherever appropriate, this memo uses the terminology defined in Section 2 of [RFC5890]. In particular, the reader is assumed to be familiar with the terms "U-label", "LDH label", and "A-label" from that document. Similarly, the reader is assumed to be familiar with the U+NNNN notation for Unicode code points used in [RFC5890] and other documents dealing with Unicode code points. In the interests of brevity and consistency, the definitions are not repeated here.

Sometimes this memo refers to names in the DNS as though the LDH rule and IDNA2008 are strict requirements. They are not. DNS labels are, in principle, just collections of octets; therefore, in principle, the LDH rule is not a constraint. In practice, applications sometimes intercept labels that do not conform to the LDH rule and apply IDNA and other transformations.

DNS, perhaps unfortunately, has produced its own jargon. Unfamiliar DNS-related terms in this memo should be found in [RFC7719].

The term "owner name" (common to the DNS vernacular; see above) is used here to apply not just to the domain names to be looked up in the DNS, but to any name that might be looked up either in the DNS or using another technology. Therefore, it includes names that might not actually exist anywhere. In addition, what follows depends on the idea that not every domain name will be looked up in the DNS. For instance, names ending in "local." (in the presentation format) are not ordinarily looked up using DNS, but instead looked up using mDNS.

DNS-SD specifies three portions of the owner name for a DNS-SD resource record. These are the <Instance> portion, the <Service> portion, and the <Domain> portion. The owner name made of these three parts is called the "Service Instance Name". It is worth observing that a portion may be more than one label long. See Section 4.1 of [RFC6763]. Further discussion of the parts is found in Section 4.

Throughout this memo, mDNS is used liberally as the alternative resolution mechanism to DNS. This is for convenience rather than rigor: any alternative name resolution to DNS could present the same friction with the prevailing operational conventions of the global DNS. It so happens that mDNS is the overwhelmingly successful alternative as of this writing, so it is used in order to make the issues plainer to the reader. Other alternative resolution mechanisms may generally be read wherever mDNS appears in the text, except where details of the mDNS specification appear.

2. Why There Could Be a Problem at All

One might reasonably wonder why there is a problem to be solved at all. After all, DNS labels permit any octet whatsoever, and anything that can be useful with DNS-SD cannot use any names that are outside the protocol strictures of the DNS.

The reason for the trouble is twofold. First, and least troublesome, is the possibility of resolvers that are attempting to offer IDNA service system-wide. Given the design of IDNA2008, it is reasonable to suppose that, on some systems, high-level name resolution libraries will perform the U-label/A-label transformation automatically, saving applications from these details. But system-level services do not always have available to them the resolution context, and they may apply the transformation in a way that foils rather than helps the application. Of course, if this were the main problem, it would presumably be self-correcting because the right answer would be, "Don’t use those libraries for DNS-SD", and DNS-SD would not work reliably in cases where such libraries were in use. This would be unfortunate, but given that DNS-SD in Internet contexts is (as of this writing) not in ubiquitous use, it should not represent a fatal issue.

The greater problem is that the "infrastructure" types of DNS service -- the root zone, the top-level domains, and so on -- have embraced IDNA and refuse registration of raw UTF-8 into their zones. As of this writing, there is (perhaps unfortunately) no reliable way to discover where these sorts of DNS services end. Nevertheless, some client programs (notably web browsers) have adopted a number of different policies about how domain names will be looked up and presented to users given the policies of the relevant DNS zone operators. None of these policies permit raw UTF-8. Since it is anticipated that DNS-SD when used with the DNS will be inside domain names beneath those kinds of "infrastructure" domains, the implications of IDNA2008 must be a consideration.

For further exploration of issues relating to encoding of domain names generally, the reader should consult [RFC6055].

3. Requirements for a Profile for Label Interoperation

Any interoperability between DNS (including prevailing operational conventions) and other resolution technologies will require interoperability across the portions of a DNS-SD Service Instance Name that are implicated in regular DNS lookups. Only some portions are implicated. In any case, if a given portion is implicated, the profile will need to apply to all labels in that portion.

In addition, because DNS-SD Service Instance Names can be used in a domain name slot, care must be taken by DNS-SD-aware resolvers to handle the different portions as outlined here, so that DNS-SD portions that do not use IDNA2008 will not be treated as U-labels and will not accidentally undergo IDNA processing.

Because the profile will apply to names that might appear in the public DNS, and because other resolution mechanisms (such as mDNS) could permit labels that IDNA does not, the profile might reduce the labels that could be used with those other resolution mechanisms.

