A Web Redirection Service for Variant Chinese Domain Name Resolution

Jeng-Wei Lin^{1,3}, Li-Ming Tseng², Jan-Ming Ho¹, and Feipei Lai³

¹ Institute of Information Science,
Academia Sinica, Taipei, Taiwan
{jwlin, hoho}@iis.sinica.edu.tw

² Dept. of Computer Science and
Information Engineering,
National Central University,
Tao-yuan, Taiwan
tsenglm@cc.ncu.edu.tw

³ Dept. of Computer Science and
Information Engineering and Dept. of
Electrical Engineering, National
Taiwan University, Taipei, Taiwan
flai@ntu.edu.tw

Abstract

Many efforts in past years have been made to lower the linguistic barriers for non-native English speakers to access the Internet. IDNA [2] focuses on access to Internationalized Domain Names (IDN) 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. In JET Guidelines [6], it is suggested that zone administrators model this concept of equivalence as an atomic IDL package that contains the variant IDLs generated according to the Language Variant Tables (LVT). In addition to the registered IDL, some of the variant IDLs are activated and stored in the zone files and thus become resolvable. However, an issue of scalability arises when the number of the activated variant IDLs is large.

In this paper, we present a mechanism to resolve the variant IDLs in an IDL package into the registered IDL. Specifically, we target Han character variants. Two Han characters are said to be variants of each other if they have the same meaning and are pronounced the same. Furthermore, Han character variants usually have similar appearances. We introduce a new resource record (RR) type, denoted as *VarIdx*, to associate the variant IDLs with the registered IDL. Experiment results show that a small number of *VarIdx* RRs are sufficient for enumerating all variant IDLs in an IDL package. We then present a redirection service that uses *VarIdx* RRs to redirect user requests with variant IDLs to the URLs with the corresponding registered IDLs.

Keyword: Internationalized Domain Name, Han Character Variant, Localization

1. Introduction

The composition of traditional Internet domain names is restricted to ASCII letters, digits, and hyphens

[1]. The introduction of Internationalized Domain Name (IDN) allows people to use their native language to name and access Internet hosts. The IETF IDN workgroup [11] focused on the development of a standard to access IDNs. This work resulted in one primary protocol [2], known as IDNA, and three supporting protocols [3]-[5] that handle the encoding and decoding of Unicode [9][10] strings into and out of domain labels. However, the use of characters that have similar appearances and/or interpretations could create confusion. For example, Cyrillic A (U+0410¹) and ASCII A (U+0041) have the same appearance but are identified by different code points in computers, as do Cyrillic a (U+0430) and ASCII a (U+0061). Note that ASCII capital letters and small letters are considered as case-insensitive, e.g. A (U+0041)=a (U+0061), when they are used in a domain name [1].

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. For example, 清 (U+6E05) and 淸 (U+6DF8) are variants of each other, as are 真 (U+771F) and 眞 (U+771E). Chinese recognize 清真寺² (mosque) and 清真寺³ as equivalents. Clearly, this will cause confusion if the two terms achieve different results when they are used to access a network object. A variant IDL (Internationalized Domain Label), derived from an IDL by replacing some characters with their variants, should match the original IDL. If variant IDLs are registered by different name holders, the confusion may cause serious consumer protection problems. For example, tricksters may register 國稅局⁴ to cheat users who want to access 國稅局⁵ (National

¹ In this paper, the Unicode characters are identified by their code points. The notation U+12AB, for example, indicates the character at the position 12AB (hexadecimal) in the Unicode table.

² U+6E05 U+771F U+5BFA.

³ U+6DF8 U+771E U+5BFA.

⁴ U+570B U+7A0E U+5C40.

⁵ U+570B U+7A05 U+5C40.

Taxation Bureau). Cybersquatters may register 中研院⁶ to extort money from the name holder who owns 中研院⁷ (Academia Sinica).

Han character variant relationships are very complicated [14][18][19]; some are context-free, while others are context-sensitive. Some are used more frequently, and some are only found in ancient literature. In addition, variant relationships recognized in one region may not be recognized in other regions.

