Variant Chinese Domain Name Resolution

JENG-WEI LIN
Tunghai University
JAN-MING HO
Academia Sinica
LI-MING TSENG
National Central University
and
FEIPEI LAI
National Taiwan University

Many efforts in past years have been made to lower the linguistic barriers for non-native English speakers to access the Internet. Internet standard RFC 3490, referred to as IDNA (Internationalizing Domain Names in Applications), focuses on access to IDNs (Internationalized Domain Names) in a range of scripts that is broader in scope than the original ASCII. However, the use of character variants that have similar appearances and/or interpretations could create confusion. A variant IDL (Internationalized Domain Label), derived from an IDL by replacing some characters with their variants, should match the original IDL; and thus a variant IDN does. In RFC 3743, referred to as JET (Joint Engineering Team) Guidelines, it is suggested that zone administrators model this concept of equivalence as an atomic IDL package. When an IDL is registered, an IDL package is created that contains its variant IDLs generated according to the zone-specific Language Variant Tables (LVTs). In addition to the registered IDL, the name holder can request the domain registry to activate some of the variant IDLs, free or by an extra fee. The activated variant IDLs are stored in the zone files, and thus become resolvable. However, an issue of scalability arises when there is a large number of variant IDLs to be activated.

In this article, the authors present a resolution protocol that resolves the variant IDLs into the registered IDL, specifically for Han character variants. Two Han characters are said to be

The work described in this article was partially supported by National Science Council, R.O.C., under Grant No. NSC95-2422-H-001-008, NSC96-2422-H-001-001, and NSC96-3113-H-001-011.

Authors' addresses: J.-W. Lin, Department of Information Management, Tunghai University, Taichung 407, Taiwan; Tel. +886-4-23590121 ext. 35906; Fax: +886-4-23504930; email: jwlin@thu.edu.tw; J.-M. Ho, Institute of Information Science, Academia Sinica, Taipei 115, Taiwan; L.-M. Tseng, Department of Computer Science and Information Engineering, National Central University, Taoyuan 32001, Taiwan; F. Lai, Department of Computer Science and Information Engineering and Department of Electronic Engineering, National Taiwan University, Taipei 106, Taiwan.

© 2008 ACM 1530-0226/2008/10-ART11 $5.00 DOI: 10.1145/1450295.1450296. http://doi.acm.org/10.1145/1450295.1450296.

variants of each other if they have the same meaning and are pronounced the same. Furthermore, Han character variants usually have similar appearances. It is not uncommon that a Chinese IDL has a large number of variant IDLs. The proposed protocol introduces a new RR (resource record) type, denoted as VarIdx RR, to associate a variant expression of the variant IDLs with the registered IDL. The label of the VarIdx RR, denoted as the variant index, is assigned by an indexing function that is designed to give the same value to all of the variant IDLs enumerated by the variant expression. When one of the variant IDLs is accessed, Internet applications can compute the variant index, look up the VarIdx RRs, and resolve the variant IDL into the registered IDL.

The authors examine two sets of Chinese IDLs registered in TWNIC and CNNIC, respectively. The results show that for a registered Chinese IDL, a very small number of VarIdx RRs, usually one or two, are sufficient to activate all of its variant IDLs. The authors also represent a Web redirection service that employs the proposed resolution protocol to redirect a URL addressed by a variant IDN to the URL addressed by the registered IDN. The experiment results show that the proposed protocol successfully resolves the variant IDNs into the registered IDNs.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network protocols—*Applications*; C.2.5 [**Computer-Communication Networks**]: Local and wide-area networks—*Internet*; H.5.2 [**Information Interfaces and Presentation**]: User interfaces—*Natural language*

General Terms: Human Factors, Languages, Algorithms, Experimentation

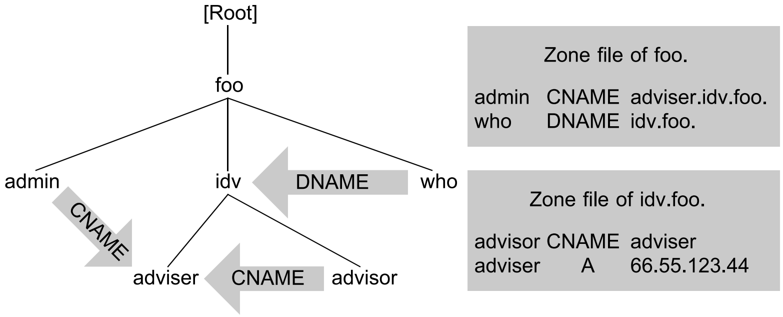
Additional Key Words and Phrases: internationalized domain name, Han character variant, localization, IDN spoof, conversion between traditional Chinese and simplified Chinese, Han character folding

ACM Reference Format:
Lin, J.-W., Ho, J.-M., Tseng, L.-M., Lai, F. 2008. Variant Chinese domain name resolution. ACM Trans. Asian Lang. Inform. Process. Article 11 (November 2008), 29 pages.
DOI = 10.1145/1450295.1450296. http://doi.acm.org/10.1145/1450295.1450296.

1. INTRODUCTION

Many efforts in past years have been made to lower the linguistic barriers for non-native English speakers to access the Internet. However, the traditional Internet Domain Name System (DNS) [Lampson 1985; Mockapetris 1987; Danzig et al. 1992] does not support multilingual scripts. Traditionally, the composition of domain labels[1] is restricted to ASCII letters, digits, and

[1] Conceptually, the DNS is a tree-like distributed database. Each node in the DNS tree, called a domain, is given a (domain) label. Note that the root node is labeled as an empty string (NULL). The complete domain name of a node is the concatenation of all the labels on the path from the node to the root node. This is represented in written form as a string of labels listed from left to right and separated by dots. For example, www.thu.edu.tw. is a complete domain name consisting of five labels, including the NULL label of the root node. People usually omit the last dot and the NULL label. As a result, www.thu.edu.tw is used instead. For administrative purposes, the name space of the DNS is partitioned into areas called zones. A zone is a point of delegation in the DNS tree and served by authoritative domain name servers. It starts at a node and extends down to the leaf nodes or to nodes where other zones start. The data associated with the nodes in a zone are stored in the authoritative domain name servers, which authoritatively answer DNS queries about the nodes in the zone. The data are stored as various forms of resource records (RRs) in the configuration files of the domain name servers, called zone files. For example, an A RR is used to associate a domain name with its IP address; a CNAME RR is used to associate an alias with its canonical domain name; and a NS RR is used to associate an authoritative domain name server for a delegated zone.



```
Zone file of foo.

admin   CNAME  adviser.idv.foo.
who     DNAME  idv.foo.
```

```
Zone file of idv.foo.

advisor CNAME  adviser
adviser   A    66.55.123.44
```

Fig. 1. The usage of CNAMR and DNAME RRs.

hyphens (abbreviated as LDH characters). The introduction of IDN (Internationalized Domain Name) allows people to use their native language to name and access Internet hosts. Examples of IDNs are 臺灣大學.tw, しんかんせん.jp, and tschüß.de. The IETF (Internet Engineering Task Force) IDN Working Group focused on the development of a specification to access domain names in a range of scripts that is broader in scope than the original ASCII. This work resulted in Internet standard RFC 3490, referred to as IDNA (Internationalizing Domain Names in Applications), that specifies the encoding and decoding of IDLs (Internationalized Domain Labels), which are Unicode strings, into and out of LDH domain labels [Faltstrom et al. 2003; Hoffman and Blanchet 2003; Costello 2003]. To process IDNs properly, IDNA requires that an IDNA-compliant module be integrated into every Internet application so that the current DNS infrastructure can be used without modification.

However, IDNA does not address the linguistic issues of individual cultures. Domain names usually appear in URLs, e-mail addresses, etc., which are visible to users. It is likely that users recognize domain names as meaningful words or phrases. Consequently, domain name resolution is a string matching problem that takes not only codepoint comparison into account. In the real world, equivalent names (or aliases) can be introduced in many ways: character variant substitution, word variant substitution, content-based synonym substitution, and so on. In the traditional DNS, ASCII letters in a domain name are case-insensitive. In addition, to associate an alias with its registered domain name, the alias is enumerated explicitly by using a CNAME [Lampson 1985; Mockapetris 1987] or DNAME [Crawford 1999] resource record (RR). As shown in Figure 1, ADMIN.FOO, admin.foo, advisor.who.foo, and adviser.idv.foo all refer to adviser.idv.foo. However, most existent systems did not support DNAME RRs when this article was written. Therefore, the domain registrars had to explicitly create these domains and properly configure the corresponding zone files.

IDNA enlarges the name space of domain names by introducing Unicode characters into the DNS. However, the use of characters that have similar interpretations and/or appearances could create confusion. For example, some Cyrillic characters fully coincide with Latin characters in lower case: a

(U+0430[2]), c, e, к, м, o, p, x, y. Also, 12 Cyrillic characters fully coincide with Latin characters in upper case: A (U+0410), B, C, E, H, K, M, O, P, T, X, y. Note that ASCII capital letters and small letters are considered as case-insensitive, for example, ASCII A (U+0041) = ASCII a (U+0061), when they are used in a domain name; as well, in the IDNA architecture, Cyrillic A is case-mapped (or case-folded) to Cyrillic a. However, ASCII a and Cyrillic a are not recognized as the same character. Consequently, a security alert, referred to as IDN spoof, was issued in February 2005 [Secunia Stay Secure 2005]. Similar linguistic issues arise in other languages. For example, a Chinese domain name may have two forms, one in Simplified Chinese and another in Traditional Chinese; a Japanese domain name may have three forms in Kanji, hiragana, and katakana, respectively. Therefore, IDNs are further required to be localized.

Due to region-specific interpretations of strings used in IDNs, the confusion may require different solutions. Individual zone administrators may find it necessary to impose restrictions and procedures to reduce the likelihood of confusion and unstable references within their own environments. Some researches suggested that while these issues are important, they could best be addressed administratively, rather than through restrictions embedded in the protocols [Konishi et al. 2004]. John Klensin proposed a multilayer search architecture that addresses the linguistic issues in a mechanism layered above the DNS [Klensin 2004].