One consequence of this is that some recommendations from [RFC6763] will not really be possible to implement using names subject to the profile. In particular, Section 4.2.3 of [RFC6763] recommends that labels always be stored and communicated as UTF-8, even in the DNS. Because of the way that the public DNS is currently operated (see Section 2), the advice to store and transmit labels as UTF-8 in the DNS is likely either to encounter problems, to result in unnecessary traffic to the public DNS, or to do both. In particular, many labels in the <Domain> part of a Service Instance Name are unlikely to be found in the UTF-8 form in the public DNS tree for zones that are using IDNA2008. By contrast, for example, mDNS exclusively uses UTF-8.

U-labels cannot contain uppercase letters (see Sections 3.1.3 and 4.2 of [RFC5894]). That restriction extends to ASCII-range uppercase letters that work fine in LDH labels. It may be confusing that the character "A" works in the DNS when none of the characters in the label has a diacritic, but it does not work when there is such a diacritic in the label. Labels in mDNS names (or other resolution technologies) may contain uppercase characters, so the profile will need either to restrict the use of uppercase or to come up with a convention for case folding (even in the presence of diacritics) that is reliable and predictable to users.

4. DNS-SD Portions

Service Instance Names are made up of three portions.

4.1. The <Instance> Portion of the Service Instance Name

[RFC6763] is clear that the <Instance> portion of the Service Instance Name is intended for presentation to users; therefore, virtually any character is permitted in it. There are two ways that a profile might address this portion.

The first way would be to treat this portion as likely to be intercepted by system-wide IDNA-aware (but otherwise context-unaware) resolvers or likely subject to strict IDNA-conformance requirements for publication in the relevant zone. In this case, the portion would need to be made subject to the profile, thereby curtailing what characters may appear in this portion. This approach permits DNS-SD to use any standard system resolver but presents inconsistencies with the DNS-SD specification and with DNS-SD use that is exclusively mDNS-based. Therefore, this strategy is rejected.

Instead, DNS-SD implementations can intercept the <Instance> portion of a Service Instance Name and ensure that those labels are never handed to IDNA-aware resolvers that might attempt to convert these labels into A-labels. Under this approach, the DNS-SD <Instance> portion works as it always does, but at the cost of using special resolution code built into the DNS-SD system. A practical consequence of this is that zone operators need to be prepared not to apply the LDH rule to all labels, and they may need to make special concessions to ensure that the <Instance> portion can contain spaces, uppercase and lowercase, and any UTF-8 code point. Otherwise, they need to prepare a user interface to handle the exceptions that would be generated. Automatic conversion to A-labels is not acceptable.

It is worth noting that this advice is not actually compatible with the advice in Section 4 of [RFC6055]. That section appears to assume that names are not really composed of subsections, but because [RFC6763] specifies portions of names, the advice in this memo is to follow the advice of [RFC6055] according to the portion of the domain name, rather than for the whole domain name. As a practical matter, this means special-purpose name resolution software for DNS-SD.

4.2. The <Service> Portion of the Service Instance Name

DNS-SD includes a <Service> component in the Service Instance Name. This component is not really user-facing data; instead it is control data embedded in the Service Instance Name. This component includes so-called "underscore labels", which are labels prepended with U+005F (_). The underscore label convention was established by DNS SRV ([RFC2782]) for identifying metadata inside DNS names. A system-wide resolver (or DNS middlebox) that cannot handle underscore labels will not work with DNS-SD at all, so it is safe to suppose that such resolvers will not attempt to do special processing on these labels. Therefore, the <Service> portion of the Service Instance Name will not be subject to the profile. By the same token, underscore labels are never subject to IDNA processing (they are formally incompatible); therefore, concerns about IDNA are irrelevant for these labels.

4.3. The <Domain> Portion of the Service Instance Name

The <Domain> portion of the Service Instance Name forms an integral part of the owner name submitted for DNS resolution. A system-wide resolver that is IDNA2008-aware is likely to interpret labels with UTF-8 in the owner name as candidates for IDNA2008 processing. More important, operators of internationalized domain names will frequently publish such names in the public DNS as A-labels; certainly, the topmost labels will always be A-labels. Therefore, these labels will need to be subject to the profile. DNS-SD implementations ought to identify the <Domain> portion of the Service Instance Name and treat it subject to IDNA2008 in case the domain is to be queried from the global DNS. (This document does not specify how to do that and does not alter the specification in [RFC6763].) In the event that the <Domain> portion of the Service Instance Name fails to resolve, it is acceptable to substitute labels with plain UTF-8, starting at the lowest label in the DNS tree and working toward the root. This approach would differ from the rule for resolution published in [RFC6763], because this approach privileges IDNA2008-compatible labels over UTF-8 labels. There is more than one way to achieve such a result, but in terms of predictability, it is probably best if the lowest-level resolution component is able to learn the correct resolution context so that it can perform the correct transformations on the various domain portions.