Due to region-specific interpretations of strings used in IDNs, the confusion may require different solutions. In [13], a multi-layer search model is proposed to address linguistic issues for the DNS. In JET Guidelines [6], a set of IDN guidelines is defined for applying restrictions to Chinese, Japanese and Korean (CJK) scripts and the zones that use these scripts. Each zone is allowed to define its own local rules for permitted characters and the handling of IDLs and their variants. The Language Variant Table (LVT) mechanisms are used to enforce language-based character variant preferences. When an IDL is registered, a collection of variant IDLs, known as an IDL package, is created according to the zone-specified LVTs. The IDL package is atomic that prevents consumer protection problems, such as cybersquatting.

In addition to the registered IDL, the name holder may ask the registry to activate some of the variant IDLs. The registry can use the DNAME [8] and CNAME [1] resource records (RR) to enumerate the activated variant IDLs in the zone files, as shown in Figure 2. Thus, the activated variant IDLs become resolvable. However, an issue of scalability arises when the number of the activated variant IDLs is large. In addition, some Internet services perform not only domain name lookups. For example, web and email services use a whole domain name to identify a network object and, therefore, service administrations have to enumerate the variant IDNs as aliases. The number of the variant IDNs is usually much larger than the number of the variant IDLs. For an IDN composed of three labels, each of which has three variant IDLs, the number of its variant IDNs is 64 (4^3). Such IDNs are not unusual when Han character variants are used. A mechanism to resolve the variant IDLs into the registered IDL is therefore needed.

This paper presents a mechanism to resolve the variant IDLs in an IDL package into its registered IDL. Specifically, we target Han character variants. A new RR type, denoted as VarIdx, is introduced to associate the variant IDLs with the registered IDL. To address the issue of scalability, OR-AND-like [12] variant

expressions are used to enumerate variant IDLs. An indexing function is designed to give the same value, denoted as the variant index, to all of the variant IDLs enumerated by a variant expression. When an IDL is registered, its variant IDLs are partitioned according to the indexing function so that the variant IDLs in each part are enumerated by a single variant expression and have the same variant index. For each part, a VarIdx RR, labeled as the variant index, is stored in the zone files to associate the variant expression with the registered IDL. Therefore, Internet applications can use any of the activated variant IDLs to look up the VarIdx RRs and thus the registered IDL.

We set up a web redirection server that uses VarIdx RRs to resolve the variant IDLs into the registered IDL. A "*" RR [1] is stored in the zone files so that user requests with unregistered labels in the corresponding domain will be sent to the redirection server. Upon receiving a user request, the redirection server tries to look up the VarIdx RRs. If the right VarIdx RR is found, the request can be redirected to the URL with the registered IDL, as shown in Figure 1. By cascading the redirection service in each domain, user requests with variant IDNs can be redirected to the URLs with the registered IDNs label by label.

Figure 1: A scenario that demonstrates how a request with a variant IDL is redirected to the URL with the registered IDL.

The remainder of this paper is organized as follows. In Section 2, we describe the VarIdx RR and indexing functions. In Section 3, we present the cascaded redirection service. Finally, in Section 4, we give our conclusions.

2. Resolution of Variant IDLs

2.1. VarIdx Resource Records

When an IDL is registered, an IDL package is created according to the zone-specific LVTs. The package consists of the registered IDL and its variant IDLs in which some character variants listed in the LVTs substitute for the original characters. For

⁶ U+4E20 U+784F U+9662.

⁷ U+4E20 U+7814 U+9662.

example, 台灣大學⁸ is a variant IDL of the IDL 臺灣大學⁹ (National Taiwan University), because 台 (U+53F0) is a variant of 臺 (U+81FA). The IDL package of the IDL 臺灣大學 can be described by the OR-AND-like variant expression, [臺台臺臺]灣灣[[大][學學]]¹⁰ according to LVT_{twcn} , the LVTs proposed in [16]. Since some (context-sensitive) variants, e.g. 檯 (U+6AAF), 簪 (U+7C49), and 颯 (U+98B1), are not proper variants of 臺 (U+81FA) in this context, only the variant IDLs enumerated by [臺台][灣灣][大][學學] are activated.