1.1 Han Character Variants

In this article, we address the resolution of variant IDNs into their registered IDNs. Specifically, we target at Han character variants [Zhang et al. 1989; Hanyu da zidian Editorial Committee 1986; Education Ministry of the ROC 2001; State Council of the PRC 1986]. Note that the relationship between Simplified Chinese and Traditional Chinese is a special case of Han character variant relationships. Two Han characters are said to be (character) variants of each other if they have the same meaning and are pronounced the same. Furthermore, Han character variants usually have similar appearances.[3] In other words, they should be matched as equivalent characters. For example, 清 (U+6E05) and 淸 (U+6DF8) are variants of each other, as are 真 (U+771F) and 眞 (U+771E). Chinese people recognize 清真寺 (mosque, U+6E05 U+771F

[2] In this article, the Unicode characters are identified by their positions, or codepoints. The notation U+12AB, for example, indicates the character at the position 12AB (hexadecimal) in the Unicode table. The codepoint of a Unicode character appears right after the first appearance of the character or in the figures.

[3] Han characters are used in many East Asia regions. The characters that have the same pronunciation in one place may have different pronunciations in another place. As well, the characters that have the same meaning in one place may have different meanings in another place. However, this does not conflict with the definition of Han character variants. Han character variants are formed in different ways by different people in different regions and dynasties. The formation of Han character variants is very complicated and beyond the scope of this article. Some Han character variants have similar appearances. However, this does not imply Han characters that have similar appearances are always variants of each other.

U+5BFA) and 淸眞寺 (U+6DF8 U+771E U+5BFA) as equivalents. Clearly, this will cause confusion if the two terms yield different results when they are used to access the same object.

A variant IDN, derived from an IDN by replacing some characters with their variants, should match the original IDN. If variant IDNs are considered as different IDNs by IDN registries and registered by different name holders, the confusion may cause serious consumer protection problems. For example, tricksters may register 國稅局 (U+570B U+7A0E U+5C40) to cheat or defraud users who want to access the Web site of 國税局 (National Taxation Bureau, U+570B U+7A05 U+5C40). In another scenario, cybersquatters may register 中硏院 (U+4E2D U+784F U+9662) to extort money from the name holder who owns 中研院 (Academia Sinica, U+4E2D U+7814 U+9662).

The codepoint substitution of Han character variants is somewhat similar to the case folding of ASCII letters. However, Han character variant relationships are more complicated. Some are context-free while others are context-sensitive. Some are used more frequently than others, and some are only found in ancient literature. In addition, variant relationships recognized in one region may not be recognized in other regions. For example, the variant relationship between 嶽 (U+5DBD) and 岳 (U+5CB3) is context-sensitive. Chinese speakers recognize 五嶽 (the Five Mounts, U+4E94 U+5DBD) as equivalent to 五岳 (U+4E94 U+5CB3), but do not recognize 嶽飛 (U+5DBD U+98DB) as equivalent to 岳飛 (a Chinese hero's name, U+5CB3 U+98DB). 拾 (U+62FE) is equivalent to 十 (U+5341) when they are used for counting. For example, Chinese speakers recognize 拾圓 (ten dollars, U+62FE U+5713) as equivalent to 十圓 (U+5341 U+5713). However, they do not substitute 十 for 拾 in the idiom 拾金不昧 (U+62FE U+91D1 U+4E0D U+6627). Chinese speakers recognize 芸 (U+82B8) as a simplified variant of 蕓 (U+8553), and 艺 (U+827A) as a simplified variant of 藝 (U+85DD). Meanwhile, people in Taiwan also recognize 芸 as a variant of 藝. In some ancient literature, 然 (U+7136) and 燃 (U+71C3) were used interchangeably to mean burning. Today, 然 and 燃 are rarely interchangeable in common usage.

The People's Republic of China published in 1964 and republished in 1986 "A Complete Set of Simplified Chinese Characters" which specifies how Traditional Chinese characters are simplified [State Council of the PRC 1986]. It includes three tables which totally list 2,235 Simplified Chinese characters simplified from 2,261 Traditional Chinese characters. It also includes an appendix which lists 39 selected variants to replace 46 Traditional Chinese characters. Note that according to the simplifying rules, there are other Simplified Chinese characters derived from Traditional Chinese characters, but not listed in the three tables. The relationships between Simplified Chinese characters and Traditional Chinese characters are not always one-to-one mapped. For example, both 發 (U+767C) and 髮 (U+9AEE) are simplified to 发 (U+53D1); both 歷 (U+6B77) and 曆 (U+66C6) are simplified to 历 (U+5386). The interpretations of such Simplified Chinese characters are context-sensitive. Furthermore, some Simplified Chinese characters are frequently-used Traditional Chinese characters, which have their own meanings in Traditional Chinese contexts. For example, 臺 (U+81FA), 檯 (U+6AAF), and 颱 (U+98B1) are simplified to 台 (U+53F0). In Traditional Chinese contexts, 臺 and 台 almost have the same meanings, but 檯, 颱, and 台 seldom do. 乾 (U+4E7E), 幹 (U+5E79), and 榦 (U+69A6) are simplified to 干 (U+5E72). However, the four characters almost have different meanings in Traditional Chinese contexts. Furthermore, 乾 is not simplified to 干 when used in the following contexts: 乾坤 (U+4E7E U+5764), 乾隆 (U+4E7E U+9686), and so on.

Over the centuries, the evolution of Han characters in different regions has given rise to the situation where one conceptual character can have different appearances and be identified by several different codepoints in computers. Consequently, the charset encodings used in different regions, such as BIG5, GB-2312/GBK, JIS X 0208, and KS C 5601[4], might give preference to some character variants. For example, 吳 (U+5433), 吴 (U+5434), and 呉 (U+5449) are the same character with different appearances in BIG5, GB-2312, and JIS X 0208 respectively. They have separate codepoints and are variants of each other in Unicode. Table I shows some other preferences in Han character variants in Chinese, Japanese, and Korean (abbreviated as CJK). Meanwhile, people in different areas may give new meanings to a Han character. For example, 丼 (U+4E3C) in Chinese resembles the sound of throwing a stone into a well (井, U+4E95); in Japanese, 丼 means a food bowl with rice in Japanese cuisine. Today, many Chinese also recognize such usage of 丼.

1.2 Language Variant Tables and IDL Packages

Due to the inherent complexity of Han character variant relationships, generic Han character folding—referring to codepoint-base substitution for Han character variants, similar to case folding of alphabetic characters—is usually treated as technically hard. Furthermore, domain names are usually short strings without a context. For example, 台风 (U+53F0 U+98CE) may refer to 颱風 (typhoon, U+98B1 U+98A8) or 臺風 (stage manner, U+81FA U+98A8). Name holders and Internet users may have different usage of Han character variants, even if they speak the same language and live in the same region. For example, although many people recognize 峰 (U+5CF0) and 峯 (U+5CEF) are context-free character variants, some still prefer 峰 and refuse to use 峯 in their names. Some researchers considered that Han character folding is out of the scope of IDN standardization and should best be addressed administratively [Seng et al. 2001]. On the other hand, in order to minimize the need to do multiple registrations and delegations for variant IDLs, Tseng et al. proposed to convert Han characters between Traditional Chinese and Simplified Chinese if the character variants used in an IDN are context-free [Tseng et al. 2001; Lee et al. 2001].

[4] GB-2312 and GBK are two charsets widely used for Simplified Chinese. GBK is a successor of GB-2312. BIG5 is a charset widely used for Traditional Chinese. JIS X 0208-1990 is a charset for Japanese. KS C 5601-1987 is a charset for Korean.

Table I. Some Preferred Character Variants in Different Han Charsets

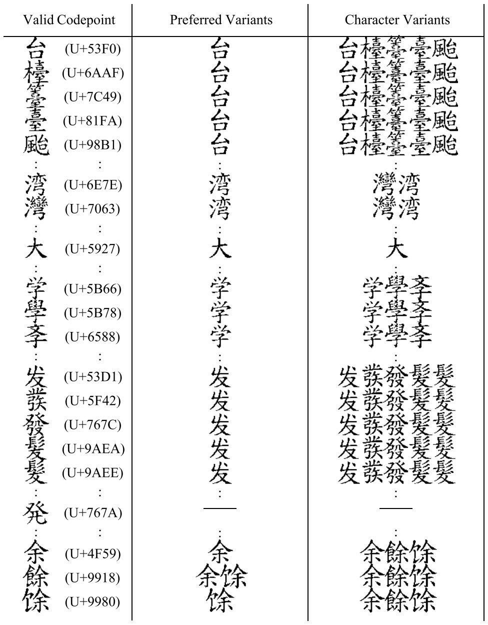
Unicode	BIG5	GB 2312-80	JIS X 0208-1990	KS C 5601-1987	Unicode	BIG5	GB 2312-80	JIS X 0208-1990	KS C 5601-1987
龍(U+9F8D)	龍	-	龍	龍	. . .	. . .	. . .	. . .	. . .
龙(U+9F99)	-	龙	-	-	單(U+55AE)	單	-	單	單
竜(U+7ADC)	-	-	竜	-	单(U+5355)	-	单	-	-
. . .	. . .	. . .	. . .	. . .	単(U+5358)	-	-	単	-
說(U+8AAA)	說	-	-	說	. . .	. . .	. . .	. . .	. . .
说(U+8BF4)	-	说	-	-	鐵(U+9435)	鐵	-	鐵	鐵
説(U+8AAC)	-	-	説	-	铁(U+94C1)	-	铁	-	-
. . .	. . .	. . .	. . .	. . .	鉄(U+9244)	-	-	鉄	-
稅(U+7A05)	稅	-	-	稅	. . .	. . .	. . .	. . .	. . .
税(U+7A0E)	-	税	税	-	溫(U+6EAB)	溫	-	-	溫
. . .	. . .	. . .	. . .	. . .	温(U+6E29)	-	温	温	-
清(U+6E05)	清	清	清	-	. . .	. . .	. . .	. . .	. . .
淸(U+6DF8)	-	-	-	淸	淨(U+6DE8)	淨	-	淨	淨
. . .	. . .	. . .	. . .	. . .	浄(U+6D44)	-	-	浄	-
真(U+771F)	真	真	真	-	凈 (U+51C8)	凈	-	-	-
眞(U+771E)	-	-	眞	眞	净(U+51C0)	-	净	-	-

Valid Codepoint		Preferred Variants	Character Variants
台	(U+53F0)	台檯籉臺颱	台檯籉臺颱
檯	(U+6AAF)	檯	台檯籉臺颱
籉	(U+7C49)	籉	台檯籉臺颱
臺	(U+81FA)	臺	台檯籉臺颱
颱	(U+98B1)	颱	台檯籉臺颱
:	:	:	:
湾	(U+6E7E)	灣	灣湾
灣	(U+7063)	灣	灣湾
:	:	:	:
大	(U+5927)	大	大
:	:	:	:
学	(U+5B66)	學	学學斈
學	(U+5B78)	學	学學斈
斈	(U+6588)	學	学學斈
:	:	:	:
发	(U+53D1)	發髪	发彂發髪髮
彂	(U+5F42)	發	发彂發髪髮
發	(U+767C)	發	发彂發髪髮
髪	(U+9AEA)	髪	发彂發髪髮
髮	(U+9AEE)	髮	发彂發髪髮
:	:	:	:
発	(U+767A)	—	—
:	:	:	:
余	(U+4F59)	余餘	余餘馀
餘	(U+9918)	餘	余餘馀
馀	(U+9980)	餘	余餘馀

Fig. 2. A snapshot of LVT_{tw}.

