Distr.
LIMITED
E/ESCWA/ICTD/2005/Technical Paper 2
15 July 2005
ORIGINAL: ENGLISH

ECONOMIC AND SOCIAL COMMISSION FOR WESTERN ASIA

DEVELOPMENT OF AN ARABIC DOMAIN NAMES SYSTEM

United Nations
New York, 2005

05-0440

Preface

The Arab region suffers from a digital divide that is largely manifested by low Internet usage rates and weak digital Arabic content. Moreover, given that language has been identified as one of the principal barriers to widespread Internet usage, there is a substantial market and a latent demand for using the Arabic language on the Internet.

The global undertaking to render the Internet more multilingual began in 1998. Within that context, the Economic and Social Commission for Western Asia (ESCWA) joined other national, regional and international entities in developing Internet domain names in Arabic. Currently and at a regional level, ESCWA, the League of Arab States (LAS) and all Arab countries are striving towards that goal.

Before 2003, most of the work in this area related only to the linguistic aspects of the intended Arabic Domain Names System (ADNS). Since then, ESCWA has maintained that an entire environment needs to be developed to satisfy the latent demand for Arabic content. The strategy adopted by ESCWA is based on a full-fledged approach that takes into account technology standardization, policy and administrative arrangements, and new applications.

In addition to linguistic aspects, this study raises the technical and operational aspects of ADNS for the first time. In so doing, this publication aspires to present a more complete description of the different aspects of ADNS with the aim of providing the basis for the full-fledged set of interoperable standards, and of presenting a substantive input to the regional pilot project launched in May 2005.

The ongoing efforts by ESCWA, LAS and all Arab countries with respect to developing ADNS are fully synchronized and are set to create the thrust for a broader Internet spread across the Arab region. While there remain some practical steps before wide-scale deployment, the significance of the regional achievements to-date cannot be underestimated. These achievements form part of a larger and noble goal, namely: the ongoing internationalization of the Internet, which is a cornerstone for local empowerment and Internet governance.

CONTENTS

LIST OF TABLES

LIST OF BOXES

CONTENTS (*continued*)

Page

LIST OF FIGURES

LIST OF ANNEXES

ABBREVIATIONS

ACE	ASCII Compatible Encoding
ADNS	Arabic Domain Name System
ADN-TF	Arabic Domain Names Task Force
AINC	Arab Internet Names Consortium
API	application program interface
BIND	Berkeley Internet Name Domain
CCS	coded character set
ccTLD	country-code top-level domain
CES	character encoding scheme
DNS	Domain Name System
FTP	File Transfer Protocol
gTLD	generic top-level domain
ICANN	Internet Corporation for Assigned Names and Numbers
ICT	information and communication technology
IDN	Internationalized Domain Name
IDNA	Internationalized Domain Names in Applications
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
JET	Joint Engineering Team
LAS	League of Arab States
MINC	Multilingual Internet Names Consortium
MLDN	Multilingual Domain Name
MUA	mail user agent
RFC	request for comments
SMTP	Simple Mail Transfer Protocol
STOP	Start-up Trademark Opposition Policy
TLD	top-level domain
TTL	Time to Live
UDRP	Uniform Domain-name Dispute-Resolution Policy
URL	Uniform Resource Locator
WG-ADN	Working Group on Arabic Domain Names
WIPO	World Intellectual Property Organization
WSIS	World Summit on the Information Society

I. INTRODUCTION

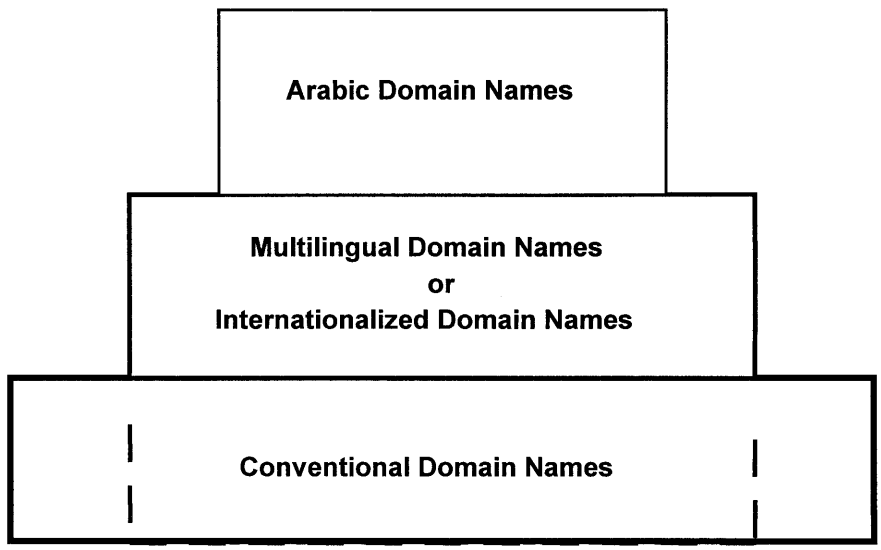
The Arabic Domain Names Task Force (ADN-TF), which was established by the Economic and Social Commission for Western Asia (ESCWA) in June 2003, published in 2004 the first Internet Draft, entitled *Guidelines for an Arabic Domain Name System*. That publication raised regional awareness with regard to ADNS issues, which were further discussed during the Second Regional Preparatory Conference for the World Summit on the Information Society (WSIS) that was organized by ESCWA in Damascus from 22 to 23 November 2004. Moreover, under the auspices of the League of Arab States (LAS), the newly established Working Group on Arabic Domain Names (WG-ADN) held its first meeting in Damascus, from 31 January to 2 February 2005, which resulted in recommendations resolving the remaining linguistic issues and adopting the work of ADN-TF. These recommendations are expected to be adopted and approved by the Council of Arab Ministers for Information and Communication Technology.

A. THE EVOLUTION OF INTERNATIONALIZED DOMAIN NAMES

The efforts exerted so far to define the Arabic Domain Names System (ADNS) were undertaken within the context of the global movement towards Internationalized Domain Names (IDNs) and Multilingual Domain Names (MLDNs), both of which were developed within the wider framework of the conventional Domain Names System (DNS). Figure 1 illustrates the evolution of ADNS within that global context.

In the conventional DNS, there are three types of players, namely: (a) organizations; (b) technology providers; and (c) service providers. Each of these three categories of players is responsible for a different set of goals and normally undertakes a differentiated set of activities.

Figure 1. The context of Arabic domain names



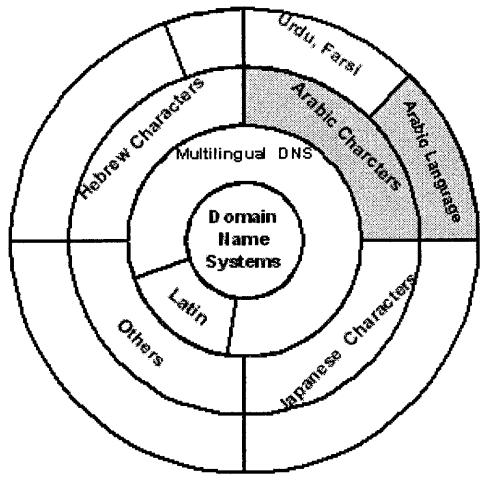
1. *Global evolution*

The evolution of MLDNs, which began in 1998, proved challenging. Given that the Internet Consortium for Assigned Names and Numbers (ICANN) was still in its developing phase, it was preoccupied with reorganization issues related to the development of the conventional Internet. It was this preoccupation that left room for uncoordinated efforts and that resulted in competing trial standards, particularly in the area of Arabic domain names, thereby creating a general state of uncertainty.

Within the context of MLDN, activities and efforts were initiated in Eastern Asia and, specifically, for the Chinese, Japanese and Korean languages. Given this lack of coordination, a plethora of MLDN technology providers, registries and registrars emerged in Eastern Asia before the Internationalized Domain Name (IDN) standards were defined.[1] Technologies differed among different MLDN providers mainly in terms of the manner in which the client-server relationship was used, in addition to differences in the character-set and the language script itself.

These IDN standards established regional and localized language-specific criteria for DNS. For example, Arabic, Farsi and Urdu share the same set of characters, whose handling therefore falls under the matching subset of multilingual DNS (see figure 2).

Figure 2. Domain name systems and languages



This document provides specific guidelines for ADNS. However, these guidelines can be used to facilitate systems in other languages that use the Arabic script, particularly Farsi and Urdu. Within that context, ADN-TF is prepared to cooperate with experts from the Farsi and Urdu communities in order to address linguistic, organizational and policy issues in an interoperable manner.

2. *Regional evolution*

During the period 1998-2003, implementations of Arabic domain names in the region varied significantly among technology providers and their respective registries. Those technology providers competed in order to impose standards upon the community, and to create a de facto standard that they could use to reinforce their position and gain recognition, thereby sustaining their innovation cycle. Moreover, in addition to competing in the area of technology, they interfered in trying to define linguistic aspects without any sufficient coordination or endorsement.

This situation led to chaos; and, consequently, standardization was not achieved. Established registries were technology-centric and adhered to "standards" and/or technologies that risked becoming obsolete in the short term, which in turn jeopardized the sustainability of the Domain Names of their end-users. Furthermore, the uniqueness of an Arabic domain name on the Internet was not guaranteed, which exacerbated the situation given that two entities or individuals could register the same name on two different

[1] In general, the Internationalized Domain Names (IDN) standards solve issues and conflicts relating to generic domain name access for scripts beyond the character set used in Latin languages, commonly referred to as the ASCII character set. See P. Faltstrom, P. Hoffman and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490 (Network Working Group, March 2003); P. Hoffman, and M. Blanchet, "Nameprep: a stringprep profile for Internationalized Domain Names (IDN)", RFC 3491 (Network Working Group, March 2003); and A. Costello, "Punycode: a bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492 (Network Working Group, March 2003).

registries. On the other hand, several registries refused to implement any solution before the formal adoption by an independent authority. Consequently, most technology providers that were unable to secure enough clients went out of business.

The Arab Internet Names Consortium (AINC), which was established in 2001, attempted to assume the role of coordinating body. Unfortunately, the Consortium suffered from internal conflicts, which impeded its activities and resulted in its suspension. Clearly, the absence of a strong regional coordinating body prevented development in this area.

In 2003, the situation in the Arab region could be summarized as follows:

(a) Professionals and consumers lacked awareness of the viability and importance of Arabic domain names in general;

(b) Time and effort was wasted on competing technologies and standards, thereby draining the resources of emerging ADNS companies;

(c) The absence of a coordinating body reduced overall effectiveness and hampered efforts to move towards a regulated environment.

However, the period 1998-2003 resulted in an accumulation of experience among the involved players, which became an asset that facilitated the next phase of the ADNS evolution.

B. The revitalization of Arabic domain names by ESCWA

In March 2003, the Internet Engineering Task Force (IETF) issued a set of requests for comments (RFCs) for Internationalized Domain Names that are intended to become the basis for domain name standards for all languages.[2] These IDN standards are set to resolve the outstanding conflicts on domain names. New and emerging technology providers will no longer need to compete on the basic standards, rather on efficiency levels and the technology costs. All registries and registrars are set to become compatible and, most importantly, the domain names themselves will be unique.

Seeking to revitalize Arab regional efforts in this area, ESCWA organized the Expert Group Meeting on Promotion of Digital Arabic Content (Beirut, 3-5 June 2003), which established a new roadmap for developing the Arabic domain name industry and discussed activities required to establish consensus on ADNS. Given the potential and impact of ADNS, this Meeting focused on identifying obstacles and setting objectives and initiatives for the promotion of that System in a coordinated fashion.[3]

Upon the recommendations of the participants to that Meeting, ADN-TF was formed under the auspices of ESCWA, which equally acts as its secretariat, and aimed at the following: (a) raising awareness among stakeholders concerning the importance of ADNS; (b) defining standards for ADNS through RFCs; (c) promoting the adoption of standards in a coordinated fashion; (d) obtaining global recognition for these adopted standards; and (e) facilitating the deployment of these standards by the various stakeholders.

In the two years since its establishment, ADN-TF has sought to achieve the first three of these objectives by developing the first Internet Draft on ADNS, entitled *Guidelines for an Arabic Domain Names*

---

[2] Ibid.

[3] Almost concurrent to that Expert Group Meeting, the Multilingual Internet Names Consortium (MINC) announced in London, 11 June 2003, its policies on the linguistic and cultural relevance for IDN development. More information is available at: www.minc.org.