To address the issue of scalability, a new RR type, denoted as VarIdx, is introduced to associate the variant expression with the registered IDL, as shown in Figure 3. For the IDL 臺灣大學, a VarIdx RR that specifies "[臺台][灣灣][大][學學]→臺灣大學" is stored in the zone files.

2.2. Construction of Indexing Functions

For Internet applications to access VarIdx RRs, we want to design an indexing function $H()$ that gives the same value, denoted as the variant index, to all of the variant IDLs enumerated by a variant expression.

The idea is to assign the same value to character variants. If C_1 and C_2 are variants of each other, we assign $H(C_1)=H(C_2)$. For an IDL $L=X_1X_2...X_d$, we assign $H(L)=H(X_1)H(X_2)...H(X_d)$. If X_k ($1 \leq k \leq d$) and X' are variants of each other, by definition, $L'=X_1...X_{k-1}X'X_{k+1}...X_d$ is a variant IDL of L . It is easy to see that $H(L')$ equals $H(L)$. We construct the indexing function as follows.

Character variants are grouped together, according to the zone-specific LVTs, as shown in Figure 4. A character that does not have any variants forms a 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. After grouping all character variants, we assign a unique value for each group. For a n -member group $G=\{C_1, C_2, ..., C_n\}$, we assign $H(C_1)=H(C_2)=...=H(C_n)=MIN(C_1, C_2, ..., C_n)$.

For each zone, we can construct an indexing function such that it will give the same variant index to all variant IDLs in an IDL package registered in that zone. For example, $H_{twcn}()$, the index function constructed according to LVT_{twcn} , will give the same variant index to all variant IDLs in an IDL package registered in a zone that adopts LVT_{twcn} .

Figure 2: Enumeration of variant IDLs.

Figure 3: Enumeration of variant IDLs as a variant expression.

Figure 4: Grouping variants according to LVT_{twcn} .

However, individual zones may adopt different LVTs. For example, an IDN registry may recognize 芸 (U+82B8) as a variant of 藝 (U+85DD), and 發 (U+767A) as a variant of 發 (U+767C). Assume that a name holder registers an IDL 才藝發表會¹¹ and its activated variant IDLs are enumerated by $E=[才][藝藝藝芸][發發發發][表][會會]$ ¹². It is easy to see that $H_{twcn}()$ will not give the same variant index to all of these variant IDLs.

Instead of using a proprietary indexing function, we can partition the variant IDLs enumerated by E into four parts as follows: [才][藝藝藝][發發發][表][會會], [才][藝藝藝][發][表][會會], [才][芸][發發發][表][會會], and [才][芸][發][表][會會]. Consequently, $H_{twcn}()$ gives 才藝發表會, 才藝發表會, 才藝發表會, and 才藝發表會 respectively to the variant IDLs in each of the four

⁸ U+53F0 U+7063 U+5927 U+5B78.

⁹ U+81FA U+7063 U+5927 U+5B78.

¹⁰ [U+81FA U+53F0 U+6AAF U+7C49 U+98B1][U+7063 U+6E7E][U+5927][U+5B78 U+5B66].

¹¹ U+624D U+85DD U+767C U+8868 U+6703.

¹² [U+624D][U+85DD U+517F U+827A U+82B8][U+767C U+5F42 U+53D1 U+767A][U+8868][U+6703 U+4F1A].

parts. Note that the four variant indices can be enumerated by [才][藝][發][表][會].

Given an indexing function, $H()$, for a registered IDL $L = X_1^1 X_2^1 \dots X_d^1$ and its activated variant IDLs, A , which can be enumerated by the variant expression $E = [X_1^1 X_1^2 \dots X_1^{m(1)}][X_2^1 X_2^2 \dots X_2^{m(2)}] \dots [X_d^1 X_d^2 \dots X_d^{m(d)}]$, where $X_1^2, \dots, X_1^{m(i)}$ are proper variants of X_1^1 for $1 \leq i \leq d$, we can decompose E such that $E = \sum_{1 \leq j(i) \leq k(i)} [\Pi_i^{j(1)}][\Pi_i^{j(2)}] \dots [\Pi_i^{j(d)}]$, where $\Pi_i^1, \Pi_i^2, \dots, \Pi_i^{k(i)}$ is a partition of $\{X_i^1, X_i^2, \dots, X_i^{m(i)}\}$ according to $H()$ for $1 \leq i \leq d$; i.e.,