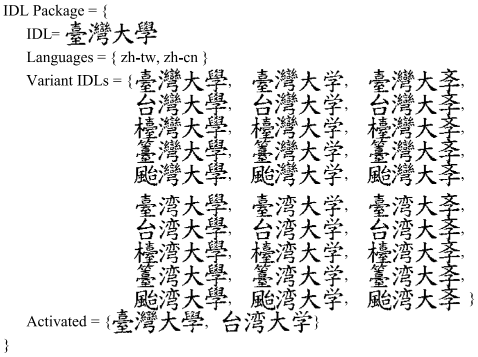
In RFC 3743 [Konishi et al. 2004], referred to as JET (Joint Engineering Team) Guidelines, a set of IDN registration and administration guidelines is defined for applying restrictions to CJK scripts and the zones that use these scripts. A domain registry could define its own local rules for permitted characters and the handling of IDLs and their variants. The Language Variant Table (LVT) mechanism is used to enforce language-based character variant preferences. In a LVT, each row lists a valid character that is permitted to be used in an IDL, its preferred variants and other variants. As shown in Figure 2 and Figure 3, LVT_{tw} [TWNIC 2005] and LVT_{cn} [CNNIC 2005] are the LVTs for Traditional Chinese and Simplified Chinese, submitted to IANA (Internet Assigned Numbers Authority) by TWNIC (Taiwan Network Information Center, the domain registry for *.tw*) and CNNIC (China Internet Network Information Center, the domain registry for *.cn*) respectively.

When an IDL is registered, a collection of its variant IDLs, known as an IDL package, is created according to the zone-specified LVTs. All variant IDLs in the IDL package are unavailable to other name holders. In addition to the registered IDL, the name holder can request the domain registry to activate some of the variant IDLs in the IDL package, free or by an extra fee. Activated IDLs are stored in the zone files and thus become resolvable, and other IDLs are reserved. If a domain registry allows an IDL to associate with more than one language or script, multiple LVTs will be used to create the IDL package. For example, TWNIC and CNNIC employ a registration policy that allows an IDL to associate with both Traditional Chinese and Simplified Chinese. When an IDL 臺灣大學 (National Taiwan Uni-



Valid Codepoint		Preferred Variants	Character Variants
台	(U+53F0)	台	台檯籉臺颱
檯	(U+6AAF)	台	台檯籉臺颱
籉	(U+7C49)	台	台檯籉臺颱
臺	(U+81FA)	台	台檯籉臺颱
颱	(U+98B1)	台	台檯籉臺颱
⋮	⋮	⋮	⋮
湾	(U+6E7E)	湾	灣湾
灣	(U+7063)	湾	灣湾
⋮	⋮	⋮	⋮
大	(U+5927)	大	大
⋮	⋮	⋮	⋮
学	(U+5B66)	学	学學斈
學	(U+5B78)	学	学學斈
斈	(U+6588)	学	学學斈
⋮	⋮	⋮	⋮
发	(U+53D1)	发	发彂發髪髮
彂	(U+5F42)	发	发彂發髪髮
發	(U+767C)	发	发彂發髪髮
髪	(U+9AEA)	发	发彂發髪髮
髮	(U+9AEE)	发	发彂發髪髮
⋮	⋮	⋮	⋮
発	(U+767A)	——	——
⋮	⋮	⋮	⋮
余	(U+4F59)	余	余餘馀
餘	(U+9918)	余馀	余餘馀
馀	(U+9980)	馀	余餘馀

Fig. 3. A snapshot of LVT_{cn}.

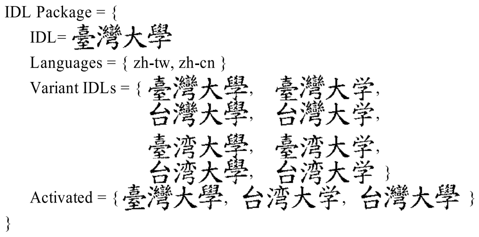


```
IDL Package = {
    IDL= 臺灣大學
    Languages = { zh-tw, zh-cn }
    Variant IDLs = { 臺灣大學, 臺灣大学, 臺灣大斈,
                     台灣大學, 台灣大学, 台灣大斈,
                     檯灣大學, 檯灣大学, 檯灣大斈,
                     籉灣大學, 籉灣大学, 籉灣大斈,
                     颱灣大學, 颱灣大学, 颱灣大斈,

                     臺湾大學, 臺湾大学, 臺湾大斈,
                     台湾大學, 台湾大学, 台湾大斈,
                     檯湾大學, 檯湾大学, 檯湾大斈,
                     籉湾大學, 籉湾大学, 籉湾大斈,
                     颱湾大學, 颱湾大学, 颱湾大斈 }
    Activated = {臺灣大學, 台湾大学}
}
```

Fig. 4. An example of IDL packages.

versity, U+81FA U+7063 U+5927 U+5B78) is registered, an IDL package is created according to LVT_{tw} and LVT_{cn}. By substituting one or more characters in the IDL with their variants, the package includes 30 IDLs that are mutually variant IDLs of each other, as shown in Figure 4. While two of them, 臺灣大學 and 台湾大学, are activated, others are reserved. Note that we can use the expression [臺台檯籉颱] [灣湾][大][學学] to enumerate the set of the variant IDLs in this package. The expression is an OR-AND-like expression [IEEE Standard 1003.2-1992], in which the n-th bracketed clause lists the character variants of the n-th character in the IDL and the



```
IDL Package = {
    IDL= 臺灣大學
    Languages = { zh-tw, zh-cn }
    Variant IDLs = { 臺灣大學, 臺灣大学,
                     台灣大學, 台灣大学,
                     臺湾大學, 臺湾大学,
                     台湾大學, 台湾大学 }
    Activated = { 臺灣大學, 台湾大学, 台灣大學 }
}
```

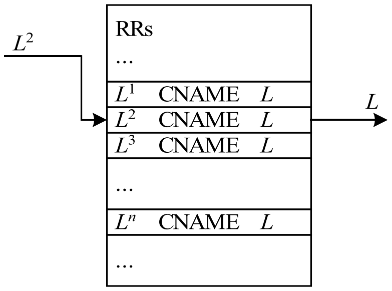
Fig. 5. A reduced version of the IDL package shown in Figure 4.

concatenation of a character in each bracketed clause forms a variant IDL.[5] We denote the expression as the variant expression of this IDL package in this article. In some cases, context-sensitive character variants might be improper in an IDL package, such as 檯, 籉, and 颱 in this example. A domain registry may optionally impose additional rules and processing activities to reduce an IDL package. For example, at the moment the authors wrote this article, when registering an IDL in TWNIC, name holders might see a Web page on which they could manually select proper character variants among all character variants listed in the zone-specific LVTs. Figure 5 shows a reduced version of the IDL package shown in Figure 4, after the domain registry has removed some improper variant IDLs, and the name holder has requested to activate one more variant IDL, 台灣大學. Again, we can enumerate the variant IDLs in the reduced IDL package by the variant expression [臺台] [灣湾][大][學学].

An IDL package is an atomic unit for IDL registration. Once an IDL package is created, no IDLs can be inserted into or removed from the IDL package during its lifetime. However, the name holders can dynamically request the domain registry to activate and deactivate some variant IDLs. When the IDL package is destroyed, due to either unregistered or revoked, all IDLs in the package are available again to all name holders at the same time.

Note that the complete and proper IDL packages from the viewpoints of the domain registry, name holder, and users may still differ. People may have different usage of some character variants even though they live in the same region and speak the same language. Furthermore, some popular domain names may be demanded by many name holders. However, an IDL should be owned by one name holder in the domain name system. That is, IDL packages should not overlap. A domain registry has to take some administrative processes to prevent from overlap with IDL packages. In general, a domain registry can employ some FCFS (first come, first serve)-like registration policy. A name holder can claim the ownership of an IDL package including an IDL and its variant IDLs according to the domain specific LVTs. When an IDL is registered, the name holder can ask help from linguists to create a proper IDL package. When there is disputation, that is, there are two or more name holders claiming the ownership of an IDL, the name holders or domain registry may refer to the

[5] A variant expression could also be viewed as a special regular expression that is a concatenation of bracketed character classes.



RRs
...
L^1 CNAME L
L^2 CNAME L
L^3 CNAME L
...
L^n CNAME L
...

Fig. 6. Enumeration of the variant IDLs via CNAME RRs. $L^1, L^2, \ldots, L^n$ are variant IDLs of an IDL L.

local laws of trademark or other related specifications to resolve the disputation. Once the ownership of an IDL or its variant IDL is transferred or revoked, it may result in the revocation of the whole IDL package and recreation of a new IDL package.

1.3 Activation of Variant IDLs

Zone-specific LVTs provide a mechanism for domain registries and name holders to find possible variant IDLs and the atomicity of IDL packages suppress the possibility of problems such as IDN spoofing, IDN cybersquattering, and so on. The JET Guidelines focuses upon the administration of IDL registration; however, it does not address the implementation issue.

It is intuitive for domain registries to enumerate each of the activated variant IDLs through DNAME and CNAME RRs explicitly so that they become resolvable, as shown in Figure 6. However, an issue of scalability arises when there are a large number of variant IDLs to be activated. The DNS is a hierarchical distributed database. It usually requires seamless cooperation from multiple domain registries to make a correct configuration so that the DNS can provide reliable and efficient domain name resolution. When the number of variant IDLs is large, it becomes very complex for domain registries to keep track of activated and reserved-only variant IDLs and to make sure which are resolvable and which are not. In addition, some Internet services need more than domain name resolution. For example, Web and e-mail services use a domain name to identify a network object, such as a Web page, an e-mail address, and so on. When a user requests a network object via a variant IDN, the server will reject the request because generic string comparison does not match the variant IDN with its registered IDN. Therefore, administrators of such services have to explicitly enumerate the variant IDNs as aliases into the server configuration files. However, the number of an IDN's variant IDNs is usually much larger. For an IDN composed of three IDLs, each of which has three variant IDLs, the number of its variant IDNs is 64. In Section 3, we will see that such IDNs are not unusual when Han characters are used. At the moment the authors wrote this article, in addition to the registered IDL, TWNIC and CNNIC by default activated two variant IDLs, one consists of only Traditional Chinese characters, and the other consists of Simplified Chinese characters.