System. This publication, which was published in June 2004 on the IETF website, became the first global Internet Draft on the subject.[4]

Moreover, WG-ADN, which was established in July 2004 by LAS, seeks to decide and agree on the various issues related to establishing an ADNS, which include the topics encompassed within the Internet Draft by ADN-TF. It was during the first meeting of the Working Group (Damascus, 31 January - 2 February 2005) that the Internet Draft was discussed and linguistic aspects were adopted after the resolving of the pending issues. Subsequently, the Draft was amended by ESCWA and finally presented to the second meeting of the Working Group (cairo, 7-9 May 2005). This second meeting also triggered the regional pilot project for the proposed ADNS. Additionally, WG-ADN established steering and technical committees aimed at studying and discussing relevant technical issues that are presented in detail in the following chapters of this publication. This study aims therefore to provide valuable input to WG-ADN for discussion, feedback and enrichment.

This document provides a thorough analysis of the three major aspects of ADNS, namely: linguistic, technical and operational, which are covered in chapters II, III and IV, respectively. While the content of the linguistic aspect has already been extensively studied, discussed and debated, leading to endorsement by WG-ADN, the technical and operational aspects are tackled for the first time and are hereby presented with the aim of seeking endorsement by WG-ADN.

This analysis of the technical and operational aspects take into consideration the full compatibility with both the IDN standards of 2003 and the Arab linguistic guidelines agreed upon by AWG-ADN in 2005.

Specifically, chapter III reviews the extended technological aspects that have been pending since the first Internet Draft, including technical considerations for Arabic top-level domain (TLD) mapping; integrating Arabic domain names with such Internet services as File Transfer Protocol (FTP) and e-mail; implementation and support of iClient, with relevant discussions and suggestions for alternative or complementary systems. Additionally, future technical considerations are highlighted in this chapter.

Chapter IV discusses and presents a thorough analysis of the operational aspects of ADNS as these relate to building relationships between registrant, registrar and registry; the essential needs for setting up new domains that support ADNS, particularly generic top-level domains (gTLDs); the challenges that can face the registry and the registrar during the creation of new domains; and the impact of IDN on the tasks of the registrars.

Chapter V discusses the reliability of DNS, particularly given that it represents the cornerstone of the root server aimed at supporting ADNS. The reliability requirements of this System are discussed, along with provisions for ensuring continuous operation following such concomitant risks as malicious attacks.

[4] In April 2004, the Joint Engineering Team (JET) produced RFC 3743 on IDN registration and administration for Chinese, Japanese and Korean. See K. Konishi et al., "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) registration and administration for Chinese, Japanese, and Korean", RFC 3743 (Network Working Group, April 2004).

II. THE LINGUISTIC ASPECTS OF ADNS[5]

Principally, ADNS is aimed at increasing Internet use among all strata of the Arabic-speaking communities. This intended wide-scale dissemination of the Internet could be hampered if the structure or hierarchy of ADNS does not meet certain core criteria. Specifically, establishing a system that is not user-friendly could add to the ambiguity and the eccentricity of the Internet to the Arabic-speaking communities, thereby contributing negatively to the spread of the Internet and leading to further isolation of these communities at the global level.

Consequently, there have been intensive efforts to reach consensus on a multitude of linguistic issues with the following goals: (a) to define the Arabic character set to be used in spelling domain names in Arabic; and (b) to define the top-level domains of the Arabic domain name tree structure, including Arabic generic top-level domains (gTLDs) and country-code top-level domains (ccTLDs).

Within that context, there are many valid criteria to evaluate the proposed Arabic gTLDs or the Arabic ccTLDs, namely:[6]

(a) Length of a given gTLD or ccTLD;

(b) Coherence and clarity;

(c) Consistency with the Arabic language;

(d) Ease of pronunciation;

(e) Extendibility;

(f) The domain name as a whole, written with dots between words, to be easily inferred by being as close as possible to its original name;

(g) The domain name as a whole to be suitable to native Arabic-speakers and, consequently, user friendly.

The last two items are necessary in order to achieve wide-scale dissemination. They are of utmost importance in the deployment and adoption of ADNS.

Furthermore, consensus was reached by WG-ADN to stress the following aims:

(a) Simplifying domain names, whenever possible, thereby facilitating interaction of the Arabic user with the Internet;

(b) Adopting solutions that do not lead to confusion in terms of both reading and writing, provided that this does not compromise the linguistic correctness of used words;

(c) Objecting to the combination of Arabic with non-Arabic letters in domain names.

A. LINGUISTIC ISSUES

A number of linguistic issues were agreed upon by ADN-TF and WG-ADN. These are summarized below.

1. *"Tashkeel" and "shadda"*[7]

In the start-up phase of ADNS, both "tashkeel" and "*shadda*" should not be supported in the zone file. However, these diacritics can be supported in the user interface and removed when preparing

---

[5] This chapter is largely based on the Internet Draft, entitled *Guidelines for an Arabic Domain Names System*, and on the report of the first meeting of WG-ADN.

[6] See A. al-Zoman, "Supporting the Arabic language in domain names" (King Abdulaziz City for Science and Technology, October 2003); and A. Al-Zoman, "Top-level Arabic Domains" (in Arabic), which was presented at the Expert Group Meeting on Promotion of Digital Arabic Content (Beirut, 3-5 June 2003).

[7] "Tashkeel" and "shadda" are diacritics.

internationalized strings, which is commonly referred to as the "stringprep" phase. This guideline concerning the use of "*tashkeel*" and "*shadda*" can be reviewed subsequent to adequate research and relevant field studies.

2. *"Kasheeda" or "tatweel"*[8]

The practice of "*kasheeda*" or "*tatweel*" should not be used in Arabic domain names.

3. *Character folding*

Character folding is the process in Arabic whereby multiple letters are folded into one shape. This includes the following: (a) folding "*teh marbuta*" and "*heh*" at the end of a word; (b) folding different forms of "*hamzah*"; (c) folding "*alef maksura*" and "*yeh*" at the end of a word; and (d) folding "*waw*" with "*hamzah*" and "*waw*".

There was a consensus that character folding was unacceptable given that it changes the meaning of the words and it is against the simplest spelling rules. Replacing a character with another character, which can have the same shape but different pronunciation, yields a diverse meaning.[9]

B. SUPPORTED CHARACTER SET

Table 1 shows the recommended Unicode for each Arabic character.[10]

TABLE 1. UNICODE FOR ARABIC CHARACTERS

Unicode	**Arabic character**
0621	*hamza* (ء)
0622	*alef* with *madda* above (آ)
0623	*alef* with *hamza* above (أ)
0624	*waw* with *hamza* above (ؤ)
0625	*alef* with *hamza* below (إ)
0626	*yeh* with *hamza* above (ئ)
0627	*alef* (ا)
0628	*beh* (ب)
0629	*teh marbuta* (ة)
062A	*the* (ت)
062B	*theh* (ث)
062C	*jeem* (ج)
062D	*hah* (ح)
062E	*khah* (خ)
062F	*dal* (د)
0630	*thal* (ذ)
0631	*reh* (ر)
0632	*zain* (ز)

[8] "Kasheeda", which is commonly known as "tatweel", is the practice in Arabic of extending the size of a character horizontally.

[9] A. al-Zoman, "Supporting the Arabic language in domain names" (King Abdulaziz City for Science and Technology, October 2003).

[10] These are based on the study and the report from the linguistic committee of the Arab Internet Names Consortium (AINC), rooted in Unicode version 3.1.

TABLE 1 (*continued*)

Unicode	Arabic character
0633	*seen* (س)
0634	*sheen* (ش)
0635	*sad* (ص)
0636	*dad* (ض)
0637	*tah* (ط)
0638	*zah* (ظ)
0639	*ain* (ع)
063A	*ghain* (غ)
0641	*feh* (ف)
0642	*qaf* (ق)
0643	*kaf* (ك)
0644	*lam* (ل)
0645	*meem* (م)
0646	*noon* (ن)
0647	*heh* (ه)
0648	*waw* (و)
0649	*alef maksura* (ى)
064A	*Yeh* (ي)
0660	Arabic digit zero (٠)
0661	Arabic digit one (١)
0662	Arabic digit two (٢)
0663	Arabic digit three (٣)
0664	Arabic digit four (٤)
0665	Arabic digit five (٥)
0666	Arabic digit six (٦)
0667	Arabic digit seven (٧)
0668	Arabic digit eight (٨)
0669	Arabic digit nine (٩)

Source: A. al-Zoman, "Supporting the Arabic language in domain names" (King Abdulaziz City for Science and Technology, October 2003).

TABLE 2. UNICODE FOR OTHER CHARACTERS USED IN ARABIC

Unicode	Character
0030	Zero (0)
0031	One (1)
0032	Two (2)
0033	Three (3)
0034	Four (4)
0035	Five (5)
0036	Six (6)
0037	Seven (7)
0038	Eight (8)
0039	Nine (9)
002D	Hyphen-minus (-)
002E	Full stop (.)

Source: A. al-Zoman, "Supporting the Arabic language in domain names" (King Abdulaziz City for Science and Technology, October 2003).

C. RECOMMENDED ARABIC gTLDs AND ccTLDs

Within the context of Arabic gTLDs, there was a consensus to use only such geographical descriptive words as "international" (دولي) and "arabic" (عربي) during the start-up phase. This set may subsequently be expanded to refer to activities, including educational and commercial in the future.[11]

In the area of ccTLDs, previous efforts paved the way for establishing and implementing country-specific Arabic domain names. Specifically, two alternative systems for coding country names were identified and discussed, namely:[12] one based on a single-word representation of the country name; and one based on a two-character coded abbreviation table.[13] Within the context of the former, the single-word option involves the use or lack thereof, depending on the country, of the Arabic noun identification letter.[14]

While the two-character names represent a higher degree of practicality, some of the resulting abbreviations carry inappropriate meanings. Moreover, the single-word option can be used within advertising material for clearer name representation. Consequently, WG-ADN recommended the adoption of the single-word representation for ccTLDs based on the specification No. 642-1985 by the Arab Standardization Organization, and not the symbolic two-character coded abbreviation.[15]

Specifically, the standard name of Arabic countries is to be used except when there is more than one word in a given country name, in which case only one indicative word should be adopted. Examples include "*leebya*", shortened from "*algamahiryia alarabyia leebya*" for the Libyan Arab Jamahiriya; and "*alimaarat*" instead of "*alimaarat alarabyia almotahhida*" for the United Arab Emirates. Table 3 illustrates the recommended ccTLD codes for Arab countries in the single-word format.

TABLE 3. THE RECOMMENDED CCTLD CODES FOR ARAB COUNTRIES

Country or territory	**Recommended ccTLD code**
Algeria	u+0627 u+0644 u+062C u+0632 u+0627 u+0626 u+0631 (الجزائر)
Bahrain	u+0627 u+0644 u+0628 u+062D u+0631 u+064A u+0646 (البحرين)
Comoros	u+0627 u+0644 u+0642 u+0645 u+0631 (القمر)
Djibouti	u+062C u+064A u+0628 u+0648 u+062A u+064A (جيبوتي)
Egypt	u+0645 u+0635 u+0631 (مصر)

[11] In Unicode, these Arabic gTLDs are given as "u+062F u+0648 u+0644 u+064A" and "u+0639 u+0631 u+0628 u+064A", respectively.

[12] W. Nasr, "TLD mapping and standardization for Arabic domain names" (i-DNS.net, December 2003).

[13] Arab Standardization and Metrology Organization (ASMO), "Arab Standard Specifications, No. 642-1985: codes for names of countries and languages" (ASMO, 1985).

[14] The Arabic noun identification letter (ال) is known as "*al-altareef*", which in Unicode is represented by "u+0627 u+0644".

[15] This is based on Arab Standardization and Metrology Organization (ASMO), "Arab Standard Specifications, No. 642-1985: codes for names of countries and languages" (ASMO, 1985); and League of Arab States (LAS), "Report of the first meeting of the Working Group on Arabic Domain Names" (LAS, 2005).