- 1) $\Pi_i^1 \cup \Pi_i^2 \cup \dots \cup \Pi_i^{k(i)} = \{X_i^1, X_i^2, \dots, X_i^{m(i)}\}$,
- 2) $\forall a \neq b, \Pi_i^a \cap \Pi_i^b = \emptyset$,
- 3) $\forall a, x \in \Pi_i^a, y \in \Pi_i^a, H(x) = H(y)$, and
- 4) $\forall a \neq b, x \in \Pi_i^a, y \in \Pi_i^b, H(x) \neq H(y)$.

In other words, we can partition A such that $H()$ will give the same variant index to all variant IDLs in each part enumerated by $[\Pi_i^{j(1)}][\Pi_i^{j(2)}] \dots [\Pi_i^{j(d)}]$, where $1 \leq j(i) \leq k(i)$ for $1 \leq i \leq d$. Note that the set of different variant indices $I = \text{Unique}(\{H(L) | L \in A\})$ is $\cup_{1 \leq j(i) \leq k(i)} \{H(x_1)H(x_2) \dots H(x_d) | x_i \in \Pi_i^{j(i)} \text{ for } 1 \leq i \leq d\}$, and the number of different variant indices $|I|$ is $k(1) * k(2) * \dots * k(d)$, where $\text{Unique}()$ is the function that returns the set of each unique member in the input set.

Since the indexing function may give two characters the same value, even if they are not variants of each other, the indexing function may give two IDLs the same variant index, even if they are not variant IDLs of each other. For example, $H()$ will give the following two IDLs 颱風¹³ (typhoon) and 臺風¹⁴ (stage manner) the same variant index. If both IDLs are registered, there will be two VarIdx RRs labeled as the same variant index. This phenomenon is referred to as a collision. Therefore, Internet applications must verify each of the returned VarIdx RRs to make sure the right VarIdx RR is used in the variant IDL resolution.

The pseudo code of variant IDL resolution is as follows:

```
Function ResolveVariantIDL (String L, String D)
{
    // L is the variant IDL
    // D is the domain
    RRs = DNS Lookup (H(L).D, VarIdx)
    foreach rr in (RRs) {
        if (L is in rr.variant_expression) {
            return rr.registered_IDL
        }
    }
    return Not_Registered
}
```

¹³ U+98B1 U+98A.

¹⁴ U+81FA U+98A8.

2.3. A Study of Indexing Functions

In this subsection, we present a preliminary study of two indexing functions. The first is $H_{twcn}()$; and the second, $H_{unihan}()$, is constructed according to the UniHan database [17]. In general, the variant relationships listed in LVT_{twcn} are recognized more specifically than those in the UniHan database. We consider an indexing function good when it meets the following requirements: 1) for an IDL, the number of VarIdx RRs required for enumerating its variant IDLs is small, and 2) there are few collisions of variant indices, even in a large domain, i.e., in which thousands or maybe more IDLs are registered.

We collected 21,374 IDLs registered in the ccTLD registry, TWNIC [22], which employs a registration policy that allows a registered IDL to associate with both Traditional Chinese and Simplified Chinese in the domain .tw. Table 1 shows some statistics of these IDLs. Although the average number of the variant IDLs that an IDL may have is small, the maximum number is large. As shown in Figure 5, the longer an IDL, the more variant IDLs it might have. We observe that 861 IDLs have more than 64 variant IDLs; and 28 IDLs have more than 512 variant IDLs. Although the ratio of these IDLs is small, they do have a large number of variant IDLs.

However, on average, we need to store only one VarIdx RR in the zone files to enumerate all variant IDLs when either $H_{twcn}()$ or $H_{unihan}()$ is used, as shown in Figure 5. The maximum numbers of required VarIdx RRs are also small, as shown in Table 2. The issue of scalability is successfully eliminated.

Table 2 also shows the maximum collisions pre variant index. At most two VarIdx RRs have the same variant index when either $H_{twcn}()$ or $H_{unihan}()$ is used. Note that the number of maximum collisions limits the foreach iterations in the procedure ResolveVariantIDL().