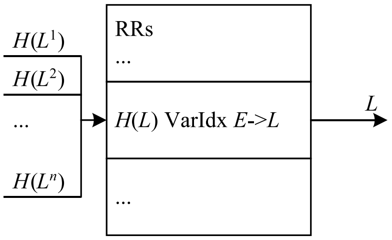


Fig. 7. Enumeration of the variant IDLs via VarIdx RRs. $L^1, L^2, \ldots, L^n$ are variant IDLs of an IDL L. H is the indexing function. $H(L^1) = H(L^2) = \ldots = H(L^n)$. E is the variant expression of $L^1, L^2, \ldots, L^n$.

However, it is not easy for generic users to understand why some variant IDLs are resolvable and others are not. Since most existent systems did not support DNAME RRs, only a few IDLs are registered as a new zone, and most of them are used only for Web access. In Lin et al. [2006], it is shown that many unsuccessful IDN accesses in the Internet are caused by incorrect configuration of domain name servers and Web servers. A mechanism to resolve the variant IDNs into the registered IDN is therefore needed.

This article presents a resolution protocol to resolve the variant IDLs in an IDL package into the registered IDL. Instead of using DNAME or CNAME RRs to enumerate each of the variant IDLs, we introduce a new RR type, denoted as VarIdx, to associate a variant expression, such as [臺台][灣湾][大][學学] in the previous example, with the registered IDL, as shown in Figure 7. The label of a VarIdx RR, denoted as the variant index, is computed by an indexing function that is designed to give the same variant index to all of the variant IDLs enumerated by the variant expression. When an IDL is registered, the registrar partitions its variant IDLs according to the indexing function so that all variant IDLs in a part have the same variant index. The registrar then stores a VarIdx RR for each part. When one of the variant IDLs is used to access a network object, Internet applications can compute the variant index, look up the VarIdx RRs, and finally, resolve the variant IDL into the registered IDL.

This article also presents a Web redirection service that employs our resolution protocol to resolve the variant IDLs in user requests and then redirects the requests by replacing the variant IDLs with the registered IDLs. We build a three-level trial IDN service of six domains in which the registrar stores the registered IDLs, as well as the related VarIdx RRs into the zone files. Meanwhile, we set up a Web redirection server that uses VarIdx RRs to resolve the variant IDLs in an IDL package into the registered IDL. In each domain, a "*" RR, that will match any unregistered domain label, is stored in the zone files so that user requests with unregistered domain labels in this domain will be sent to the redirection server. On receipt of a user request, the redirection server finds the unregistered label, computes its variant index, and looks up the VarIdx RRs. If the right VarIdx RR is found, the request can be redirected to the URL with the registered IDL. Label by label, user requests with variant IDNs can be redirected to the URLs with the registered IDNs.

The remainder of this article is organized as follows. In Section 2, we describe the construction of the indexing functions, the partition of the variant IDLs, and the resolution protocol. In Section 3, we further study different indexing functions based on different LVTs. In Section 4, we present a multilevel redirection service that employs our resolution protocol to resolve the variant IDNs in user requests into the registered IDNs. Finally, in Section 5, we present our conclusions.

2. VARIANT IDL RESOLUTION

2.1 Variant Expression

When an IDL is registered, an IDL package is created according to the zone-specific LVTs.

As the examples shown previously, given a registered IDL $L = X_1^1 X_2^1 \cdots X_d^1$ and that $X_i^2, \cdots, X_i^{n_i}$ are variants of X_i^1 listed in the LVTs for $1 \leq i \leq d$, the variant IDLs of L can be enumerated by the variant expression

$$E = \left[X_1^1 X_1^2 \cdots X_1^{n_1}\right] \left[X_2^1 X_2^2 \cdots X_2^{n_2}\right] \ldots \left[X_d^1 X_d^2 \cdots X_d^{n_d}\right].$$

Note that individual registries may optionally impose additional rules and processing activities to modify the IDL package. If some (context-sensitive) variants, say $X_i^{m_i+1}, \cdots, X_i^{n_i}$ for $1 \leq i \leq d$, are improper for L, the variant IDLs in the reduced IDL package can be enumerated by

$$E' = \left[X_1^1 X_1^2 \cdots X_1^{m_1}\right] \left[X_2^1 X_2^2 \cdots X_2^{m_2}\right] \ldots \left[X_d^1 X_d^2 \cdots X_d^{m_d}\right].$$

The numbers of the variant IDLs enumerated by E and E' are $n_1 {*} n_2 {*} \cdots {*} n_d$ and $m_1 {*} m_2 {*} \cdots {*} m_d$ respectively.

2.2 Construction of Indexing Functions

In this subsection, we describe the construction of an indexing function H that will give the same value to the variant IDLs enumerated by a variant expression.

The idea is to assign the same value to character variants. If X and Y are character variants of each other, we assign

$$H(X) = H(Y).$$

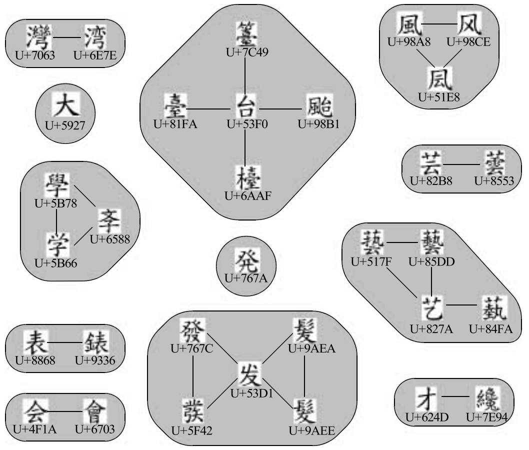
For an IDL $L = X_1 X_2 \cdots X_d$, we assign

$$H(L) = H(X_1)\, H(X_2) \cdots H(X_d),$$

as the variant index. If X' is a character variant of X_i $(1 \leq i \leq d)$, by definition, $L' = X_1 \cdots X_{i-1} X' X_{i+1} \cdots X_d$ is a variant IDL of L. It is easy to see that $H\left(L'\right)$ equals $H(L)$.

We can construct an indexing function as follows.

Character variants are grouped together according to language variant tables. Figure 8 shows a snapshot of the graph of variant groups for the construction of an indexing function H_{twcn} according to LVT_{twcn} (the union of LVT_{tw} and LVT_{cn}). Each node in the graph represents a character and each edge



X	$H_{twcn}(X)$
台	台
檯	台
籉	台
臺	台
颱	台
⋮	⋮
湾	湾
灣	湾
⋮	⋮
大	大
⋮	⋮
学	学
學	学
孛	学
⋮	⋮

X	$H_{twcn}(X)$
才	才
纔	才
⋮	⋮
兿	兿
艺	兿
蓺	兿
藝	兿
⋮	⋮
芸	芸
蕓	芸
⋮	⋮
凨	凨
风	凨
風	凨
⋮	⋮

X	$H_{twcn}(X)$
⋮	⋮
発	発
⋮	⋮
发	发
彂	发
發	发
髪	发
髮	发
⋮	⋮
表	表
錶	表
⋮	⋮
会	会
會	会
⋮	⋮

Fig. 8. A snapshot of variant groups according to LVT_{twcn}.

represents a variant relationship between two character variants. A connected component represents a group of character variants. A character that does not have any variants forms a variant group on its own. Note that because of the transitive law, two characters may be grouped together, even if they are not variants of each other. For example, 發 (U+767C) and 髮 (U+9AEE) are grouped together because they are variants of 发 (U+53D1). After grouping all character variants, we assign a unique value for each variant group. For an n-member variant group $G = \{C_1, C_2, \ldots C_n\}$, we assign

$$H_{twcn}(C_1) = H_{twcn}(C_2) = \ldots = H_{twcn}(C_n) = \mathrm{MIN}(C_1, C_2, \ldots, C_n)$$

where the function MIN() returns the minimal value of the input set.

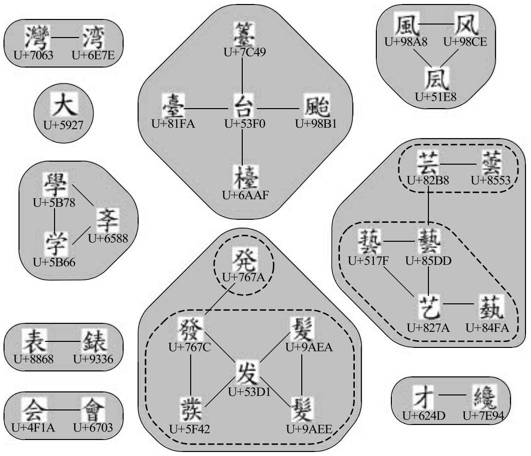
Referring to Figure 8, we have

$$\begin{aligned} H_{twcn}(\text{臺灣大學}) &= H_{twcn}\,(\text{台灣大學}) = H_{twcn}\,(\text{臺湾大學}) = H_{twcn}\,(\text{台湾大學}) \\ &= H_{twcn}(\text{臺灣大学}) = H_{twcn}\,(\text{台灣大学}) = H_{twcn}\,(\text{臺湾大学}) \\ &= H_{twcn}\,(\text{台湾大学}) = \text{台湾大学}. \end{aligned}$$

2.3 Partition of Activated Variant IDLs

For each zone, the domain registry can construct an indexing function according to the zone-specific LVTs such that the indexing function will give the same variant index to all of the variant IDLs in an IDL package registered in that zone. For example, H_{twcn} will give the same variant index to all of the variant IDLs in an IDL package registered in a zone that adopts LVT_{twcn}.