Table 3 (*continued*)

Country or territory	Recommended ccTLD code
Iraq	u+0627 u+0644 u+0639 u+0631 u+0627 u+0642 (العراق)
Jordan	u+0627 u+0644 u+0623 u+0631 u+062F u+0646 (الأردن)
Kuwait	u+0627 u+0644 u+0643 u+0648 u+064A u+062A (الكويت)
Lebanon	u+0644 u+0628 u+0646 u+0627 u+0646 (لبنان)
Libyan Arab Jamahiriya	u+0644 u+064A u+0628 u+064A u+0627 (ليبيا)
Mauritania	u+0645 u+0648 u+0631 u+064A u+062A u+0627 u+0646 u+064A u+0627 (موريتانيا)
Morocco	u+0627 u+0644 u+0645 u+063A u+0631 u+0628 (المغرب)
Oman	u+0639 u+0645 u+0627 u+0646 (عُمان)
Palestine	u+0641 u+0644 u+0633 u+0637 u+064A u+0646 (فلسطين)
Qatar	u+0642 u+0637 u+0631 (قطر)
Saudi Arabia	u+0627 u+0644 u+0633 u+0639 u+0648 u+062F u+064A u+0629 (السعودية)
Somalia	u+0627 u+0644 u+0635 u+0648 u+0645 u+0627 u+0644 (الصومال)
Sudan	u+0627 u+0644 u+0633 u+0648 u+062F u+0627 u+0646 (السودان)
Syrian Arab Republic	u+0633 u+0648 u+0631 u+064A u+0629 (سورية)

Table 3 (*continued*)

Country or territory	Recommended ccTLD code
Tunisia	u+062A u+0648 u+0646 u+0633 (تونس)
United Arab Emirates	u+0627 u+0644 u+0625 u+0645 u+0627 u+0631 u+0627 u+062A (الإمارات)
Yemen	u+0627 u+0644 u+064A u+0645 u+0646 (اليمن)

Source: Compiled by ESCWA, based on League of Arab States (LAS), "Report of the first meeting of the Working Group on Arabic Domain Names" (LAS, 2005).

D. ARABIC DOMAIN NAME STRUCTURE

A domain name consists of multiple words (codes) that are separated by dots (u+002E). After considering and weighing a multitude of alternatives and combinations, a structure was recommended for an Arabic domain name based on an adopted geographical classification and, moreover, on a deliberate lack of activity classifications, including, for example, ".com" and ".org". The proposed structure has the following syntax to be read from right to left: <A-TLD>.<entity-name>

In such a structure, <entity-name> represents the Arabic name of a particular entity and <A-TLD> represents an Arabic TLD. Unicode values in hexadecimal are written below from left to right and correlate to the original Arabic characters, which are typed from right to left. Consequently, one of the features of this structure is the need to switch the order of reading and writing the category identifier to be at the beginning and a part of the name. The rationale behind the sequence is that, in Arabic, it is considered more correct to use the <com-company name> structure, rather than the <company name>.<com> structure. Box 1 illustrates some examples of Arabic domain names according to the proposed structure.

Box 1. Examples of Arabic domain names with their Unicode values

Example 1

شركة-زومان. السعودية

u+0634 u+0631 u+0643 u+0629 u+02D u+0632 u+0648 u+0645 u+0627 u+0646 u+002E u+0627 u+0644 u+0633 u+0639 u+0648 u+062F u+064A u+0629

Example 2

شركة-أرامكو. السعودية

u+0634 u+0631 u+0643 u+0629 u+02D u+0623 u+0631 u+0627 u+0645 u+0643 u+0648 u+002E u+0627 u+0644 u+0633 u+0639 u+0648 u+062F u+064A u+0629

Example 3

المركز-التجاري. سورية

u+0627 u+0644 u+0645 u+0631 u+0643 u+0632 u+02D u+0627 u+0644 u+062A u+062C u+0627 u+0631 u+064A u+002E u+0633 u+0648 u+0631 u+064A u+0629

Example 4

اتحاد-كرة-الطائرة. عربي

u+0627 u+062A u+062D u+0627 u+062F u+02D u+0643 u+0631 u+0629 u+02D u+0627 u+0644 u+0637 u+0627 u+0626 u+0631 u+0629 u+002E u+0639 u+0631 u+0628 u+064A

Example 5

جامعة-الخرطوم. السودان

u+062C u+0627 u+0645 u+0639 u+0629 u+02D u+0627 u+0644 u+062E u+0631 u+0637 u+0648 u+0645 u+002E 0627 u+0644 u+0633 u+0648 u+062F u+0627 u+0646

For such gTLDs as "international" (دولـــي) and "arabic" (عربي), it is proposed that a single regional body acts as the authority for the standardization and governance of Arabic domain names. This body needs to appoint either a single company to run a unique registry of Arabic domain names or multiple companies that can work together in coordination. At a national level, ccTLDs could be managed independently in each country by a national network administrator; and a single company could be appointed to manage gTLDS.

E. ARABIC LINGUISTIC ISSUES AFFECTED BY TECHNICAL CONSTRAINTS

1. *Numerals*

Two sets of numerals are used in the Arab region, namely: (a) set I (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), which is typically used in the western part of the Arab region; and (b) set II: (٠ ،١ ،٢ ،٣ ،٤ ،٥ ،٦ ،٧ ،٨ ،٩) which tends to be used in the eastern part of the Arab region.[16]

While visual differentiation between the Arabic zero (٠) and the dot (.) in printed material is possible given that the zero is larger and is printed higher than the dot, the differentiation is less immediate in a domain name and can lead to confusion. WG-ADN decided that folding set II to set I eliminates the problem of the zero and unifies the representation of numerals in general. It was also decided that both sets can be supported in the user interface and folded into set I when preparing internationalized strings at the "stringprep" phase.[17]

2. *The space character*

The space character is strictly disallowed in domain names, given that it is a control character in the IDN standard. Instead, the hyphen (-) is proposed as a separator between Arabic words to avoid confusion. While hyphens can be used to separate words within a given domain name label, it is still recommended to find technical solutions that can enable the use of the space character.[18]

3. *TLD mapping and ccTLD*

A TLD mapping process allows a set of valid substitutions to be made, including those conventions previously used by older implementations. This TLD mapping process, described in detail in chapter III, facilitates migration paths for existing systems. Within that framework, users are able to type domain names in one of the following formats: (a) long ccTLD name with or without the Arabic noun identification letter (ال); and (b) short two-letter coded abbreviation, thereby supporting older conventions. Table 4 lists the valid TLD mappings to accommodate the two-letter code abbreviation and/or the inclusion of the Arabic noun identification letter.

TABLE 4. ARABIC TLD MAPPING ALTERNATIVES

Country or territory	**Country code**		
	Default with the Arabic noun identification letter (where applicable)	**Alternative 1: two-letter abbreviation**	**Alternative 2: without the Arabic noun identification letter (where applicable)**
Algeria	(الجزائر)	u+062C u+0632 (جز)	u+062C u+0632 u+0627 u+0626 u+0631 (جزائر)

[16] Set II is represented by the following Unicode values: u+0660, u+0661, u+0662, u+0663, u+0664, u+0665, u+0666, u+0667, u+0668, u+0669.

[17] In other words, the storage of numerals in the zone file is carried out in ASCII format. See League of Arab States (LAS), "Report of the first meeting of the Working Group on Arabic Domain Names" (LAS, 2005).

[18] Ibid.

TABLE 4 (*continued*)

Country or territory		Country code	
Bahrain	(البحرين)	u+0628 u+062D (بح)	u+0628 u+062D u+0631 u+064A u+0646 (بحرين)
Comoros	(القمر)	u+0642 u+0645 (قم)	Not applicable
Djibouti	(جيبوتي)	u+062C u+064A (جي)	Not applicable
Egypt	(مصر)	u+0645 u+0635 (مص)	Not applicable
Iraq	(العراق)	u+0639 u+0631 (عر)	u+0639 u+0631 u+0627 u+0642 (عراق)
Jordan	(الأردن)	u+0627 u+0631 (ار)	u+0623 u+0631 u+062F u+0646 (أردن)
Kuwait	(الكويت)	u+0643 u+0648 (كو)	u+0643 u+0648 u+064A u+062A (كويت)
Lebanon	(لبنان)	u+0644 u+0628 (لب)	Not applicable
Libyan Arab Jamahiriya	(ليبيا)	u+0644 u+064A (لي)	Not applicable
Mauritania	(موريتانيا)	u+0645 u+0648 (مو)	Not applicable
Morocco	(المغرب)	u+0645 u+063A (مغ)	u+0645 u+063A u+0631 u+0628 (مغرب)
Oman	(عُمان)	u+0639 u+0645 (عم)	Not applicable
Palestine	(فلسطين)	u+0641 u+0644 (فل)	Not applicable
Qatar	(قطر)	u+0642 u+0637 (قط)	Not applicable
Saudi Arabia	(السعودية)	u+0633 u+0639 (سع)	u+0633 u+0639 u+0648 u+062F u+064A u+0629 (سعودية)
Somalia	(الصومال)	u+0635 u+0648 (صو)	u+0635 u+0648 u+0645 u+0627 u+0644 (صومال)
Sudan	(السودان)	u+0633 u+062F (سد)	u+0633 u+0648 u+062F u+0627 u+0646 (سودان)
Syrian Arab Republic	(سورية)	u+0633 u+0631 (سر)	Not applicable

TABLE 4 (*continued*)

Country or territory		Country code	
Tunisia	(تونس)	u+062A u+0648 (تو)	Not applicable
United Arab Emirates	(الإمارات)	u+0627 u+0645 (ام)	u+0625 u+0645 u+0627 u+0631 u+0627 u+062A (إمارات)
Yemen	(اليمن)	u+064A u+0645 (يم)	u+064A u+0645 u+0646 (يمن)

Source: Compiled by ESCWA.

III. THE TECHNICAL ASPECTS OF ADNS

A. DNS-BASED SOLUTION

Historically, there have been different approaches towards resolving conflicts in MLDNs. Within that context, solutions have tended to be grouped within two categories, namely: DNS solutions and keyword solutions.

In addition to being compliant to IETF, DNS-based solutions preserve the language integrity and allow hyperlinks. On the other hand, "keywords" are not domain names. Rather, they exist as an additional layer above DNS. Consequently, while DNS-based solutions only require the use of the DNS resolution infrastructure of the Internet, keyword-based solutions require a Uniform Resource Locator (URL) forwarding technique to map simple references, names and phrases to domain names or Internet Protocol (IP) addresses.

As a prerequisite to using keywords, each resolvable domain name is registered in a keyword-based directory in addition to the DNS registry. During a Domain Name Resolution process, the keyword directory is looked up; and matches in the keyword registry are used to locate a particular URL or a list of matching sites under that particular keyword. Keyword approaches cannot replace a DNS-based structure; they are viable only as a supplemental scheme over and above a robust DNS-based solution.

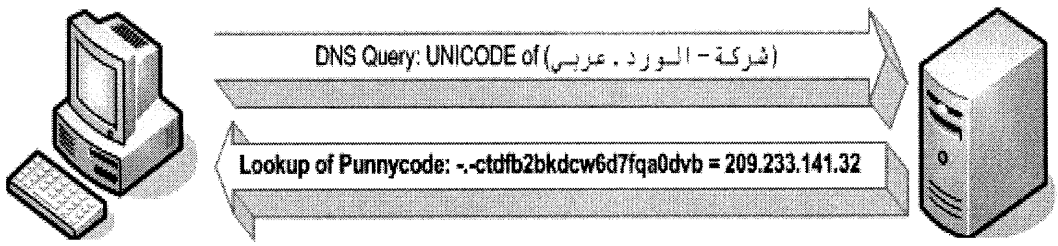
Consequently, it is recommended to use a DNS-based solution to preserve the integrity of the Arabic language, thereby eliminating any confusion and to develop a fully interoperable system with existing DNS schemes.[19]

B. A CLIENT-BASED VERSUS A SERVER-BASED APPROACH

In general, there are two schemes for resolving MLDNs with regard to DNS-based solutions. These are "server-based" and "client-based" schemes.

The proposed architecture for ADNS is in accordance with the IDN standards, which recommend that a client-side resolution scheme accommodate languages that do not use a Latin script, including Arabic. Figure 3 presents both schemes in a layer above the current Internet structure.

