Internet Draft   Paul Hoffman, Editor
draft-ietf-idn-ace-report-00.txt
June 14, 2001
Expires in six months

Report of the IDN ACE Design Team

Status of this memo

Abstract

This document is a summary of the work of the ACE design team of the Internationalized Domain Name (IDN) Working Group. If the IDN WG selects a single ACE, the design team suggests DUDE. There are many factors that the IDN WG might consider that may lead it to choose a different ACE; those factors and proposals that some of the design team members favored are also described in this document.

1. Introduction

The chairs of the IDN WG appointed an ACE design team to study the many ACE proposals that had come to the working group and to make a recommendation based on that study. The design team consisted of Adam Costello, Paul Hoffman, Makoto Ishisone, David Lawrence, Brian Spolarich, and Rick Wesson. There were three advisors: Marc Blanchet, Patrik Faltstrom, and Erik Nordmark.

The design team evaluated the large number of ACEs that have been proposed in the IDN WG. In comparing them, we looked primarily at two factors:

- how easy they are to understand and implement

- whether they would restrict long names that are likely to be used

Our discussions led us to discover that neither factor was particularly easy to measure. Given that it was hard to measure either factor, it was also difficult to decide how to weigh the two of them against each other.

2. Recommendation

Based on the two factors, the design team recommends to the IDN WG that it picks the DUDE algorithm as the ACE to be used in its protocol. There was general agreement that DUDE was fairly easy to implement (particularly with the design changes starting with the -02 draft) and did not restrict long names that were likely to be used in domain names.

There was not complete agreement in the design team on recommending DUDE. Members disagreed about how much less complex DUDE was compared to the other proposed ACEs. In addition, some members felt that price of higher complexity of other proposed ACEs was worth the greater compression that they give.

The design team chose the DUDE algorithm after the release of the -02 draft, which has some significant design changes from earlier drafts. Even among the DUDE supporters, there was not universal acclaim. Some felt that the discussion of "mixed-case annotation" should be removed, but were willing to recommend the protocol anyway and ask the IDN WG to remove that optional part of the protocol later.

It is important to note that the ACEs we considered most strongly do not provide for special treatment of any particular script or language. The design team members felt that there was no way to provide for such handling that would not dramatically increase the complexity of the protocol, and the apparent benefits in efficiency were relatively modest. All the algorithms provide for relatively efficient treatment of all scripts, and do not impose unreasonable limitations on label size for users of particular scripts; the variation for particular scripts is small in the proposed ACEs.

3. Weighing the Design Goals

3.1 Complexity

It is very difficult to analyze how complex an algorithm is. The proposed ACE algorithms had different types of complexity and were therefore difficult to compare accurately. For example, it is not clear how to compare the complexity of a two-pass algorithm such as RACE or LACE with one-pass algorithm with binary arithmetic such as DUDE. It was pointed out that other algorithms that are quite complex have been implemented well on the Internet.

It is not clear how important complexity is in the long run. One argument says that most applications that use ACE will use an ACE conversion toolkit supplied by an outside source, and there is likely to be only a small number of such toolkits. An opposing argument is that, even if that is true in most cases, there still has to be dozens if not hundreds of toolkits for the various platforms on which IDN will be supported. Further, many companies insist on writing all their own software, even when it is complex (the IPsec market is a good example of this).

3.2 Restrictions on long names that are likely to be used

The IDN WG had earlier agreed with the statement that the purpose of compression is not to reduce the number of octets on the wire, but to allow longer sensible name parts within the 63-octet limit. Unfortunately, it is impossible to determine how long "sensible name parts" would be in various scripts and languages. Some of what makes a name part sensible is its usefulness in non-computer environments such as on billboards, business cards, and radio commercials. Stringing together many words is common in most languages, but it reduces the reproducibility of a name.

The other side of this argument is that the domain name system requires every name at a particular level of the name hierarchy to be unique. It is quite common to see English names in the .com zone that clearly are not the first choice of the companies or people who got them, most likely because the desired (shorter) name was already taken. Because of name exhaustion and the currently tightly-restricted choice in the TLD zone, the length of sensible names is higher than it might be with more TLDs available.

After asking many language experts, some of the people on the design team came to the conclusion that 15 characters for Han-based languages and 30 characters for alphabetic-based languages would put very few restrictions on names that would reasonably be expected to be used. Of course, any limit can be viewed as too restrictive, even the 63 character limit for current names. For example, the name:

computerengineeringdepartmentatuniversityofcaliforniasantabarbara

makes linguistic sense, but is unlikely to be used because it runs together too many words, and would be unwieldy to type.