3. Cascaded Redirection Services

We built a trial IDN service of a three-level DNS hierarchy consisting of six domains. Complying with JET Guidelines, when an IDL is registered, an atomic IDL package is created according to LVT_{twcn} . However, we store only the registered IDL in the zone files. Figure 6 is a snapshot of the six domains. We use $H_{twcn}()$ to partition the activated variant IDLs. For each part, we store a VarIdx RR to associate the variant expression with the registered IDL. The VarIdx RR is simulated by using a TXT RR [1] as follows.

```
variant_index TXT VARIDX:\
variant_expression→registered_IDL
```

Table 1: Statistics of registered IDLs in TWNIC

Total IDLs	Avg. length	Avg. number of variant IDLs	Max. number of variant IDLs
21,374	6.15	11.1	2,048

Table 2: Experiment results.

	$H_{twcn}()$	$H_{unihan}()$
Maximum number of required VarIdx RRs per registered IDL	2	4
Maximum collisions per variant index	2	2

Figure 5: Number of required VarIdx RRs.

We set up a web redirection server that uses VarIdx RRs to resolve variant IDLs into the corresponding registered IDL. In each of the six domains, we store a "*" RR in the zone files so that the domain name servers will return the IP address of the redirection server when an unregistered label in that domain is queried. The redirection server runs an Apache HTTP server that listens on popular HTTP service ports and is configured to execute a redirection script when a user requests an object on a host whose name matches "*.idns".

When an unregistered label in the six domains is used to access a web object, the request will be sent to the redirection server. The redirection script examines the domain name in the request header and tries to resolve the domain name into the registered IDN. Since different browsers send the URL in different charset encodings, such as GB-2312/GBK, BIG5 or UTF-8, the script first tries to recognize the charset encoding by converting the URL from a candidate encoding to Unicode. The encodings to be tested, together with their priorities, are pre-assigned by the administrators of the redirection service. If the conversion succeeds, the script retrieves the unregistered IDL from the domain name, computes the variant index, and tries to look up VarIdx RRs. If the DNS returns a VarIdx RR that enumerates the IDL in the variant expression, the script finds the registered IDL. It then replaces the variant IDL with the registered IDL and redirects the request to the new URL.

If there are other unregistered labels in the new URL, the resolution described above will be invoked in a cascading manner in the corresponding subdomain until the redirection server finally returns the URL with the registered IDN, as shown in Figure 7, or the redirection server cannot resolve the unregistered label. Note that the response time of a request may be long if multiple redirections are involved. However, subsequent requests will use the registered IDN.

In some zone administrations, the URL can be redirected to more than two URLs because different charset encodings are used. For example, 公司.南京¹⁵ encoded in GB-2312 is the same as 鼠侗.歙儔¹⁶ encoded in BIG5. We believe that this situation can best be prevented 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.

3.1. Experiment Results

Under the domain *idns*, we registered five IDN subdomains forming a three-level DNS hierarchy. Under these subdomains, we registered 48 IDNs, which had a total of 1,915 variant IDNs. Table 3 shows the number of the variant IDLs and required VarIdx RRs. The issue of scalability is eliminated because only one VarIdx RR is required per registered IDL.

We used variant IDNs with different web browsers on different platforms to access the registered IDNs. Character variants that are either context-free or context-sensitive were used to form the variant IDNs. Experiment results show that the VarIdx RRs provide sufficient information for the redirection server to redirect user requests by resolving the variant IDNs into the registered IDNs.

4. Conclusions

Based on the LVT mechanisms, we present the resolution of the variant IDLs in an IDL package into its registered IDL with the help of VarIdx RRs. The complexity of Han character variant relationships is handled by atomic IDL packages. To address the issue of scalability that arises when the number of the variant IDLs 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. Therefore, Internet applications can use any of the variant IDLs to look up the VarIdx RRs and, thus, the registered IDL.

¹⁵ U+516C U+53F8. U+5357 U+4EAC.

¹⁶ U+9F20 U+4F97. U+9C0D U+5114.

Figure 6: A snapshot of the trial IDN service.