One might argue against the above restriction on either of two grounds:

1. It is possible that the names may be in the DNS in UTF-8, and RFC 6763 already specifies a fallback strategy of progressively attempting first the UTF-8 label lookup (it might not be a U-label) and then, if possible, the A-label lookup.

2. Zone administrators that wish to support DNS-SD can publish a UTF-8 version of the zone along side the A-label version of the zone.

The first of these is rejected because it represents a potentially significant increase in DNS lookup traffic. It is possible for a DNS-SD application to identify the <Domain> portion of the Service Instance Name. The standard way to publish IDNs on the Internet uses IDNA. Therefore, additional lookups should not be encouraged. When [RFC6763] was published, the bulk of IDNs were lower in the tree. Now that there are internationalized labels in the root zone, it is desirable to minimize queries to the Internet infrastructure if they are sure to be answered in the negative.

The second reason depends on the idea that it is possible to maintain two names in sync with one another. This is not strictly speaking true, although in this case the domain operator could simply create a DNAME record [RFC6672] from the UTF-8 name to the IDNA2008 zone. This still, however, relies on being able to reach the (UTF-8) name in question, and it is unlikely that the UTF-8 version of the zone will be delegated from anywhere. Moreover, in many organizations, the support for DNS-SD and the support for domain name delegations are not performed by the same department; depending on a coordination between the two will make the system more fragile, slower, or both.

Some resolvers -- particularly those that are used in mixed DNS and non-DNS environments -- may be aware of different operational conventions in different parts of the DNS tree. For example, it may be possible for implementations to use hints about the boundary of an organization’s domain name infrastructure in order to tell, for instance, that example.com. is part of the Example Organization, while com. is a large delegation-centric zone on the public Internet. In such cases, the resolution system might reverse its preferences to prefer plain UTF-8 labels when resolving names below the boundary point in the DNS tree. The result would be that any lookup past the boundary point and closer to the root would use LDH labels first, falling back to UTF-8 only after a failure; but a lookup below the boundary point would use UTF-8 labels first, and try other strategies only in case of negative answers. The mechanism to learn such a boundary is beyond the scope of this document.

5. IANA Considerations

This document does not require any IANA actions.

6. Security Considerations

This memo presents some requirements for future development, but does not specify anything. It makes no additional security-specific requirements. Issues arising due to visual confusability of names apply to this case as well as to any other case of internationalized names, but interoperation between different resolution systems and conventions does not alter the severity of those issues.

7. Informative References

[RFC952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, DOI 10.17487/RFC0952, October 1985, <https://www.rfc-editor.org/info/rfc952>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <https://www.rfc-editor.org/info/rfc5198>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, DOI 10.17487/RFC5891, August 2010, <https://www.rfc-editor.org/info/rfc5891>.

[RFC5892] Faltstrom, P., Ed., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, DOI 10.17487/RFC5892, August 2010, <https://www.rfc-editor.org/info/rfc5892>.

[RFC5893] Alvestrand, H., Ed. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, DOI 10.17487/RFC5893, August 2010, <https://www.rfc-editor.org/info/rfc5893>.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, DOI 10.17487/RFC5894, August 2010, <https://www.rfc-editor.org/info/rfc5894>.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, DOI 10.17487/RFC5895, September 2010, <https://www.rfc-editor.org/info/rfc5895>.

[RFC6055] Thaler, D., Klensin, J., and S. Cheshire, "IAB Thoughts on Encodings for Internationalized Domain Names", RFC 6055, DOI 10.17487/RFC6055, February 2011, <https://www.rfc-editor.org/info/rfc6055>.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012, <https://www.rfc-editor.org/info/rfc6672>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <https://www.rfc-editor.org/info/rfc7719>.

Acknowledgments

The author gratefully acknowledges the insights of Joe Abley, Stuart Cheshire, Paul Hoffman, Warren Kumari, Eliot Lear, Kerry Lynn, Juergen Schoenwaelder, and Dave Thaler. Kerry Lynn deserves special gratitude for his energy and persistence in pressing unanswered questions. Doug Otis sent many comments about visual confusability.

Author’s Address

Andrew Sullivan
Oracle Corporation
100 Milverton Drive
Mississauga, ON L5R 4H1
Canada

Email: andrew.s.sullivan@oracle.com