However, individual domains may adopt different LVTs. For example, in some domain, 芸 (U+82B8) is recognized as a variant of 藝 (U+85DD), and 発 (U+767A) as a variant of 發 (U+767C). However, the two variant relationships are not listed in LVT_{twcn}, as shown in Figure 8. We assume that a name holder registers an IDL 才藝發表會 (U+624D U+85DD U+767C U+8868 U+6703) in



X	$H'(X)$
台	台
檯	台
籉	台
臺	台
颱	台
⋮	⋮
湾	湾
灣	湾
⋮	⋮
大	大
⋮	⋮
学	学
學	学
斈	学
⋮	⋮

X	$H'(X)$
才	才
纔	才
⋮	⋮
兿	兿
艺	兿
蓺	兿
藝	兿
⋮	⋮
芸	兿
蕓	兿
⋮	⋮
凮	凮
风	凮
風	凮
⋮	⋮

X	$H'(X)$
⋮	⋮
発	发
⋮	⋮
发	发
彂	发
發	发
髪	发
髮	发
⋮	⋮
表	表
錶	表
⋮	⋮
会	会
會	会
⋮	⋮

Fig. 9. The construction of H'.

that domain and its variant IDLs are enumerated by the variant expression E^*=[才][藝兿艺芸][發彂发発][表][會会]. Refer to Figure 8, H_{twcn}(藝)=H_{twcn}(兿) = H_{twcn}(艺) $\neq$ H_{twcn}(芸) and H_{twcn}(發)=H_{twcn}(彂) = H_{twcn}(发) $\neq$ H_{twcn}(発). Thus, H_{twcn}(才藝發表會) $\neq$ H_{twcn}(才藝発表會). H_{twcn} will not give the same variant index to all of the variant IDLs enumerated by E^*.

The domain registry can construct a proprietary indexing function H', whose graph of variant groups is shown in Figure 9, such that H' gives the same variant index to all of the variant IDLs enumerated by E^*. On the other hand, the domain registry can use H_{twcn} to partition the set {藝, 兿, 艺, 芸} into two subsets {藝, 兿, 艺} and {芸}, and also partition {發, 彂, 发, 発} into {發, 彂, 发} and {発}. Therefore, the variant IDLs enumerated by E^* can be partitioned into four parts as follows:

[才][藝兿艺][發彂发][表][會会]
[才][藝兿艺][発][表][會会],
[才][芸][發彂发][表][會会], and
[才][芸][発][表][會会].

As a result, H_{twcn} gives 才兿发表会, 才兿発表会, 才芸发表会, and 才芸発表会 respectively to the variant IDLs in each of these four parts. Note that the four variant indices can be enumerated by the variant expression [才][兿芸][发発][表][會会].

Given an indexing function H, for a registered IDL $L = X_1^1 X_2^1 \cdots X_d^1$ and the set of its variant IDLs enumerated by a variant expression $E = \left[X_1^1 X_1^2 \cdots X_1^{m_1}\right]\left[X_2^1 X_2^2 \cdots X_2^{m_2}\right]\ldots\left[X_d^1 X_d^2 \cdots X_d^{m_d}\right]$, where $X_i^2, \cdots, X_i^{m_i}$ are proper variants of X_i^1 for $1 \leq i \leq d$, we can decompose E such that

$$E = \bigcup_{\substack{1 \leq j_i \leq k_i \\ 1 \leq i \leq d}} \left[\Pi_1^{j_1}\right]\left[\Pi_2^{j_2}\right]\cdots\left[\Pi_d^{j_d}\right],$$

where $\Pi_i^1, \Pi_i^2, \cdots, \Pi_i^{k_i}$ is a partition of $\left\{X_i^1, X_i^2, \cdots, X_i^{m_i}\right\}$ according to H for $1 \le i \le d$; that is,

(1) $\Pi_i^1 \cup \Pi_i^2 \cup \cdots \cup \Pi_i^{k_i} = \left\{X_i^1, X_i^2, \cdots, X_i^{m_i}\right\}$,
(2) $\Pi_i^a \cap \Pi_i^b = \emptyset$, if $a \neq b$,
(3) $H(x) = H(y)$, if $x, y \in \Pi_i^a$, and
(4) $H(x) \neq H(y)$, if $a \neq b$, $x \in \Pi_i^a, y \in \Pi_i^b$.

In other words, we can decompose E into $k_1{*}k_2{*}\cdots{*}k_d$ parts, each of which is enumerated by $\left[\Pi_1^{j_1}\right]\left[\Pi_2^{j_2}\right]\cdots\left[\Pi_d^{j_d}\right]$, where $1 \le j_i \le k_i$ for $1 \le i \le d$, such that H will give the same variant index to all variant IDLs in a part. The number of variant IDLs enumerated by E is

$$|E| = m_1{*}m_2{*}\cdots{*}m_d.$$

The set of different variant indices is

$$\begin{aligned} I_H(E) &= Unique(\{H(L')|L' \in E\}) \\ &= \bigcup_{1 \le j_i \le k_i} \left\{ H(x_1)\,H(x_2)\cdots H(x_d) \,\middle|\, x_1 \in \Pi_1^{j_1}, x_2 \in \Pi_2^{j_2}, \cdots, x_d \in \Pi_d^{j_d} \right\}, \end{aligned}$$

and the number of different variant indices is

$$|I_H(E)| = k_1{*}k_2{*}\cdots{*}k_d,$$

where $Unique()$ is the function that returns the set of each unique member in the input set.

2.4 Variant IDL Resolution Protocol

To provide information for variant IDL resolution, a new RR type, denoted as VarIdx, is introduced to associate a variant expression with its registered IDL. When an IDL L is registered, its variant IDLs are partitioned according to an indexing function H. We assume $E = \cup E_i$. For each part E_i, the registrar stores a VarIdx RR in the zone files to associate E_i with L . The label of that RR is $H(L_i)$, where L_i is a variant IDL of L in E_i. The format of the VarIdx RR is as follows:

$$H(L_i) \qquad \text{TTL}_{\text{optional}} \qquad \text{Class}_{\text{optional}} \qquad \textbf{VarIdx} \qquad E_i \rightarrow L^6$$

Instead of storing $|E|$ CNAME (or DNAME) RRs, the registrar stores $|I_H(E)|$ VarIdx RRs in the zone files. For example, in a zone that adopts H_{twcn}, the registrar stores the following VarIdx RR for the IDL 臺灣大學 in the zone files.

H_{twcn} (臺灣大學) VarIdx [臺台][灣湾][大][學学] → 臺灣大學

[6] $\text{TTL}_{\text{optional}}$ and $\text{Class}_{\text{optional}}$ are two optional parameters in a DNS RR, which specify the time-to-live value and address-type of this record, respectively.

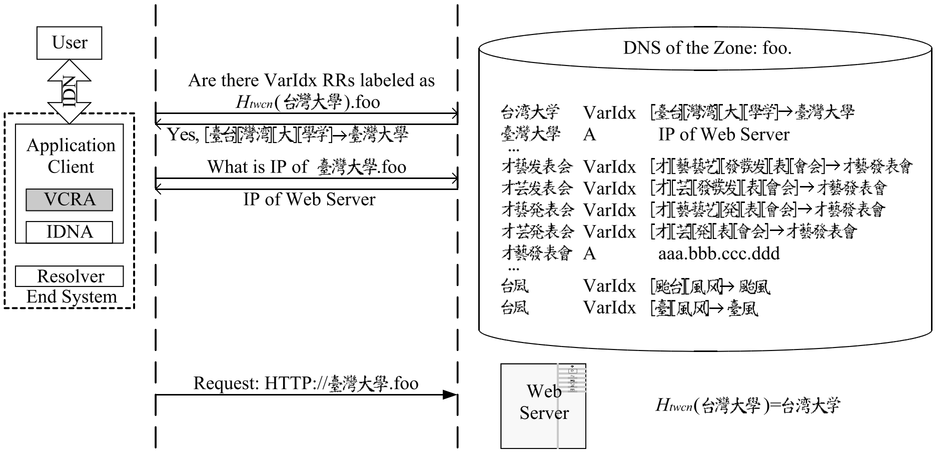

User
IDN
Application Client
VCRA
IDNA
Resolver
End System
Are there VarIdx RRs labeled as Htwcn(台灣大學).foo
Yes, [臺台][灣湾][大][學学]→臺灣大學
What is IP of 臺灣大學.foo
IP of Web Server
Request: HTTP://臺灣大學.foo
DNS of the Zone: foo.
台湾大学 VarIdx [臺台][灣湾][大][學学]→臺灣大學
臺灣大學 A IP of Web Server
...
才藝发表会 VarIdx [才][藝蓺艺][發𤼵发][表][會会]→才藝發表會
才芸发表会 VarIdx [才][芸][發𤼵发][表][會会]→才藝發表會
才蓺発表会 VarIdx [才][藝蓺艺][発][表][會会]→才藝發表會
才芸発表会 VarIdx [才][芸][発][表][會会]→才藝發表會
才藝發表會 A aaa.bbb.ccc.ddd
...
台凮 VarIdx [颱台][颩风]→ 颱颩
台凮 VarIdx [臺][颩风]→ 臺颩
Web Server
Htwcn(台灣大學)=台湾大学

Fig. 10. VCRA (Variant Chinese domain name Resolution in Applications).

For the IDL 才藝發表會, the registrar stores the following four VarIdx RRs in the zone files.

H_{twcn}(才藝發表會)	VarIdx	[才][藝蓺艺][發𤼵发][表][會会] → 才藝發表會
H_{twcn}(才藝発表會)	VarIdx	[才][藝蓺艺][発][表][會会] → 才藝發表會
H_{twcn}(才芸發表會)	VarIdx	[才][芸][發𤼵发][表][會会] → 才藝發表會
H_{twcn}(才芸発表會)	VarIdx	[才][芸][発][表][會会] → 才藝發表會

When an Internet application gets a variant IDL L' , either from user input or application content, it can compute $H_{twcn}(L')$, look up the VarIdx RRs, and find the registered IDL, as shown in Figure 10.

Since an indexing function may give two characters the same value, even if they are not variants of each other, it may give two IDLs the same variant index, even if they are not variant IDLs of each other. For example, H_{twcn} will give the same variant index 台凮 (U+53F0 U+51E8) to the following two IDLs 颱颩 (typhoon, U+98B1 U+98A8) and 臺颩 (stage manner, U+81FA U+98A8), that is,

$$H_{twcn}(\text{颱颩}) = H_{twcn}(\text{臺颩}) = \text{台凮}.$$

If two IDL packages, enumerated by [颱台][颩风] and [臺][颩风] respectively, are both registered, there will be two VarIdx RRs labeled as the same variant index, as shown in Figure 10. This phenomenon is referred to as a collision. Note that the two IDL packages do not overlap. The ownership of 台风 is determined by the domain-specific disputation resolution process. As a result, when one variant IDL of the two IDLs is queried, the domain name server will return two VarIdx RRs at the same time, one for each other. Therefore, Internet applications must verify each of the returned VarIdx RRs to ensure that the right VarIdx RR is used to resolve the variant IDL in question.