Figure 3. Different approaches for the resolution of conflicts in Multilingual Domain Names

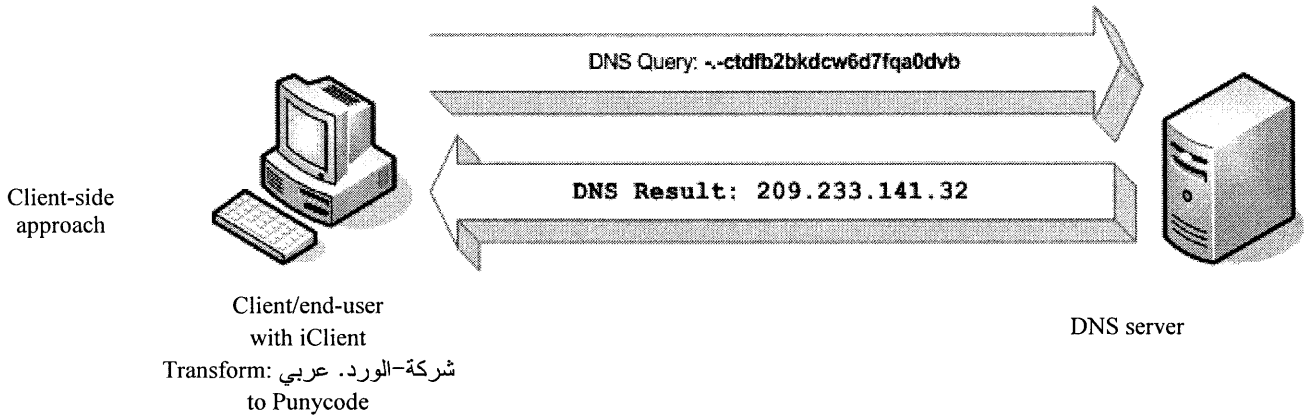


Server-side approach

Client/end-user

iDNS compatible server:
Converts queried strings to Punycode and performs DNS lookup

[19] RFC 3743 from the JET adopts a similar solution for Chinese, Japanese, and Korean languages. See K. Konishi et al., "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) registration and administration for Chinese, Japanese, and Korean", RFC 3743 (Network Working Group, April 2004).



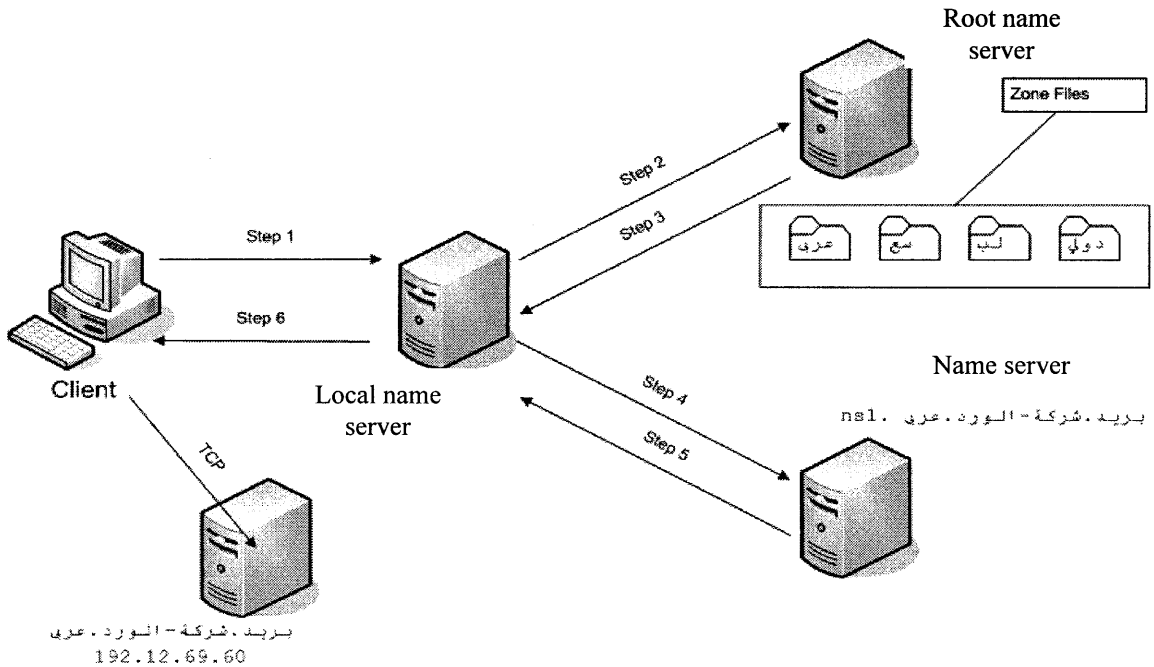
Source: ESCWA.

C. NETWORK STRUCTURE AND RELATED COMPONENTS

The proposed architecture for ADNS is based on a client-side resolution scheme. On the client side, workstations will be running some DNS resolution agent service at the system level. When a given local agent receives a DNS resolution request from upper-level applications, it will assume the duty to communicate with DNS servers configured for the particular workstation. Subsequently, once that agent receives responses from the DNS server, it relays the results to the upper-level applications.

In terms of IDN resolution, a software client intercepts the resolution request before it reaches the local resolution agent, and replaces the multilingual query with ASCII Compatible Encoding (ACE) formatted value, known in this case as Punycode. Consequently, local resolution agents simply follow the normal DNS resolution process as they do for ASCII formatted queries. Figure 4 provides an example of such a resolution process.[20]

Figure 4. Example of a resolution process of a domain name



[20] In this example, the domain name is بريـــد.شـــركة-الـــورد.عربـــي; and has the following Unicode values: u+0628 u+0631 u+064A u+062F u+002E u+0634 u+0631 u+0643 u+0629 u+02D u+0627 u+0644 u+0648 u+0631 u+062F u+002E u+0639 u+0631 u+0628 u+064A.

Figure 4 *(continued)*

Step 1: The client converts the domain name to Punycode and sends a query containing the domain name to the local name server.

Step 2: The local name server, which may not have the information concerning the domain name, sends the query to one of the root servers. In this example, there is a dedicated root server for gTLD "عربي"

Step 3: The root server cannot match the entire name, so it returns the best match, namely, the name resolution record (ns1) for شركة-الورد.عربـــي (u+0634 u+0631 u+0643 u+0629 u+02D u+0627 u+0644 u+0648 u+0631 u+062F u+002E u+0639 u+0631 u+0628 u+0649). It also returns all records that are related to this record.

Step 4: The local name server sends the same query to the authoritative name server for the mail zone ns1. بريد.شركة-الورد.عربـــي (u+06E u+073 u+0031 u+002E u+0628 u+0631 u+064A u+062F u+002E u+0634 u+0631 u+0643 u+0629 u+02D u+0627 u+0644 u+0648 u+0631 u+062F u+002E u+0639 u+0631 u+0628 u+0649).

Step 5: The server has information about the domain and returns the answer: IP address = 192.12.69.60.

Step 6: The local name server then responds to the client with the IP value. The client can then establish connection with the desired destination.

Source: ESCWA.

D. TECHNICAL CONSIDERATIONS FOR ARABIC TLD MAPPING

There are several considerations concerning the selection and subsequent deployment of Arabic TLDs, including both Arabic gTLDs and ccTLDs. These are the length, coherence/clarity, consistency, ease of pronunciation, extendibility, and whether the name can be deduced and is user-friendly.[21]

While, as discussed in chapter II, the default ccTLD is stored as a unique entry in the zone file,[22] there are cases where it is required to support the use of alternatives in order to promote flexibility or to support a status-quo.

Moreover, while it is technically possible to deploy an Arabic DNS by assigning an individual Punycode TLD for each of these representations,[23] such a deployment is not operationally feasible given that this leads to the creation of separate and distinct DNS zones, thereby resulting in administrative difficulties in terms of matching the zone data with the mapped TLD representations. Furthermore, these administrative difficulties are compounded given the need to insert supplementary DNS entries for every additional Punycode TLD; and the number of additional DNS entries could grow exponentially once sub-trees are taken into consideration.

In the light of these challenges, a "TLD mapping mechanism" has been proposed to give some flexibility at the user interface.[24] Within that framework, a particular Arabic TLD can have one or more alternative representations in the user interface, including, where applicable, alternative single-word format for ccTLDs without or with the Arabic noun identification letter (ال); long ccTLD names; and the two-letter abbreviations.[25] Several solutions aimed at addressing this issue have been proposed. These solutions are based either on "server-side aliasing" or "client-side aliasing".

---

[21] W. Nasr, "TLD mapping and standardization for Arabic domain names" (i-DNS.net, December 2003).

[22] Given, in other words, as the single-word country name with or without the Arabic noun identification letter (ال).

[23] A. Costello, "Punycode: a bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492 (Network Working Group, March 2003).

[24] W. Nasr, "TLD mapping and standardization for Arabic domain names" (i-DNS.net, December 2003).

[25] Ibid.

One of the ways to circumvent the multiple DNS entry requirement is "server-side aliasing" whereby a DNS alias scheme is used at the server side to map all the possible representations of a particular TLD into a single, normalized representation.[26]

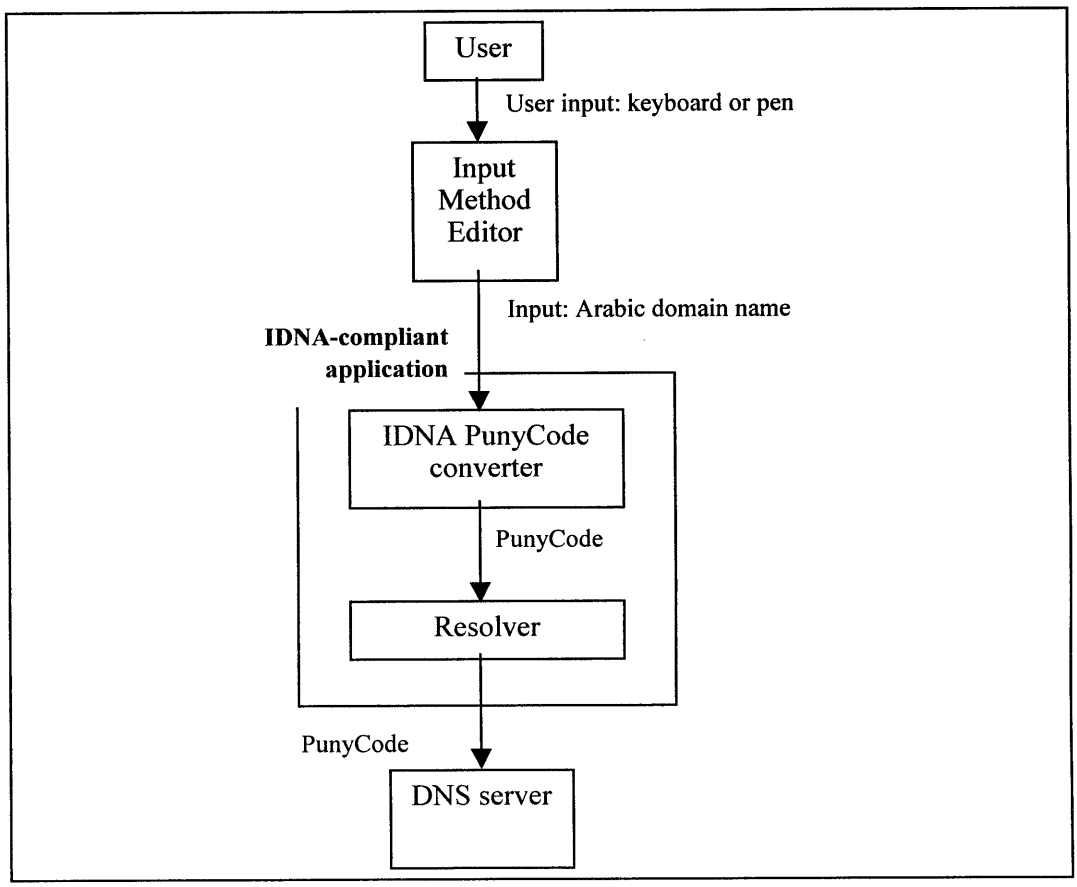
On the other hand, "client-side aliasing" involves the preloading of a TLD mapping table within client applications before any DNS resolution occurs, such that the requested TLD is mapped to its Arabic-normalized form before it is sent out for resolution. For example, all requests to any domain name with the TLD "ام" will be mapped on the client-side onto the TLD "الإمارات" (Emirates) before resolution, such that the administrator only needs to maintain one set of zone files for the TLD "الإمارات".

As the number of mappings between TLDs is limited and is not expected to change frequently, client-side aliasing for TLDs could be sufficient for the purpose of solving this technical issue.

E. Integrating Arabic domain names with other Internet services

Given that ADNS is set to be fundamentally based on the design principles and architecture of the Internationalized Domain Names in Applications (IDNA) standard, no modifications are required to the existing DNS infrastructure or to lower-layer protocols.[27]

Figure 5. Architecture for Internationalized Domain Names in Applications (IDNA)



Source: W. Nasr, "Technical documents" (i-DNS.net, 2005).

[26] DNS has several ways of performing aliasing through such resource records as CNAME, which maps only a single node of the name space; and DNAME, which is used to map or rename an entire sub-tree of the DNS name space to another domain. See M. Crawford, "Non-terminal DNS name redirection", RFC 2672 (Network Working Group, August 1999).