Figure 7: Cascaded Redirection Services

Table 3: Number of RRs registered in the zone files

Variant subdomains	Registered RRs	Variant IDLs	Registered RRs	Variant IDNs
52	5 VarIdx + 5 NS	195	48 VarIdx + 48 A	1,915

We started with a trial IDN service of six domains. Meanwhile, we set up a redirection server and stored a "*" RR in the zone files of each of the six domains so that user requests with unregistered labels in these domains will be served by the redirection server. Upon receiving a user request, the redirection server computes the variant index of the unregistered label, and tries to look up the VarIdx RRs. If there exists the right VarIdx RR, the server redirects the user request to the new URL by replacing the variant IDL with the registered IDL. Experiment results show that VarIdx RRs provide sufficient information for variant IDL resolution.

In our trial IDN service, we used the same LVT and, thus, the same indexing function in the six domains. However, individual domains may have their own rules about permitted characters and the variant relationships of these characters. The preliminary study of different indexing functions described in Section 2 indicates that an indexing function may exist for global use. In our

future research we will focus on the investigation of the global indexing function and other applications of VarIdx RRs, including IDN emails.

References

- [1] Paul Mockapetris, "Domain names - concepts and facilities" (RFC 1034) and "Domain names - implementation and specification" (RFC 1035), STD 13, November 1987.
- [2] Patrik Faltstrom, Paul Hoffman, Adam M. Costello, "Internationalizing Domain Names in Applications (IDNA)," RFC 3490, March 2003.
- [3] Adam M. Costello, "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)," RFC 3492, March 2003.
- [4] Paul Hoffman, and Marc Blanchet, "Preparation of Internationalized Strings (stringprep)," RFC 3454, December 2002.
- [5] Paul Hoffman, and Marc Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names," RFC 3491, March 2003.
- [6] Kazunori Konishi, Kenny Huang, Qian Hualin, Ko YangWoo, "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean," RFC 3743, April 2004.
- [7] Michael Mealling, Ron Daniel, "The Naming Authority Pointer (NAPTR) DNS Resource Record," RFC 2915, September 2000.
- [8] Matt Crawford, "Non-Terminal DNS Name Redirection," RFC 2672, August 1999.
- [9] ISO/IEC, "Information Technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane," ISO/IEC 10646-1:2000, October 2000.
- [10] The Unicode Consortium, "The Unicode Standard," <http://www.unicode.org/unicode/standard/standard.html>.
- [11] IETF Internationalized Domain Names Working Group, now concluded, <http://www.ietf.org/html.charters/idn-charter.html>.
- [12] IEEE, "IEEE Standard for Information Technology – Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)," IEEE Std 1003.2-1992, January 1993.
- [13] John Klensin, "A search-based access model for the DNS", Internet Draft, February 2004
- [14] Li Ming Tseng, Jan Ming Ho, Hua Lin Qian, Kenny Huang, "Internationalized Domain Names and Unique Identifiers/Names," Internet Draft, July, 2001.
- [15] XiaoDong Lee, Nai-Wen Hsu, Erin Chen, GuoNian Sun, "Traditional and Simplified Chinese Conversion," Internet Draft, June 2001.
- [16] XiaoDong Lee, Erin Chen, Li-Ming Tseng, Wei Mao, Nai-Wen Hsu, YanFeng Wang, "Registration and Administration Guideline for Chinese Domain Names," Internet Draft, August 2003.
- [17] The Unicode Consortium, "UniHan Database," <ftp://ftp.unicode.org/Public/UNIDATA/UniHan.txt>.
- [18] A Complete Set of Simplified Chinese Characters, published in 1986 by the Committee of National Language and Chinese Character of China.
- [19] Dictionary of Chinese Character Variants, compiled by Mandarin Promotion Council of Taiwan. Version 2 was published in Aug 2001 on Web site.<http://140.111.1.40/>.
- [20] Butler W. Lampson, "Designing a global name service," 4th ACM Symposium on Principles of Distributed Computing, August 1985.
- [21] Peter B. Danzig, Katia Obraczka, and Anant Kumar, "An analysis of wide-area name server traffic: A study of the domain name system," SIGCOMM 92, January 1992.
- [22] TWNIC, Taiwan Network Information Center. <http://www.twnic.net>.