Assume H is the adopted indexing function. The pseudo code of variant IDL resolution is as follows:

```
Function ResolveVariantIDL (String $\boldsymbol{L}$, String $\boldsymbol{D}$) {
        // $\boldsymbol{L}$ is the IDL to be looked up
        // $\boldsymbol{D}$ is the domain
        $\boldsymbol{RRs}$ = DNS Lookup ($H(\boldsymbol{L}).\boldsymbol{D}$, VarIdx) // Look up VarIdx RRs labeled as $H(\boldsymbol{L}).\boldsymbol{D}$
        foreach $\boldsymbol{rr}$ in ($\boldsymbol{RRs}$) {
                // There may be more than one VarIdx RRs returned.
                // The resolution protocol should examine each of them to find the right
                // VarIdx RR.
                if ($\boldsymbol{L}$ is included in $\boldsymbol{rr}$.variant_expression) {
                        return $\boldsymbol{rr}$.registered_IDL
                }
        }
        // If there is no VarIdx RR enumerating $\boldsymbol{L}$, $\boldsymbol{L}$ is not registered.
        return Not_Registered
}
```

For an IDL package enumerated by a variant expression E, the space requirement is reduced from $|E|$ CNAME (or DNAME) RRs to $|I_H(E)|$ VarIdx RRs. In Section 3, we will see that although the number of variant IDLs is usually large, the number of variant indices is significantly reduced to 1 or 2 by using a good index function, such as H_{twcn}. Since there are less RRs in the database of the domain name server, the time for each RR lookup is reduced. The time to compute the variant index of an IDL is trivially linear to the length of the IDL by table lookups. However, small devices, such as PDAs or cellular phones, may suffer the large memory size of the indexing function.

Note that individual zones may define their own indexing functions, or just adopt a well-defined indexing function, such as H_{twcn}. From the viewpoint of software practice, there must be a mechanism to specify the indexing function for each zone; otherwise, all zones must use the same indexing function. We will evaluate different indexing functions in Section 3.

The proposed resolution protocol can be integrated into domain name servers, DNS resolvers, application servers, clients, or proxies. In Section 4, we will investigate a server-side redirection service that employs the protocol to redirect user requests to the URLs with the registered IDNs.

3. EVALUATION OF INDEXING FUNCTIONS

In this section, we further study different indexing functions constructed according to different variant tables. We consider an indexing function good when it meets the following requirements: 1) for an IDL, the number of VarIdx RRs required for activating its variant IDLs is small, and 2) there are few collisions of variant indices, even in a large domain in which thousands or maybe more IDLs are registered. Note that on receipt of a VarIdx query, the domain name server will return all VarIdx RRs that associate with the specified variant index. If there are many VarIdx RRs in match, the DNS response will cost

Table II. Statistics of Registered IDLs in CNNIC and TWNIC

	Total number of IDLs	Average length of an IDL	Average number of the variant IDLs of an IDL	Maximum number of the variant IDLs of an IDL
CNNIC	72,020	4.23	5.2	4,096
TWNIC	21,374	6.15	11.1	2,048

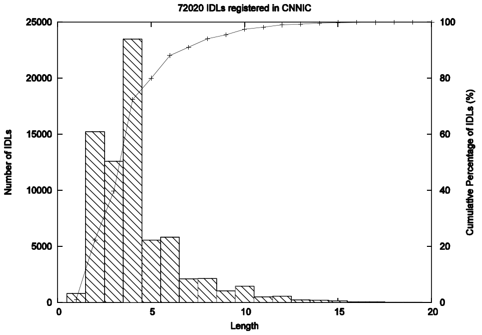


Fig. 11. Distribution of IDLs registered in CNNIC.

a large data transmission on the Internet. We collect Chinese IDLs registered currently and calculate the numbers of VarIdx RRs required for activating all of their variant IDLs and collisions of variant indices when different indexing functions are applied.

We collect Chinese IDLs from two registries, TWNIC for the *.tw* domain and CNNIC for the *.cn* domain. They both implement a registration policy that allows a registered IDL to associate with Traditional Chinese and Simplified Chinese. However, when a Traditional/Simplified Chinese IDL is registered, the name holder can activate only one variant Simplified/Traditional Chinese IDL in the IDL package. Table II shows some statistics of these IDLs. On average, the IDLs registered in CNNIC are shorter than those in TWNIC. Although the average number of variant IDLs that an IDL may have is small, the maximum number is large.

Figure 11 and Figure 12 show the distribution of the registered IDLs in CNNIC and TWNIC respectively, according to their lengths, that is, the number of characters in an IDL. In CNNIC, 72% of the IDLs are shorter than five characters. In TWNIC, 40% of the IDLs are shorter than five characters, but there is a peak distribution (19%) where the length is seven characters. In both registries, long IDLs are usually the names of entities in hierarchical organizations, such as 台北市中山區公所 (the Office of Zhongshan District, Taipei City). It reveals the inadequacy of the current implementation for IDN delegations in the DNS hierarchy.

Figure 13 and Figure 14 show the distribution according to the numbers of their variant IDLs. In CNNIC, 94% of the IDLs have fewer than eight variant IDLs, while in TWNIC, 91% of the IDLs have fewer than 16 variant IDLs. However, 607 IDLs in CNNIC and 861 IDLs in TWNIC have more than 64

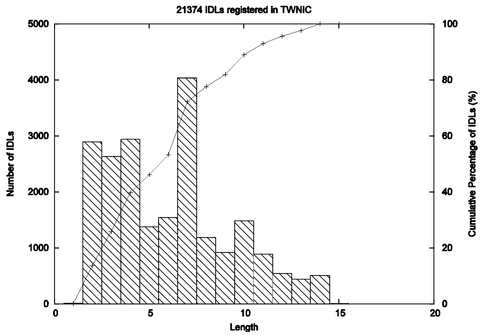


Fig. 12. Distribution of IDLs registered in TWNIC.

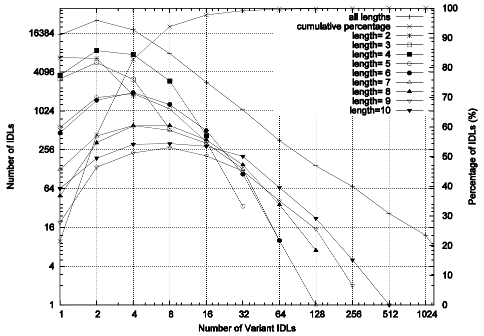


Fig. 13. Number of variant IDLs in CNNIC.

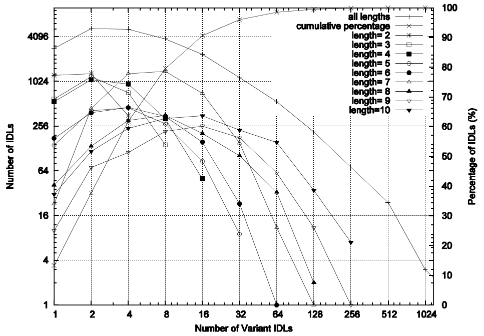


Fig. 14. Number of variant IDLs in TWNIC.

variant IDLs; and 41 IDLs in CNNIC and 28 IDLs in TWNIC have more than 512 variant IDLs. Although the ratio of these IDLs is small, they do have a large number of variant IDLs. Figure 13 and Figure 14 also show the distribution curves of different lengths, which shift to the right as the length increases. This is reasonable, because the longer an IDL, the more variant IDLs it might have. Figure 15 and Figure 16 also exhibit this property.

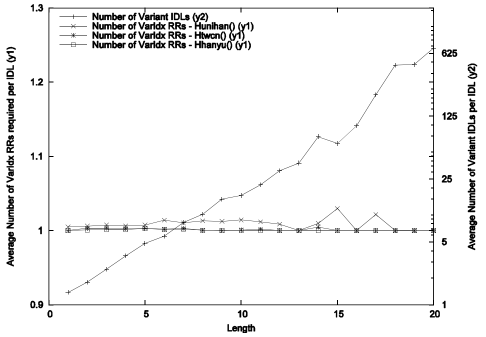


Fig. 15. Number of VarIdx RRs required for CNNIC.

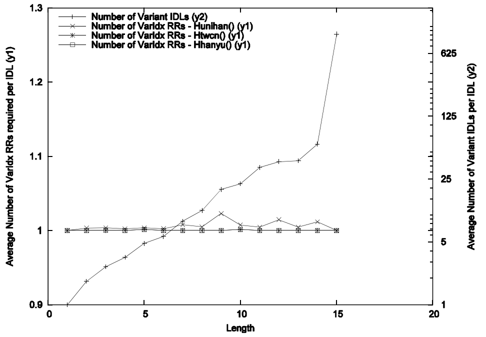


Fig. 16. Number of VarIdx RRs required for TWNIC.

We study three indexing functions in this section. The first is H_{unihan}, constructed according to LVT_{unihan} (the Unihan database [The Unicode Consortium 2002]); the second is H_{twcn}, constructed according to LVT_{twcn}; the third is H_{hanyu}, constructed according to LVT_{hanyu} (the character variant table in Hanyu Da Dictionary [Hanyu da zidian Editorial Committee 1986]). Table III shows some statistics of the three indexing functions. LVT_{hanyu} lists several more variant relationships than the others. Figure 17 and Table IV show some differences among the three index functions.

We calculate the numbers of VarIdx RRs required for activating all of the variant IDLs and collisions of variant indices when these indexing functions are adopted. Figure 15 and Figure 16 show that, on average, for a Chinese IDL, we only need to store one VarIdx RR in the zone files to activate all of its variant IDLs. LVT_{unihan} lists fewer variant relationships. If one relationship out of LVT_{unihan} is used in an IDL registration, H_{unihan} cannot give the same variant index to all of its variant IDLs. Therefore, the IDL package is partitioned and the registries need to store slightly more VarIdx RRs. Table V shows the maximum number of VarIdx RRs required in the two domains. The numbers are all small. In fact, 99.99% of the IDL packages in both *.tw* and *.cn* domains require just one VarIdx RR. The scalability issue that arises

Table III. Some Statistics of the Three Indexing Functions

	Source of variant relationships	Number of variant relationships	Number of variant groups that two or more character variants	Maximum number of character variants in a variant group
H_{unihan}	LVT_{unihan}	3,999	3,142	8
H_{twcn}	LVT_{twcn}	4,911	3,477	8
H_{hanyu}	LVT_{hanyu}	13,010	4,472	81

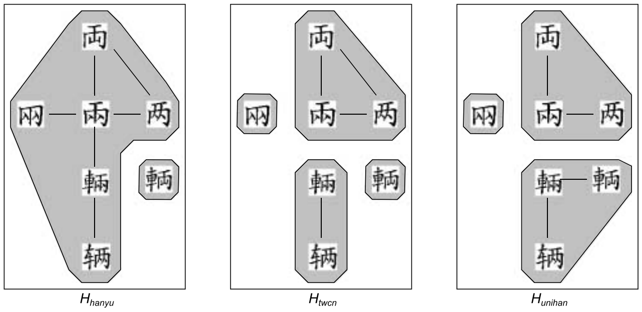


Fig. 17. Snapshots of the variant graphs constructed according to different LVTs.