[27] P. Faltstrom, P. Hoffman and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490 (Network Working Group, March 2003).

The basic premise of IDNA is that any Internationalized Domain Name used in applications is converted to Punycode before DNS resolution such that the data format of DNS requests on-the-wire will be ASCII-compliant Punycode, thereby preserving interoperability between existing DNS infrastructure elements. Figure 5 illustrates the architecture in its various layers.

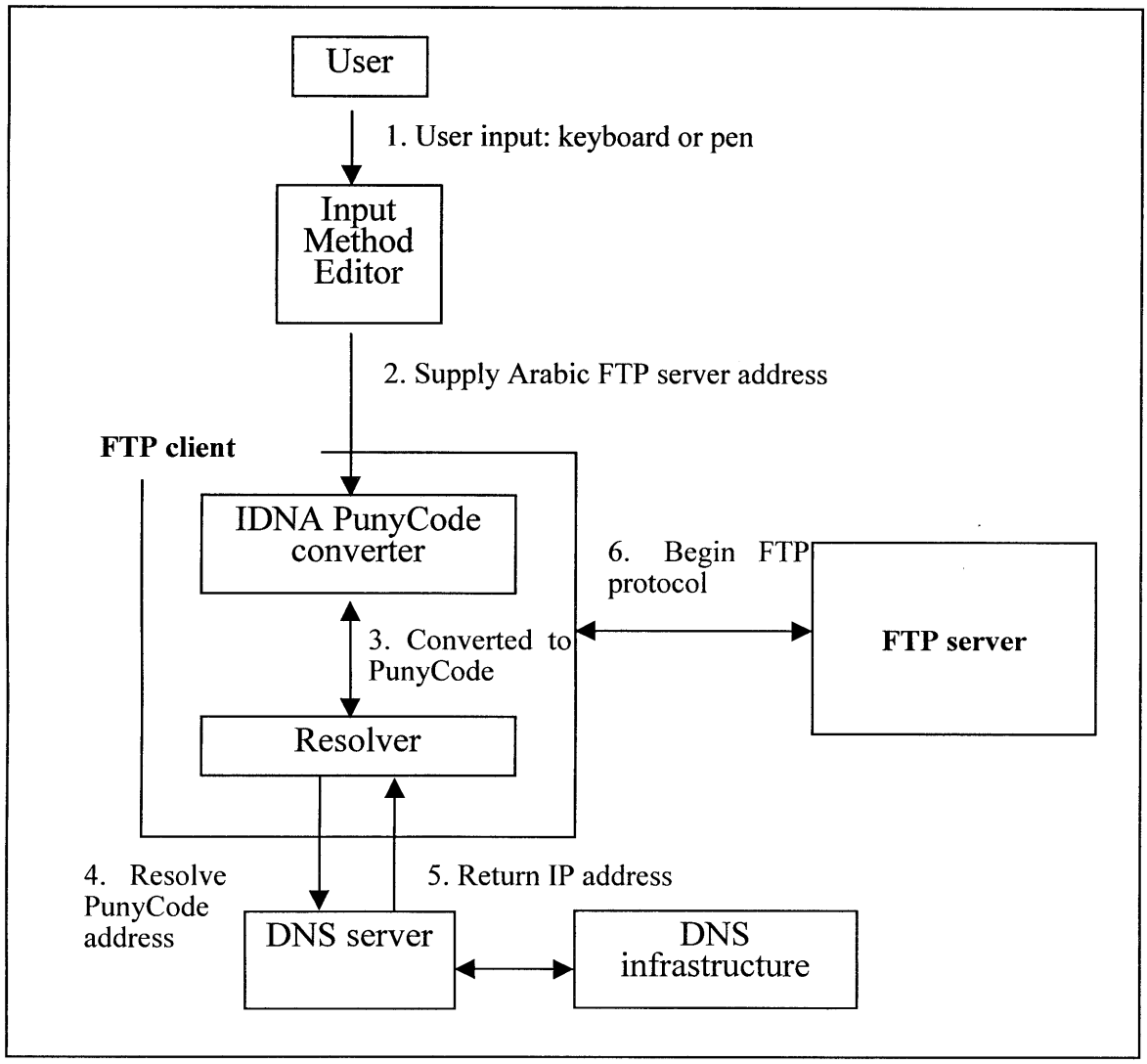
However, the applications themselves have to support IDNA and Punycode conversion before resolution in order to work with Arabic domain names and/or any other IDN.

1. *File Transfer Protocol (FTP)*

As specified by the IDNA standard, the only modification needed is to the FTP client whereby an additional Punycode converter is first needed to convert the Arabic FTP server address into Punycode before sending it to the resolver.

The resolver will then resolve the Punycode domain name with the DNS infrastructure, subsequently obtaining the IP address of the FTP server and continuing with the regular FTP process. Figure 6 illustrates the IDNA and FTP scenario.

Figure 6. IDNA in an FTP scenario



Source: W. Nasr, "Technical documents" (i-DNS.net, 2005).

2. *E-mail*

Previously, Internet e-mail addresses could only be represented using the 7-bit US-ASCII character set owing to legacy problems in existing mail systems.[28] Consequently, Internet e-mail addresses in languages that have character repertoires outside the US-ASCII range, including Arabic e-mail addresses, cannot currently be used on Internet e-mail systems.

Typically, these languages use a different coded character set (CCS), including, among others, CP1256 and ISO/IEC 10646. When these coded character sets are encoded into multi-byte octets by a character encoding scheme (CES), they form a "Charset", which can provide representation for those languages outside the US-ASCII range on Internet e-mail software and systems.[29]

While recent standards have been developed to allow different Charsets to be represented inside Internet e-mail,[30] these solutions are only applicable to the e-mail message content and some fields of the e-mail header, including the subject field, but not to the e-mail address itself. This is therefore not a complete solution given that the e-mail addresses within the current Internet system can only be represented using US-ASCII.

Within that context, several solutions based on IDNA have been proposed by IETF either through "client software modification" or through "web server/application modification" to allow the representation of internationalized e-mail addresses.

(a) *Client software modification*

This solution involves a modification to the e-mail client software, which is commonly known as a mail user agent (MUA) and which typically, albeit not always, resides on the computer that is accessing Internet e-mails. Examples of such MUA software are Microsoft's Outlook Express, Qualcomm's Eudora Pro, and Netscape's Messenger.

Whenever the end-user composes an e-mail using the MUA software and types an internationalized e-mail address in any Charset into any field where e-mail addresses are used, including the "from", "to" and "cc" fields, the IDNA conversion is applied to the internationalized e-mail address such that a resulting Punycode string is achieved.

This Punycode string then replaces the internationalized e-mail address in the specified field within the MUA software, and the internationalized e-mail address is appended to the end of the "username" portion of the specified field.

Upon the end-user's command, the MUA transmits the e-mail using the converted string to the Simple Mail Transfer Protocol (SMTP) server specified by the end-user. Given that the e-mail address now has characters within the US-ASCII range, this poses no problem to the SMTP server and subsequent e-mail programs that deal with this particular e-mail.

The recipient of such an e-mail is still able to view the original Arabic e-mail address stored at the end of the "username" portion, despite having the Punycode as the real e-mail address (see figure 7).

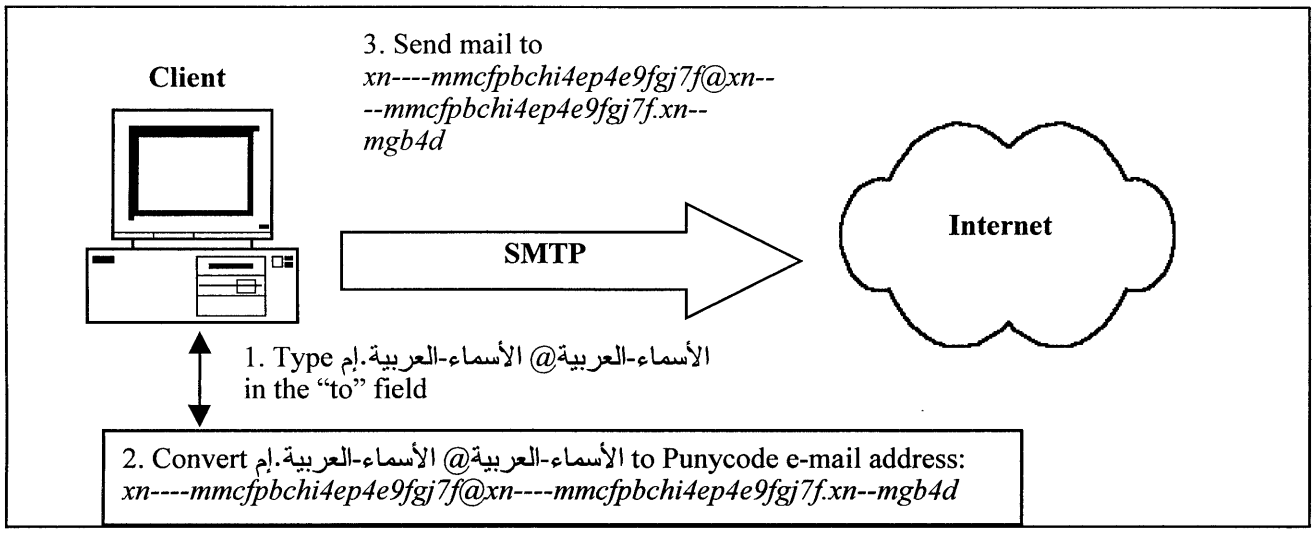
---

[28] This restriction is specified in many related e-mail protocol standards, including Simple Mail Transfer Protocol (SMTP) and the standard Internet mail format. See J. Klensin, "Simple Mail Transfer Protocol", RFC 2821 (Network Working Group, April 2001); and P. Resnick, "Internet Message Format", RFC 2822 (Network Working Group, April 2001).

[29] See N. Freed and J. Postel, "IANA Charset registration procedures", RFC 2278 (Network Working Group, January 1998).

[30] Among such standards is the Message Header Extensions for non-ASCII text. See K. Moore, "MIME (Multipurpose Internet Mail Extensions) part three: Message Header Extensions for Non-ASCII Text", RFC 2047 (Network Working Group, November 1996).

Figure 7. Client software modification solution

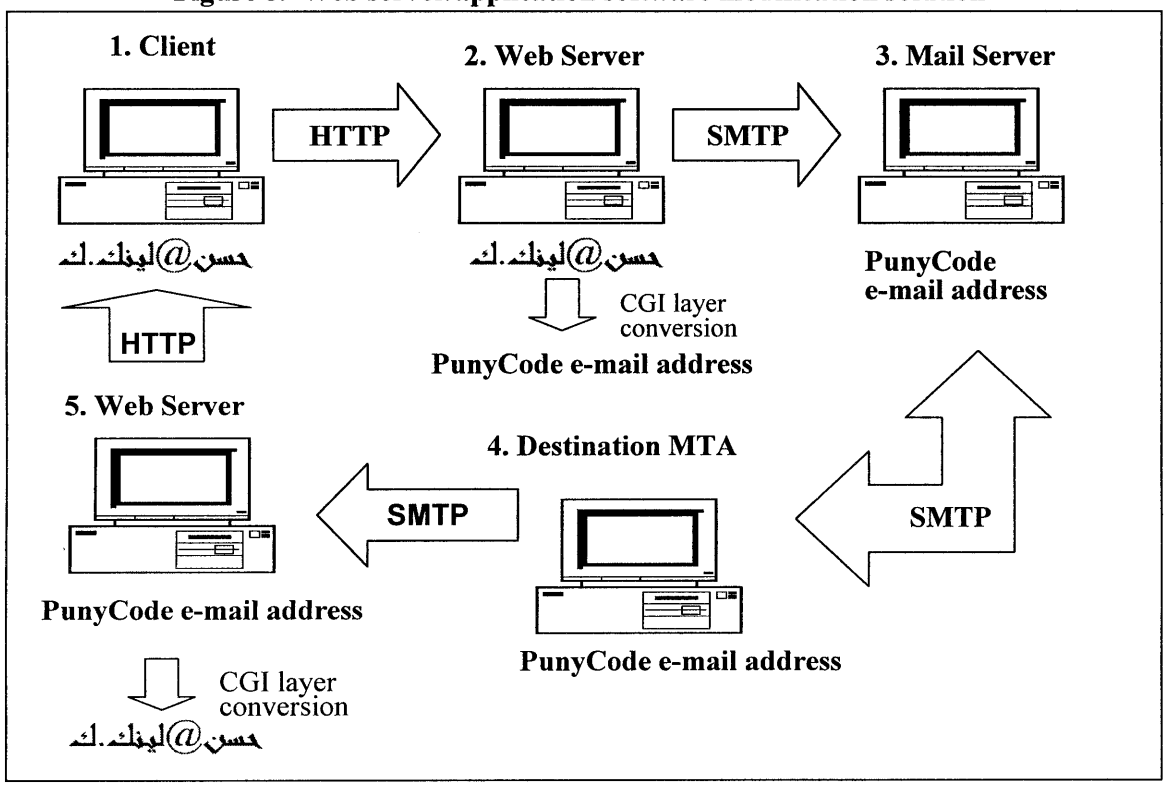


Source: W. Nasr, "Technical documents" (i-DNS.net, 2005).