4. Analysis of the ACEs

The ACE drafts considered by the design team are listed here. Note that these are not long-term documents and are therefore not listed in the references section of this document.

4.1 All ACE proposals

draft-ietf-idn-altdude-00.txt -- AltDUDE. Withdrawn by author.

draft-ietf-idn-amc-ace-*-00.txt -- A series of one-step encodings with varying degrees of complexity and compression.

draft-ietf-idn-brace-00.txt -- BRACE: Bi-mode Row-based ASCII-Compatible Encoding for IDN. Withdrawn by author.

draft-ietf-idn-dude-02.txt -- Differential Unicode Domain Encoding (DUDE). Uses a one-step encoding that uses the binary XOR of successive characters, encoded with Base32.

draft-ietf-idn-dunce-00.txt -- DUNCE: A proposal for a Definitely Unencumbered New Compatible [ACE] Encoding. Specifies multiple different ways to encode strings directly but does not say how to make the encoding unique. Also, does not specify a compression mechanism.

draft-ietf-idn-lace-01.txt -- LACE: Length-based ASCII Compatible Encoding for IDN. Uses a two-step encoding: first compress (using a simple run-length encoding algorithm), then use Base32 on the compressed string.

draft-ietf-idn-race-03.txt -- RACE: Row-based ASCII Compatible Encoding for IDN. This document expired.

draft-ietf-idn-sace-01.txt -- Simple ASCII Compatible Encoding (SACE). This document expired.

draft-ietf-idn-step-00.txt -- StepCode- A User Access Oriented IDN Encoding. Denotes Chinese characters with their phonetic elements. It does not apply to other languages or scripts and is not based on the ISO/IEC 10646 character repertoire.

draft-ietf-idn-utf6-00.txt -- UTF-6 - Yet Another ASCII-Compatible Encoding for IDN. This document expired.

draft-ietf-idn-vidn-01.txt -- Virtually Internationalized Domain Names (VIDN). Uses phonetic transliteration to create ACEs. There were many problems for many languages that were pointed out on the WG mailing list. The proposal is at least partially covered by a patent.

A draft on MACE, Modal ASCII-Compatible Encoding, is expected to be published soon. The design team considered a preliminary version of the encoding described my Makoto Ishisone.

4.2 Primary choices

The design team focused on three classes of ACE: LACE, DUDE, and the AMC series. The ACEs had different levels of complexity and different amounts of compression for mixes of one-row and multi-row input.

The following table summarizes the maximum length for an input string for two cases: the entire string is a typical mix from one row (such as a single-row script), and the entire string is in Han, which usually is a mix of widely-divergent rows. Other comparisons are possible, of course; you might compare how well each ACE does for primarily Latin names (which use a mix from two rows), or names that are mostly non-ASCII characters but use an occasional ASCII character such as a dash.

	Equation for	Max for	Equation	Max for Han
	all one row	all one row	for Han	typical
	typical	typical	typical	
DUDE	1.5n	39	3.8n	15
AMC-W	1.5n	39	1+3n	19
AMC-V	1.5n	39	1+3n	19
LACE	3.2+1.6n	34	1.6+3.2n	17

Two observations come out of this:

- All of the proposals give 34 or more characters for one row typical. Except for strung-together names and some very long German or Thai nouns, that is probably sufficient for most typical names.

- All of the proposals give 15 or more characters for Han typical. Again, that is probably be fine for the vast majority of names, even those with a few sub-names strung together.

Although LACE allows names with two more Han characters than DUDE, the authors of LACE feel that the two-step process is indeed more complicated and therefore did not warrant its use when compared to DUDE.

When compared to DUDE, AMC-W, and AMC-V get four more Han characters with no loss of one-row characters. However, they are both more complicated than DUDE. The members of the group disagreed as to how much more complicated they were, with one group saying that they were "much more complicated" and another group saying "only a little more complicated".

5. Security Considerations

The design team did not perform security reviews on the ACE candidates. A cursory review was done to see whether every Unicode string could result in only one ACE string, and every ACE string could result in zero or one Unicode strings. It is assumed that the authors of each ACE proposal did more intense testing for the one-to-one correspondence.

6. References

References to particular ACE implementations are not given here because none are currently RFCs and it is assumed that only one (or a small number) will eventually reach RFC status.

7. Editor Contact Information

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