Table IV. Some Differences Among the Three Indexing Functions

X	$H_{hanyu}(X)$	$H_{twcn}(X)$	$H_{unihan}(X)$
㒳 (U+34B3)	㒳	㒳	㒳
両 (U+4E21)	㒳	両	両
两 (U+4E24)	㒳	両	両
兩 (U+5169)	㒳	両	両
輌 (U+8F0C)	輌	輌	輌
輛 (U+8F1B)	㒳	輛	輌
辆 (U+8F86)	㒳	輛	輌

when there are a large number of variant IDLs to be activated is successfully eliminated.

Table V also shows the maximum numbers of collisions of a variant index (the maximum number of VarIdx RRs that have the same variant index) and the numbers of total collisions (the number of variant indices that are used by two or more VarIdx RRs) in the two domains when the three indexing functions are adopted. Among the 72,020 IDLs and 21,734 IDLs registered in CNNIC and TWNIC respectively, at most two VarIdx RRs have the same variant index when H_{unihan} or H_{twcn} is used and at most five VarIdx RRs have the same index when H_{hanyu} is used. These numbers are small. The data transfer for VarIdx RRs in a single DNS response is small. However, the numbers of total collisions significantly increase when H_{hanyu} is used. LVT_{hanyu} lists several more variant relationships than the other LVTs. Many characters are grouped together even though they are not variants of each other. We observe that the largest variant group in H_{hanyu} has 81 character variants, which are divided into 65 variant groups in H_{unihan} and 62 variant groups in H_{twcn}, respectively. The possibility of collisions increases when H_{hanyu} is adopted.

Table V. Experiment Results when the Three Indexing Functions are Applied

	Maximum number of VarIdx RRs required per IDL		Maximum number of VarIdx RRs that have the same variant index		Number of variant indices that are used by s two or more VarIdx RR	
	CNNIC	TWNIC	CNNIC	TWNIC	CNNIC	TWNIC
H_{unihan}	4	4	2	2	24	8
H_{twcn}	2	2	2	2	18	10
H_{hanyu}	2	2	5	2	1,005	86

LVT_{unihan} lists the fewest variant relationships. Registrars need to store slightly more VarIdx RRs to activate all of the variant IDLs in an IDL package. On the other hand, LVT_{hanyu} lists several more variant relationships. The possibility of collisions increases. We believe that H_{twcn} is a good candidate for global use in resolving variant Chinese IDNs.

4. VARIANT IDN REDIRECTION SERVICES

The proposed resolution protocol can be integrated into the DNS or Internet applications. For some Internet applications, such as Telnet and FTP, which simply use a domain name to look up the IP address for connection setup, integration of the resolution protocol into the DNS can provide transparent variant IDN resolution. However, some Internet applications, such as Web and e-mail, also use domain names to identify network objects. These applications have to resolve a variant IDN into the corresponding registered IDN. Generally speaking, when resolving variant IDLs in the server side, the load on the servers of a single query increases. When resolving variant IDLs in the client side, however, it requires additional messages and introduces a longer delay. Furthermore, resource-constraint devices, such as cellular phones and PDAs, may suffer the large memory size of the indexing function. In this section, we investigate a server-side redirection architecture which employs our resolution protocol to redirect a user request with a variant IDL to the URL with the registered IDL.

4.1 Redirection Service Architecture

To comply with JET Guidelines, when an IDL is registered, an IDL package is created according to LVT_{twcn}. However, we store only the registered IDL in the zone files. We use H_{twcn} to partition its variant IDLs and store a VarIdx RR for each part, as stated in Section 2. The VarIdx RR is simulated by using a TXT RR [Mockapetris 1987] as follows:

variant_index **TXT** **VARIDX:** variant_expression → registered_IDL

We set up a Web redirection server that uses VarIdx RRs to resolve the variant IDL in a user request into the corresponding registered IDL. In the zone files of the IDL domain, we store a "*" RR so that when an unregistered domain label under that domain is queried, the domain name server will return the IP address of the redirection server. Therefore, when a user accesses an (unregistered) variant IDL rather than the corresponding registered IDL, the request will be sent to the redirection server. The redirection server is configured to execute a redirection script that will examine the domain name in the

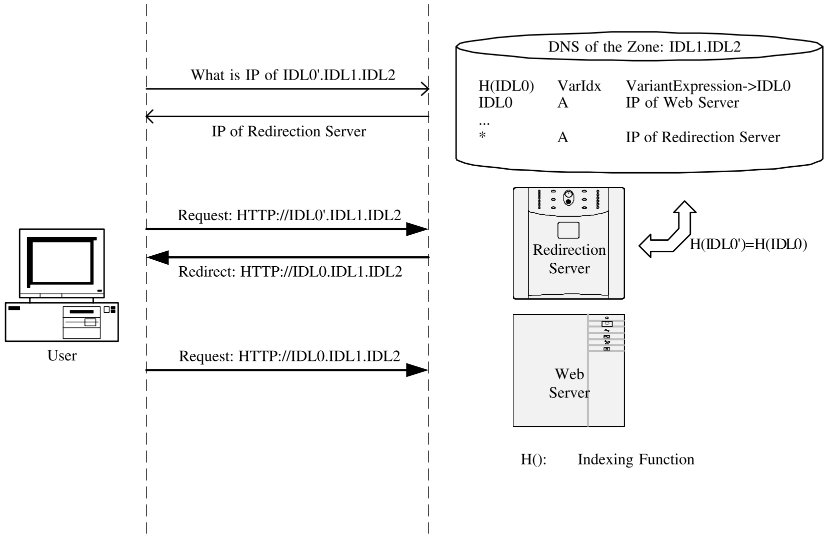

DNS of the Zone: IDL1.IDL2
H(IDL0) VarIdx VariantExpression->IDL0
IDL0 A IP of Web Server
...
* A IP of Redirection Server
What is IP of IDL0'.IDL1.IDL2
IP of Redirection Server
Request: HTTP://IDL0'.IDL1.IDL2
Redirect: HTTP://IDL0.IDL1.IDL2
Request: HTTP://IDL0.IDL1.IDL2
User
Redirection Server
H(IDL0')=H(IDL0)
Web Server
H(): Indexing Function

Fig. 18. A scenario that demonstrates how a URL with a variant IDL is redirected to the URL with the registered IDL.

request header [Fielding et al. 1997] and try to resolve the variant IDL into the registered IDL. When the redirection server receives a user request, the script locates the unregistered domain label, computes the variant index, and looks up the VarIdx RRs. If a VarIdx RR enumerating the unregistered domain label is returned, the registered IDL is found. The script redirects the request to the new URL with the registered IDL, as shown in Figure 18. If there is no VarIdx RR enumerating the unregistered domain label, the script returns an error message to the user.

An IDN service provider can duplicate such redirection service to form a multi-level redirection service that resolves a variant IDN into its registered IDN, label by label, as shown in Figure 19. Note that individual domains can adopt different indexing functions and redirection services in their own environments. They can customize their products independently so that users will experience better services. For simplification, we use the same indexing function and redirection server in the experiment.

4.2 Experiment Results

We built a trial IDN service of a three-level DNS hierarchy consisting of six domains. Figure 20 is a snapshot of the six domains. Under the domain *vcdnr.cc*, we register five IDN subdomains, which have a total of 52 variant IDN subdomains. Under these subdomains, we register 40 IDNs, which have a total of 1,880 variant IDNs. For example, the IDN subdomain 龍的傳人.vcdnr.cc has six variant IDN subdomains enumerated by [龍竜龙][的][傳传][人].vcdnr.cc. Instead of storing six NS RRs and six zone files, we store only one VarIdx RR and one zone file to activate all six variant IDN subdomains. The IDN 岳飛.龍的傳人.vcdnr.cc has 12 variant IDNs enumerated by [岳][飛飞].[龍竜龙][的][傳

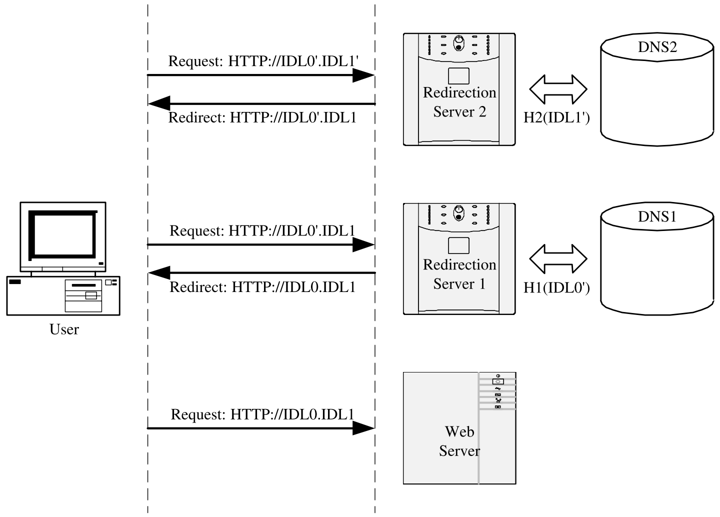

Request: HTTP://IDL0'.IDL1'
Redirect: HTTP://IDL0'.IDL1
Redirection Server 2
H2(IDL1')
DNS2
User
Request: HTTP://IDL0'.IDL1
Redirect: HTTP://IDL0.IDL1
Redirection Server 1
H1(IDL0')
DNS1
Request: HTTP://IDL0.IDL1
Web Server

Fig. 19. Multi-level redirection services.

传][人].vcdnr.cc. Instead of storing 12 A RRs, we store one VarIdx RR and one A RR. Furthermore, we host only one Web site rather than create 12 virtual hosts. Table VI shows some comparisons between the traditional approach and the proposed one.

We set up a generic PC that runs FreeBSD operating system as the redirection server. An Apache Web server that listens on popular HTTP service ports is configured to execute a redirection script when a user wants to access an object on a host whose hostname matches "*.*vcdnr.cc*". In the zone files of each of the six domains, a "*" RR is stored and associates with the IP address of the redirection server.