(b) *Web server/application software modification*

This solution is similar to the client software modification solution, except that the conversion is undertaken by the Web server/application MUA rather than a user desktop MUA. Such Web server/application MUAs include Web-based e-mail software or services, including, for example Microsoft's Hotmail and Yahoo! Mail.[31] Figure 8 illustrates the solution in this case.

Figure 8. Web server/application software modification solution



Source: W. Nasr, "Technical documents" (i-DNS.net, 2005).

[31] These Web-based e-mail services are available at: www.hotmail.com and mail.yahoo.com.

IETF is set to opt for the client side solution in e-mail addresses whereby all the modifications are undertaken on the user's computer.

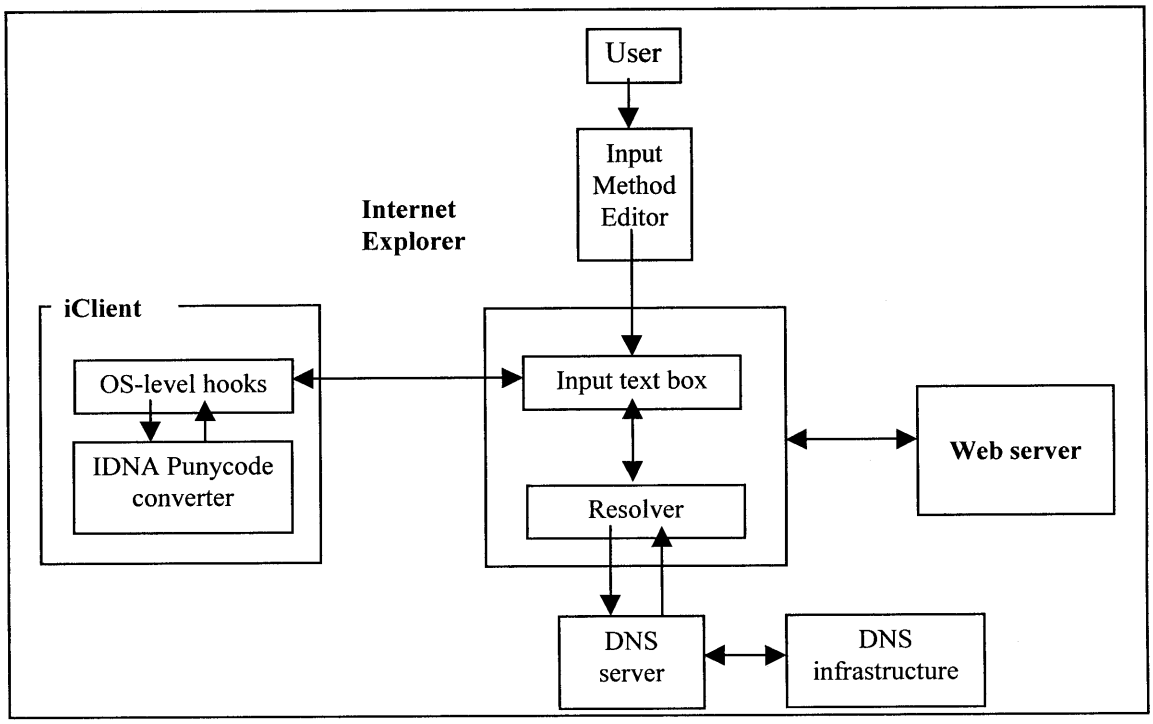
F. ICLIENT IMPLEMENTATION AND SUPPORT

Given that IDNA has been established as a standard-track IETF document for a brief period of time,[32] not many major applications support it yet.[33] Consequently, in order to provide IDNA support to existing applications, an application known as an "iClient" must be loaded by Arabic domain name users to enable using and resolving Arabic domain names.

The IDN standards clearly recommend a client-based approach in dealing with the translation of Unicode to ACE strings. In this case, the term "client" denotes a DNS client, thereby signifying that it is the responsibility of the DNS client to prepare the query in ACE before sending it to the DNS server, which will then handle the classical DNS resolving mechanism. It is essential to define the term "client", particularly given that there are two possible connotations, as follows:

(a) The client can be the application or process that needs to undertake the DNS resolving. In this case, a client per application can be identified, including, for example, Web browsing, e-mailing and instant messaging. The implementation of one extension per application is therefore needed, provided that such extensions are acceptable and an application program interface (API) is provided that permits it. One example is the plug-in provided by VeriSign under the name i-Nav, which provides Internet Explorer with the possibility to type domain names using international strings. Additionally, VeriSign provides a way for Outlook Express to send and receive e-mails that have been written in a script other than Latin;[34]

Figure 9. IDNA in an Internet Explorer scenario



Source: W. Nasr, "Technical documents" (i-DNS.net, 2005).

---

[32] P. Faltstrom, P. Hoffman and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490 (Network Working Group, March 2003).

[33] As of December 2004, the only major application that incorporates IDNA is the Mozilla Firefox browser, which is available at: www.mozilla.org. Other applications, including Microsoft's Internet Explorer are expected to move to incorporate IDNA in the future.

[34] More information is available at: www.idnnow.com/index.jsp?lang=ar.

(b) The client can be the machine running the applications. In this case, this machine could act as a client and server simultaneously. The machine can become a server for the applications running on it, while playing the role of a DNS client sending ACE strings to the server.

Within the context of the first connotation, iClient is essentially a small, unobtrusive application that runs in the background of the user's operating system. When the user starts to type an internationalized domain name into the input boxes of any application supported by iClient, the iClient system-level hooks detect this action and convert the domain name into Punycode upon completion. Figure 9 depicts its use in Internet Explorer.

Currently, iClient software from commercial companies working on multilingual solutions is compatible with Windows 95, 98, ME, NT, 2000 and XP; supports such popular browsers as Internet Explorer (version 4.0 and later) and Netscape Navigator; and can be used with a number of e-mail programs, including Eudora, Netscape Messenger, Microsoft Outlook, Outlook Express and Foxmail. Moreover, iClient supports such popular Web-based mail services as Lycos Mail, Hotmail, Yahoo! Mail and Netscape Mail.

The i-Client implementation provided by i-DNS.net International tries to intercept the string typed by the user and convert it into Punycode.[35] While this approach is certainly workable, it remains intrusive to the client operating system and is clearly dependent on the ability of the programmer to detect the string correctly and convert it before sending it to the resolver. However, it remains a valid, fully tested and operational solution for a considerable number of cases. Consequently, the second type of i-Client is recommended, whose function can be described as follows:

(a) The user enters the name of the URL regardless of the application, provided it accepts an input in Arabic;

(b) The application sends the URL to the DNS server in the standard way;

(c) A local process operating as a proxy-DNS is defined as an intermediate server, which can do the following:

(i) Accept the string provided by the application and check if it is in Latin or Arabic script;

(ii) If the string is in Latin script, it sends it to the original DNS server and forwards the result to the application that asked for it;

(iii) If the string is in Arabic script, it converts it to Punycode and sends it as in the previous case.

The only condition for this solution to work correctly is that the application must not reject the Arabic string prior to DNS resolving. Preliminary checks with such popular applications as Outlook Express and Internet Explorer show that the application can accept the Unicode string without problems.

The principle of installing a proxy server in front of another is almost standard; such a setup already exists in antivirus software. In this case, the software spawns a process acting as a mail server, which scans the message before forwarding it to the true e-mail server. In order to achieve the same approach for DNS, the settings of the DNS client need to be modified in such a way that, rather than using the DNS server of the Internet Service Provider (ISP) or of the local network, it uses the proxy DNS server (address 127.0.0.1, port 53), which in turn forwards the URL after processing.

The advantages of this solution are as follows:

(a) It is not dependent on a specific application or operating system, and requires only that the application accepts a Unicode string as URL;

(b) No special upgrading is needed with evolving operating systems, given that the principle is universal;

---

[35] i-DNS.net International is available at: www.i-dns.net.

(c) In addition to generating Punycode, this solution enables the client to perform domain mapping before the string is sent to the resolver.

Consequently, it is recommended to validate this approach through adequate tests and to add the domain mapping feature to the i-Client. Given that this i-Client would be indispensable to all users who want to make full use of ADNS, it is suggested that it could also be expanded to support other features, including, for example, replacing space characters in the URL with hyphens before translating into Punycode.

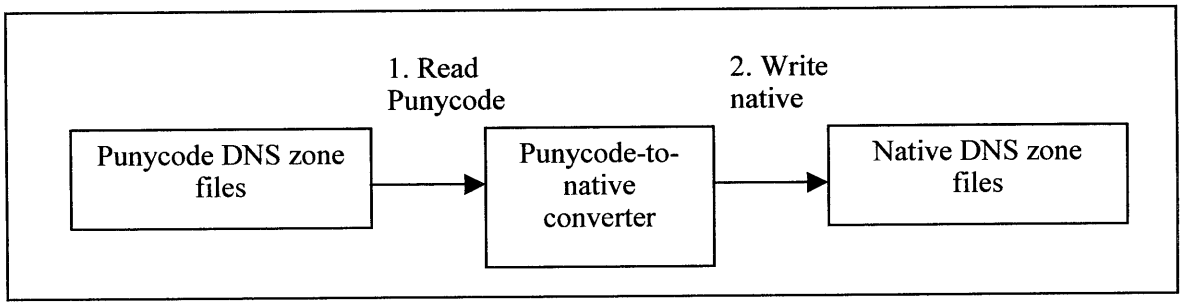
G. FUTURE CONSIDERATIONS AND MISCELLANEOUS ISSUES

1. *Unicode-compliant implementations*

The IDNA standard does not specify a roadmap or upgrade path to any Unicode-compliant, in other words "native", DNS implementation. This is due to the fact that one of the main design principles of IDNA is to preserve interoperability among all DNS infrastructure elements.

Given that any native DNS implementation is likely to face interoperability issues with existing systems, it is set to remain incompatible with the IDNA standard. However, despite the significant hurdles, were a native DNS implementation to be mandated on a specific day, which is referred to as a "flag-day" upgrade, then data in the existing IDNA Domain Name System could be upgraded through the use of a simple script (see figure 10).

Figure 10. IDNA to Unicode-compliant upgrade script



Source: W. Nasr, "Technical documents" (i-DNS.net, 2005).

However, the use of the upgrade script only upgrades host/domain names within the DNS zone files. There could be other Punycode host/domain names referenced elsewhere within the system that have to be detected and upgraded manually.

2. *Internet Protocol version 6 (IPv6) implications*

Given that the IDNA standard does not impact the existing DNS infrastructure in any way, the support of the Internet Protocol version 6 (IPv6) by the prevailing DNS infrastructure can be similarly extended to Arabic domain names.[36]

3. *Migration for existing customers to ADNS*

As specified by IDNA, no modification needs to be made to the existing DNS infrastructure. However, clients will have to install iClient or use an IDNA-compliant application in order to start using the Arabic domain names.

---

[36] R. Bush et al., "Representing Internet Protocol version 6 (IPv6) addresses in the Domain Name System", RFC 3363 (Network Working Group, August 2002).

Several users have already registered their Arabic domain names using some proprietary tools, or through regular IDN mechanisms that allow registration of Arabic domain names in existing gTLDs.

Given that the number of these users is relatively small and that they have registered their domain names outside the ".arab" gTLD,[37] there is no need to adopt a special policy for these users or to give them a systematic priority over other users. A better approach is to rely on the TLD mapping mechanisms described above in order to avoid conflicts.

4. *Migration from ADNS to IPv6*

There are no known issues with this proposed migration step given that it is expected to be comparatively seamless. The suggested solutions do not rely on any features or limitations of the current version (IPv4), and is therefore expected to be operate smoothly in the IPv6 environment.

[37] While there are no current estimations, Arabic URLs are rarely seen in the media or on the Internet.