On receipt of a user request, the script recognizes the charset encoding by converting the requested URL from a candidate charset encoding to Unicode. Note that browsers may send the URL in various charset encodings, such as GB-2312/GBK, BIG5, or UTF-8. The charset encodings to be tested, and their priorities, are determined by the policy of the redirection service. In some zone administrations, the URL can be converted to two or more Unicode URLs, because different charset encodings are used. For example, 公司 (U+516C U+53F8).vcdnr.cc encoded in GB-2312 has the same octet sequence as 鼠伺 (U+9F20 U+4F97).vcdnr.cc encoded in BIG5. We believe that it would be better to prevent this situation during IDL registration. Currently, the script returns a page that lists these URLs so that users can click the correct URL. In other words, we give different charset encodings the same priority currently.

The script then locates the unregistered domain label, computes the variant index, and looks up the VarIdx RRs. If a VarIdx RR that enumerates the unregistered domain label is returned, the script redirects the request to the new URL with the registered IDL. If there are other unregistered domain labels in the new URL, the resolution process will be invoked in a cascading manner in the corresponding subdomain until the redirection server finally returns the

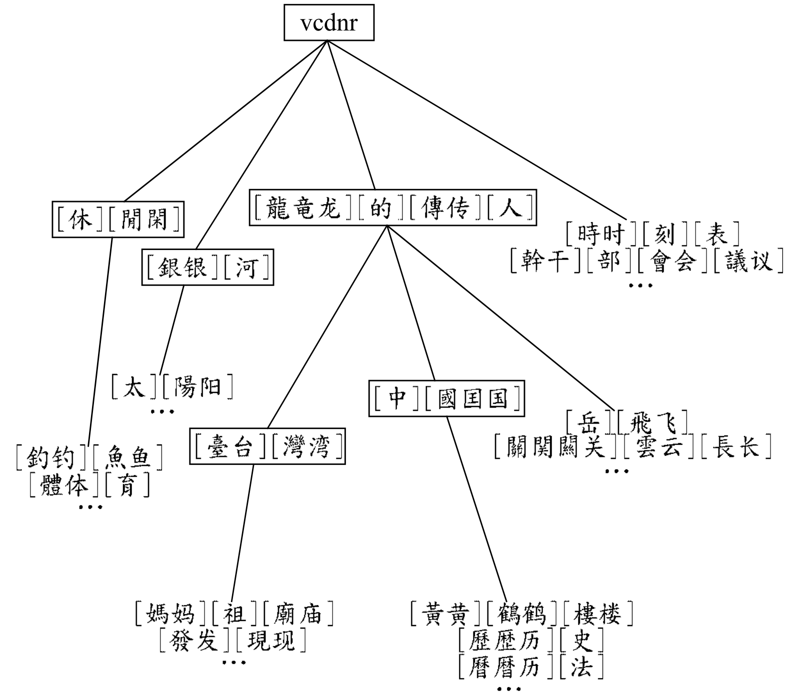


Fig. 20. A snapshot of the trial IDN service.

Table VI. Some Comparisons between the Traditional Approach and the Proposed One

	IDN subdomains	Variant IDN subdomains	Registered RRs	Zone files	IDNs	Variant IDNs	Registered RRs	Web virtual hosts
Traditional	5	52	52 NS	52	40	1,880	1,880 A	1,880
Our approach			5 NS + 5 VarIdx	5			40 A + 40 VarIdx	40

URL with the registered IDN, or the redirection server cannot resolve the unregistered domain label.

We use variant IDNs in different Web browsers on different platforms to access the registered IDNs. Context-free and context-sensitive variants listed in the LVT_{twcn} are used to form the variant IDNs. For example, for the IDN 岳飛.龍的傳人.vcdnr.cc, we use LVT_{twcn} to generate 32 IDNs enumerated by [岳嶽][飛飞].[龍竜龙龔][的][傳传][人].vcdnr.cc. Among these, 12 IDNs enumerated by [岳][飛飞].[龍竜龙][的][傳传][人].vcdnr.cc. are variant IDNs and the others are not. We examine each of these IDNs for IDN resolution. As shown in Table VII, the experiment gives a perfect result. In total, 12,330 IDNs are generated according to LVT_{twcn} for the registered IDNs and IDN subdomains. Among these, 1,932 IDNs are variant IDNs of these registered IDNs and IDN subdomains. They are all successfully resolved into the registered ones. As well, other IDNs are not resolved as expectedly.

Table VII. Result of IDN Resolution

IDNs tested	Variant IDNs	Resolved IDNs	Other IDNs	Resolved IDNs
12,330	1,932=1,880+52	1,932 (100%)	10,398	0 (0%)

Experiment results show that the VarIdx RRs provide sufficient information for the redirection server to redirect user requests with the variant IDNs by resolving the variant IDLs into the registered IDLs.

In this architecture, the initial delay may be long if several redirections are invoked. However, subsequent requests will use the registered IDN and, therefore, no more redirections are required.

5. CONCLUSIONS

The JET Guidelines focuses upon the administration of IDL registration; however, it does not address the implementation issue. An issue of scalability arises when there is a large number of variant IDLs to be activated. When the authors wrote this article, in addition to the registered IDL, TWNIC and CNNIC by default activated two variant IDLs, one consists of only Traditional Chinese characters and the other consists of Simplified Chinese characters. However, it is not easy for generic users to understand why some variant IDLs are resolvable and others are not.

Based on the LVT mechanisms, we present the resolution of the variant IDLs in an IDL package into the registered IDL with the help of VarIdx RRs. To address the issue of scalability that arises when the number of variant IDLs to be activated is large, VarIdx RRs use variant expressions to enumerate the variant IDLs. An indexing function is designed to give the same variant index to the variant IDLs enumerated by a variant expression so that Internet applications can use any of the variant IDLs to look up the VarIdx RRs and find the registered IDL.

Individual zones may have their own rules about permitted characters and the variant relationships of these characters. From the viewpoint of software practice, there must be a mechanism to specify the indexing function for each zone; otherwise, all zones must use the same indexing function. The study in Section 3 indicates that an indexing function may exist for global use. LVT_{unihan} lists the fewest variant relationships. Registries that adopt H_{unihan} may need to store more VarIdx RRs to activate all variant IDLs. LVT_{hanyu} lists several more variant relationships; however, when registries adopt H_{hanyu}, the possibility of collisions increases. We believe that H_{twcn} is a good candidate for global use.

We start with a trial IDN service of six domains. Meanwhile, we set up a redirection server and store a "*" RR in the zone files of each of the six domains so that user requests with unregistered domain labels under these domains will be served by the redirection server. On receipt of a user request, the redirection server computes the variant index of the unregistered domain label and looks up the VarIdx RRs. If the right VarIdx RR is found, the server redirects the user request to the new URL by replacing the variant IDL with the registered IDL. Although the initial delay may be long if several redirections are invoked, subsequent requests will use the registered IDN and no more redirections are required. Our experiments show that VarIdx RRs provide sufficient information for variant IDL resolution.

Human language can take multiple forms, and especially Chinese language can be very flexible. For example, sometimes people may say 臺灣的東海大學 (Tunghai University in Taiwan) instead of 東海大學.臺灣. We will further study the real name redirection in the future.

REFERENCES

China Internet Network Information Center (CNNIC). Retrieved from http://www.cnnic.net.cn.

China Internet Network Information Center (CNNIC). 2005. IANA IDN Languages Table: CN Chinese Character Table.

COSTELLO, A. M. 2003. Punycode: A bootstring encoding of unicode for internationalized domain names in applications (IDNA). RFC 3492.

CRAWFORD, M. 1999. Non-terminal DNS name redirection. RFC 2672.

DANZIG, P. B., OBRACZKA, K., AND KUMAR, A. 1992. An analysis of wide-area name server traffic: A study of the domain name system. In *Proceedings of the Annual Conference of the ACM SIGCOMM on Communications Architectures and Protocols (SIGCOMM'92)*, 281–292.

FALTSTROM, P., HOFFMAN, P., AND COSTELLO, A. M. 2003. Internationalizing domain names in applications (IDNA). RFC 3490.

FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H., AND BERNERS-LEE, T. 1997. Hypertext Transfer Protocol – HTTP/1.1. RFC 2068.

Hanyu da zidian Editorial Committee. 1986. Hanyu Da Zidan (Great dictionary of Chinese characters). Sichuan Cishuan Publishing, Chengdu, China. ISBN 780-543-001-2.

HOFFMAN, P. AND BLANCHET, M. 2002. Preparation of internationalized strings (stringprep). RFC 3454.

HOFFMAN, P. AND BLANCHET, M. 2003. Nameprep: A stringprep profile for internationalized domain names. RFC 3491.

IEEE Standard 1003.2-1992. IEEE standard for information technology – Portable Operating System Interface (POSIX) - Part 2: Shell and utilities, vol. 1.

IETF Internationalized Domain Names Working Group.
Retrieved from http://www.ietf.org/html.charters/idn-charter.html.

KLENSIN, J. 2004. A search-based access model for the DNS. Internet Draft.

KONISHI, K., HUANG, K., QIAN, H., AND KO, Y. 2004. Joint engineering team (JET) guidelines for internationalized domain names (IDN) registration and administration for Chinese, Japanese, and Korean. RFC 3743.

LAMPSON, B. W. 1985. Designing a global name service. In *Proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computing (PODC'85)*, 1–10.

LEE, X. D., HSU, N. W., CHEN, E., AND SUN, G. N. 2001. Traditional and simplified Chinese conversion. Internet Draft.

LIN, J. W., HO, J. M., TSENG, L. M., AND LAI, F. 2006. IDN server proxy architecture for internationalized domain name resolution and experiences with providing Web services. *ACM Trans. Internet Technol. 6*, 1.

MOCKAPETRIS, P. 1987. Domain names: Concepts and facilities (RFC 1034) and Domain names: Implementation and specification (RFC 1035). STD 13.

Secunia Stay Secure. 2005. Retrieved from
http://secunia.com/multiple_browsers_idn_spoofing_test/.

SENG, J., YONEY, A. Y., HUANG, K., AND KIM, K. 2001. Han ideograph (CJK) for internationalized domain names. Internet draft.

State Council of the People's Republic of China. 1986. A complete set of simplified Chinese characters.

The Unicode Consortium. 2002. Unihan database, version 3.2. Retrieved from ftp://ftp.unicode.org/Public/UNIDATA/Unihan.txt.

TSENG, L. M., HO, J. M., QIAN, H., AND HUANG, K. 2001. Internationalized domain names and unique identifiers/names. Internet draft.

TWNIC. 2005. IANA IDN Language table: TW Chinese character table.

ZHANG, Y., CHEN, T. ET AL. 1989. The KangXi dictionary. ISBN 962-231-006-0.

Received May 2005; revised December 2006, February 2008; accepted March 2008