IV. THE OPERATIONAL ASPECTS OF ADNS

A. INTRODUCTION

It is important to describe the operational aspects of ADNS in order to provide ccTLD owners with a set of guidelines and policies for operation. Within that context, the Joint Engineering Team (JET) has made a similar effort in RFC 3743, which was published in April 2004.[38]

This chapter discusses the organizational aspects of ADNS; describes the concepts of registries and registrars as adopted by ICANN, along with the applicability of this approach in the Arab region; and takes into consideration the current evolution of ADNS technical discussions as well as the orientation towards launching a new TLD specific for the Arab language.[39] The discussion equally focuses on the launching phase of a new TLD in terms of the details and inherent challenges, including the following: (a) managing the first period rush; (b) handling trademark protection handled; (c) ensuring equity among registrars; (d) ascribing the additional tasks for the registries and the registrars with regard to the application of IDN, and the specific mechanics for such application within the framework of ADNS in particular.

Current IETF documents that describe registry and registrar management methods often develop applications aimed at building DNS records based on data collected from domain owners within the guidelines of the adopted policies.

As discussed in chapter II, a single regional body needs to act as the authority for the standardization and governance of Arabic domain names. This regulatory authority needs to appoint either a single company to run a unique registry of Arabic domain names or multiple companies that can work together in coordination. At a national level, ccTLDs could be managed independently in each country by a national network administrator. Moreover, each country could manage its own Arabic ccTLD along with the standard ASCII ccTLD; while the regional regulatory authority could equally be in charge of approving the Arabic representation of ccTLD names for non-Arab countries requesting such representation.

A good commercial model is the one implemented by ICANN whereby accredited registrars are entitled to appoint resellers at a premium. Within that framework, companies that are technically qualified can act as registrars, and a wide reseller network can be established.

B. REGISTRIES AND REGISTRARS IN THE ARAB REGION

1. *Definition and issues*

The new policy and its associated business model by ICANN are aimed at increasing the competition and improving the domain name registration market. Its principal function is to divide the roles between registries and registrars.

A registry is an entity that maintains the master database of domain names for a particular TLD. This definition applies to both gTLDs and ccTLDs. The registry receives domain name information from registrars, which register domain names on behalf of registrants, institutions or people who would like to register and use a particular domain name. The registry puts that information into a "zone file", which allows computers to route Internet traffic to and from domains across the world. Using common software engineering terminology, the registry function can be considered as similar to that of a "back-office", while the registrar represents a "front-office" (see box 2).

---

[38] K. Konishi et al., "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) registration and administration for Chinese, Japanese, and Korean", RFC 3743 (Network Working Group, April 2004).

[39] A similar request has already been filed for Asian scripts.

Box 2. The process of registering a domain name

The process of registering a standard domain name entails the following sequence:

(a) The registrant chooses to reserve a particular domain name;

(b) The registrant asks the registrar through a reservation system (usually online) to undertake the registration task;

(c) The registrar sends a request to the registry to check the availability of the particular domain name;

(d) If the registry confirms the availability of the desired domain name to the registrar, the registration is performed; otherwise the registrar asks the registrant to choose another domain name;

(e) Once the registration is done, the registrant information maintained by the registrar is used by DNS services; the most important services being DNS resolving and the popular Whois service.

The registry is not a recent concept; it dates back to the early days of the Internet. However, its role and responsibilities were heavily affected by the spread of the Internet and by its concomitant use as a business tool. The role of the registry was considered to be a public service. Moreover, it was deemed more "appropriate to be concerned about 'responsibilities' and 'service' to the community" rather than being concerned with "rights" and "ownership".[40] Furthermore, the designated authorities were perceived as the trustees "of the top-level domain for both the nation, in the case of a country code, and the global Internet community".[41]

Several attributes of a domain registry derive from this classification as a public resource, including as follows: (a) openness whereby the rules and procedures must be made public; (b) neutrality such that the registry must not favour a particular party; (c) reliability; and (d) efficiency whereby procedures, including the registration of a domain, need to be timely and straightforward.

The evolution of the Internet forced a new set of attributes and constraints. Registries have emerged as the entities responsible for the management and allocation of one of the most important resources of the Internet. The increased use of the Internet by businesses transformed the domain name into a part of their identities and, consequently, engendered a plethora of issues, including the protection of trademarks and cyber-squatting. The introduction of registrars as intermediaries between the registry and the registrants added a new set of issues. Currently, in addition to ensuring neutrality between registrars, the registry has to take heed of those registrars that seek loopholes in the established registration mechanism in order to favour their customers. Several issues related to the relationship between registries and registrars have been raised during the elaboration of the business model (see box 3).

Box 3. Issues in the relationship between registries and registrars

Registries have a full monopoly over the TLD they are managing. For example, there is a single company managing the domain ".com" as a registry, namely, VeriSign. Other companies that handle the registration procedures for their customers are "registrars". In this sense, registrars operate in a way similar to retailers, with the particularity that they get their products from the same source (the registry).

The most important issue for registrars is in terms of equal access to registry services. ICANN introduced several new policies aimed at providing more choices and increasing the competition between the registries. The underlying efficiency of registries can, in turn, be reflected in the overall service the registrar offers its customers.

Safeguards need to be implemented to ensure that even in cases where the registry manages its own registrar, this must not result in unfair competition with other registrars. Such behaviour, if tolerated, simply results in driving the registrars out of business and enforcing a de facto monopoly of the registry.

As a general rule, the boundaries and lines of "split of services" between registries and registrars need to be well-defined. In particular, the issue of the control level and responsibilities of a registry over the TLD under its management needs to be studied carefully in the light of new services that could be introduced with the domain name management. For example, a registry could choose to provide the registration services directly to the end-user, thereby depriving the registrars from new sources of revenue.

[40] J. Postel, "Domain name system structure and delegation", RFC 1591 (Network Working Group, March 1994).

[41] Ibid.

Finally, the strategy by ICANN is aimed at providing a liberal and competitive approach to the domain name allocation and management. It is important to note that the issues raised above are not directly related to IDN, which raises its own, different set of issues.

2. *Registries and registrars in the Arab region: the domain name structure*

There are substantial challenges involved in projecting the previous business model on ADNS, which depend largely on the type of domains to be managed. Specifically, problems related to gTLDs are comparatively more acute than those relating to ccTLDs. If an Arabic gTLD is to be implemented, it needs to span the entire Arab region that comprises more than 20 countries, with widely varying levels of progress in all aspects, including Internet penetration. The legal and economic structure of these countries, which regulates the exchanges between them, is still comparatively unwieldy and has not evolved sufficiently to address adequately the above-mentioned issues. In particular, the economic model in most Arab countries remains largely protectionist, especially in the areas of telecommunications and the Internet, which is somehow incompatible with the registry/registrar model.

In the Arab region, there is an urgent need to answer the following question: what are the domains that the registry and registrars will be managing? The answer to this question is directly related to the structure of Arabic domains, including the selection of gTLDs and ccTLDs, and the possible support of TLD mapping aimed at providing several aliases for the same domain. A particularly useful application of TLD mapping is to enable ccTLDs to be written in different ways in the user interface following the recommendations of the first meeting of WG-ADN.[42]

In order to gain a full understanding of why TLD mapping is needed, it is important to recall the steps involve for resolving a domain name, namely:

(a) The DNS resolver asks a root server if it can identify the name server that is responsible for the domain included in the string;

(b) The root server sends the address of the appropriate name server, specifically that which is maintained by the domain registry;

(c) The classic hierarchical domain name resolving process is launched and results in the IP address corresponding to the domain name.

The Arabic TLD problem could occur in steps (b) and (c) above. The root server maintains a mapping between registered TLDs and their correspondent name servers typically maintained by their respective registries. For example, if the user types the following string: "جامعة-الدول-العربية.هيئة", the TLD part, "هيئة" is converted into its Punycode counterpart, which is not a registered domain; and the root server is therefore unable to identify the appropriate DNS server responsible for resolving the domain name.

In a global context, where all users worldwide are able to use an ADNS solution, then the TLD part needs to be recognized by all root servers. Given that it is unlikely that adding Arabic TLDs in Punycode to root servers would be easily accepted, the only viable solution is to rely on TLD mapping, which could map the Arabic TLD into a registered domain name. Consequently, regardless of the eventual structure of ADNS, TLD mapping will remain necessary.

The current IDN standards do not support domain mapping.[43] If these standards are applied literally, then users would be obliged to type a hybrid string whereby a part of the domain name is written in Arabic, and the TLD would be written in Latin script.[44] If the TLD part is kept in Latin, then it would pass the

[42] League of Arab States (LAS), "Report of the first meeting of the Working Group on Arabic Domain Names" (LAS, 2005).

[43] For example, mapping "شركة" to ".com".

[44] An example of this hybrid is "جامعة-الدول-العربية.org".

conversion algorithm unchanged and could be recognized by the root server. However, writing a hybrid domain name is totally defeating the purpose of being able to write the domain name in a language that is more natural to the end-user. While this could prove a solution for scripts that are not widely different from Latin, including, for example, Cyrillic, this could prove problematic in Arabic, particularly given the need to switch the direction of the writing in the middle of the domain name. Moreover, WG-ADN has clearly stated that it is unacceptable to use to such mixing between Arabic and Latin characters in the domain name.[45]

There are clear reasons for the lack of support of domain mapping in IDN. Chiefly, IDN was intended as a way to allow current registries to expand their clients and be able to accept registrants who want to register domain names written in their native languages. More specifically, the scope of IDN is to support a name structure of the form "arabic.ASCII", rather than "arabic.arabic". It is, naturally, this latter structure that is the ultimate goal of an adequate ADNS system.

Summarizing, a fully operational ADNS "arabic.arabic", which is accessible from anywhere and which does not mix Arabic and Latin characters, requires TLD mapping. This in turn raises the following two issues:

(a) The root server needs to be able to associate the Arabic TLD string with an appropriate DNS server;

(b) The TLD server maintained by the registry needs to map the TLD sent by the client to one of the TLDs under its supervision.

Given that the problem relates to the root server, it is there that the Punycode string provided by the resolver, which is equivalent to the ".arabic" string, needs to be associated with the corresponding TLD server. There are two ways to implement this, namely:

(a) The root server has the appropriate entry that associates the converted ".arabic" string with an existing TLD, regardless of the top-level domain;

(b) The root server does not have such an entry.

The only chance for the first case to occur is to register officially all the Punycode equivalents of the Arabic gTLDs, which, while theoretically feasible, could result in a lengthy process. Moreover, there are no guarantees that the registration process would succeed.

Even where a "rogue" root server has been installed, it would not necessarily be accessible to everyone. In particular, most users who have a dialup access through their ISPs use the DNS resolver of their service providers and cannot dictate the choice of the root server. Additionally, even if they try to install their own DNS resolvers on their machines, the security system in most ISPs blocks those resolvers.

Consequently, the second case is more realistic whereby the root server remains unaware with regard to the converted ".arabic" string. The only possible solution in this case is to provide the root server with a TLD string that it recognizes; and the mapping between the original ".arabic" string and the conversion can occur at the client side, prior to sending the string to the root server.

A client-based software managing domain mapping is therefore unavoidable, given that it constitutes the only feasible solution towards resolving the root server problem. The alternative is to establish an agreement with ICANN aimed at adding domain-mapping functions to root servers. However, such a strategy can only be considered as a long-term solution.

---

[45] League of Arab States (LAS), "Report of the first meeting of the Working Group on Arabic Domain Names" (LAS, 2005).

The downside of the client-based domain mapping solution is the limitation of the accessibility of the ADNS name space, which will be limited to the users who have access to this client. In other terms, all users across the world who want to use ADNS need to install special client software on their personal computers. While the validity of this solution can be debated for the TLD server, it is clearly the only possible solution to resolve the root server problem.[46]

Given, therefore, the need of TLD mapping, what are the possible top-level domains in an Arabic context? At this stage, the final structure of ADNS has not yet been determined, particularly regarding gTLDs. It is evident that the ongoing discussions on Arabic gTLDs preclude the use of gTLDs used in the Latin script, including, among others, ".com", ".org" and ".net". Consequently, the number of both these domains and potential registries cannot be anticipated.

Instead, it is more fruitful to adopt a reverse approach, whereby strategies must first aim at addressing the problem of creating and managing new TLDs. This approach, which is based on taking relationships with ICANN into account, needs to be considered first given that it involves considerable restrictions. Specifically, there is a vital need by ICANN to meet several requirements and undergo several phases in order to create new adequate TLDs. Within that context, the following two possibilities can be identified:

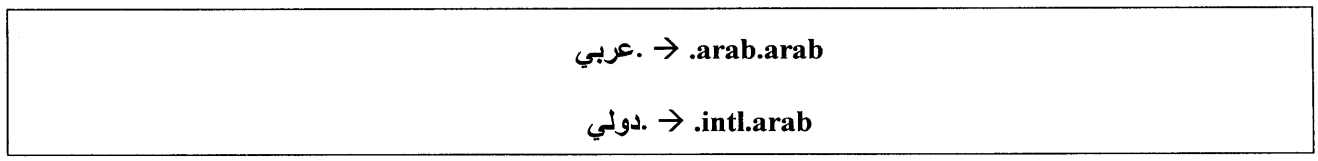
(a) To rely on TLD mapping to map the Arabic TLDs into already existing top-level domains, which already have their own registries and registrars;

(b) To define new TLDs and, following which, to initiate the relevant procedures with ICANN.

In the case of ccTLDs, the first possibility is a perfect fit, given that ccTLDs already have a counterpart in the Latin script with an identical meaning in both Arabic and Latin TLDs. The registry for Arab ccTLDs can therefore be the same as the Latin ccTLD for each country. Moreover, this is particularly reasonable given that ccTLDs are usually handled by national authorities and that, from a purely organizational perspective, the registry that already exists for Latin scripts can equally serve as a registry for Arabic domain names for the same country. This can therefore be seen as an extension of the services already provided by a given registry.

The situation is totally different in the case of gTLDs. Given that no clear mapping between Arabic gTLDs and current Latin gTLDs has been established, the first possibility cannot apply for gTLDs.[47] The only viable option is to define new gTLDs and undergo the registration procedures of ICANN, with all the associated legal and financial constraints that such registration entails. In February 2005, WG-ADN strongly recommended the registration of ".arab" TLD in order to support Arabic domain names.[48] The registration of multiple gTLDs is a lengthy and expensive process, and needs a strong consensus on accepted gTLDs before launching the process. Consequently, the following practical steps are proposed for the first phase: (a) the ".arab" gTLD needs to be registered officially, thereby paving the way for Arabic gTLDs to be defined as second-level domains under ".arab"; and (b) TLD mapping could be used to handle the translation (see figure 11).

Figure 11. Proposed TLD mapping for Arabic gTLDs[49]



[46] See chapter III, section F for an example of such a solution.

[47] Within that context, however, some registries, particularly VeriSign that owns the prestigious ".com" gTLD, offer support for IDN registration, including support for Arabic.

[48] Emulating, in a sense, the current application for ".asia" TLD by Asian countries. See League of Arab States (LAS), "Report of the first meeting of the Working Group on Arabic Domain Names" (LAS, 2005).

[49] The proposed gTLDs in figure II are those suggested by the work of the Arabic Domain Names Task Force (ADN-TF) and publications by A. al-Zoman.

3. *Recommendations regarding TLDs in ADNS*

This proposed solution requires only the registration of ".arab", given that the second-level domains are handled by the registry directly and can be created in a timely fashion. Moreover, this solution can be adopted for the short- and medium-term because it reduces the time it takes to deploy an operational ADNS system.

It is key to note that the adoption of this proposal does not in any way restrict the possibility to register other gTLDs with ICANN that are fully dedicated to Arabic support, whenever such a need arises. Registration priority could then be given to those already registered within the ".arab" domain.

The final recommendations can be summarized as follows:

(a) For the short term, efforts need to aim at the following:

(i) Installing a dedicated root server and a pilot registry serving the ".arab" TLD;

(ii) Configuring the root server to point towards the DNS of the ".arab" TLD registry;

(iii) Establishing a client that could implement Arab TLD mapping such that ccTLDs are mapped to their Latin equivalents, and gTLDs arise from the second level of the ".arab" TLD;

(b) For the long term, efforts need to aim at the following:

(i) Replacing the root server with the official root servers once the ".arab" TLD becomes officially registered;

(ii) Proposing amendments to IDN standards and DNS standards in order to resolve the problem of writing TLDs in native languages, including Arabic. The best solution could be to extend the character set of DNS to accept Unicode, rather than forcing a translation into ASCII Compatible Encoding (ACE);

(iii) Suggesting modifications to ICANN, assuming the resolution of the preceding point, aimed at facilitating the creation of Arabic-Unicoded TLDs as well as TLDs in other languages. It is perfectly reasonable to request such modifications that several Arab gTLDs could be registered with ICANN in their native forms, rather than resorting to mapping under the ".arab" TLD.

While no formal proposal has been formulated yet, it is recommended that all future Arabic gTLDs be registered as second-level domains of the ".arab" domain, and that registrant domain names be registered at a third level. Given the inevitable delay caused by the process of registering officially the ".arab" domain, the selected registry could be visible by a dedicated root server. This forms part of the pilot project that is set to be accessible across the Arab region. The subsequent official registration of the ".arab" TLD would make the registry operational for users across the world.

In terms of ccTLDs, it is recommended to rely on domain mapping to map the Arabic ccTLDs into already existing Latin ccTLDs and, moreover, to support already existing registries to expand their services aimed at supporting the Arabic script and in accordance of ICANN recommendations described below.

C. ADNS PROPOSED REGISTRIES AND REGISTRARS

The model defined by ICANN advocates a unique registry for each TLD, with ICANN behaving as a regulator in order to ensure that the registry respects all the relevant issues. Clearly, the most appropriate registry is one that enjoys considerable experience managing a TLD, along with a good reputation within the Arab region.

The registrars issue is even more complex, given that it engenders the challenge posed by the direct relationship with the customer or registrant. Within that context, the most important issue is the scope of work of each registrar and, more specifically, how those responsibilities and functions fit with the geographic decomposition of the economic space in the Arab region. Given that most Arab countries regulate their telecom markets in a very restrictive way, it is likely that the first reaction of the national telecom regulatory bodies will be to limit the scope of action of registrants at the national level. National markets are typically too small for a registrar to survive, particularly given the competition from international registrars who can currently register companies with Arabic domain names in the ".com" that supports IDN. Given the initial size of the market, it is therefore unlikely that more than a handful of local registrars will be able to survive during the initial two-year period.

The business model of registry/registrar in Arab countries will be difficult to implement if there is no direct support from the regulatory bodies in the Arab region and from LAS. The following scenario can be described as the most realistic for the current stage:

(a) A unique registry is selected for all the Arab gTLDs. This registry needs to operate as a non-profit organization aimed at supporting the development of the Internet in the Arab region. Registrar companies must be allowed to operate in all Arab countries under a licence agreement with the registry. This operation could have a period of three years, for example, which allows sufficient time for the concept of registry/registrar to be implemented and for all outstanding issues related to domain name registration and management to be dealt with appropriately;[50]

(b) The process of selecting the company to run the registry is to be left to the procedures adopted by LAS, with a recommendation that the registry needs to collaborate with an already established international registry that has a proven track record of running a registry in an IDN environment. At the expiration of the initial three-year period, the registry experience can be evaluated and a new scenario can be chosen, eventually leading to the transformation of the registry operation into a commercial one or to the selection of a new registry through an open auction;

(c) Registrars for gTLDs need to be allowed to operate freely in the Arab region. Owing to the virtual nature of their services, registrars do not need to be physically present in order to provide registration services. The lack of a governing body to resolve disputes between the registry and registrars can become a delicate problem. Consequently, the relationships between these bodies need to be studied carefully and their respective obligations must be clearly determined within the framework of contracts signed by the registry and the registrars;

(d) In the area of ccTLDs, the current registry of ccTLDs in Latin script is a natural candidate for the registry of corresponding Arabic ccTLDs. For such a case, national ISPs could be allowed to operate as registrars.

1. *Establishing a new registry and associated challenges*

In addition to the standard tasks undertaken by the registry,[51] the establishment of a new registry poses various significant challenges, which can be summarized as follows:

(a) Handling the initial rush of registrants. While under normal operations, requests by registrants are handled on a first-come-first-served basis and do not pose any major difficulties, in the initial stage such treatment is not possible. The solution for this problem consists of creating a "queue" and sorting requests coming from different registrars;

---

[50] The registry could receive funds from LAS or other donor organizations. Additionally, the establishment of the registry is one of the activities of the ESCWA project for developing ADNS and is included in the list of projects of the Regional Plan of Action for building the Information Society, as part of the WSIS process.

[51] These standard tasks include maintaining the registry database; accepting and validating requests from registrars; and providing information to the Whois service.

(b) Resolving conflicts between registrants who are asking to reserve the same domain name. While the sorting system needs to resolve most of the conflicts, trademark protection remains an issue given that the owner of trademark still has the right to challenge the registration of a domain name that uses proprietary trademarks. The registry needs to have a policy aimed at reducing the number of conflicts and ensuing legal actions;

(c) Protecting trademarks and restricting abusive registrations represent some of the most important tasks for a registry. This particular issue was raised during the establishment of such new gTLDs as ".biz" and ".info". The intellectual property community made it clear that strong protection for trademark holders was essential to secure its support for launching a new registry. The ICANN Intellectual Property Constituency (IPC) assessed each application for a new gTLD on the basis of several criteria, including protecting the rights of trademark holders; providing efficient dispute resolution mechanisms; using enforcement mechanisms; ensuring the adequacy of Whois service; and discouraging abusive registrations.[52]

New Arabic TLDs are expected to face similar challenges once they are launched. Annex II provides the salient techniques and recommendations for an adequate policy aimed at addressing these issues.

2. *Recommendations for the new gTLD*

The launch of a new gTLD should start by a stage during which verifying registrations by the use of online databases and other means occurs in a cost-effective manner.[53] Subsequently, a notice is to be sent to prospective registrants and trademark holders of their respective claims prior to adjudication, perhaps on the basis of the familiar Uniform Domain-name Dispute-Resolution Policy (UDRP), rather than on the new Start-up Trademark Opposition Policy (STOP).[54]

The process used to allocate names by ".info" and ".biz", which is referred to as a "round robin", suffers from being exposed to manipulation of the system. Some registrars kept their list of desired names short and offered coveted slots to their best customers. Others used registrars they controlled to do the same, while they opened their own lists to the general public. While the ".name" registry sought to eliminate the advantage of submitting shorter lists by using random batch processing, it did not prevent registrants from submitting duplicate requests through multiple registrars. Admittedly, the dilemma of how best to allocate names does not have an easy solution. The most appropriate method depends to a great extent on which underlying values should be given priority. The standard approach of first-come-first-served, which is usually adopted in the daily operations of a registry, cannot be adopted here. Moreover, in terms of already established domains, the allocation is being done in real-time, while the launch of a new domain is accompanied by a much higher number of requests coming simultaneously from all registrars. Consequently, the concept of "who came first" becomes speculative and cannot be applied here.

D. General considerations for IDN-based registries and registrars

1. *ICANN guidelines*

On 20 June 2003, ICANN issued a set of guidelines for the implementation of IDN.[55] These guidelines concern mainly the registries, given that these are largely responsible for the maintenance of DNS data (see box 4).

---

[52] Several approaches were adopted to address the trademark issues. For an in-depth analysis of each of these approaches, see Summit Strategies International, "Evaluation of the new gTLDs: policy and legal issues" (10 July 2004).

[53] This stage is often referred to as the "sunrise period".

[54] The Start-up Trademark Opposition Policy (STOP) was used particularly for the ".biz" registry given the comparatively higher incidence of trademark registrations in that gTLD.

[55] Internet Corporation for Assigned Names and Numbers (ICANN), "Guidelines for the implementation of Internationalized Domain Names", version 1.0 (ICANN, 20 June 2003).

Box 4. Summary of ICANN guidelines for the implementation of IDN

- Strict compliance with IDN standards (RFCs 3490, 3491, 3492);
- In implementing the IDN standards, top-level domain registries will employ an "inclusion-based" approach (meaning that code points that are not explicitly permitted by the registry are prohibited) for identifying permissible code points from among the full Unicode repertoire;
- The registry must associate registration with one or more languages and employ language-specific registration rules, including, for example, reservation of domain names associated with character variants;
- Registries and registrars need to provide informational resources and services in all languages for which they offer IDN registration.

Out of these four ICANN guiding principles, the second and third are the most important. The second guideline is concerned with the respect of the code points by the registry. Given that the code table has been agreed upon, enforcing the respect of this code table must therefore be part of the agreement with the registry.

When a registrant requires the registration of a domain name, several variants of that domain name may also be reserved automatically. This is referred to as a "bundle" and is the substance of the third guideline. Moreover, such "character variants" may correspond to the case when two or more Unicode points share one meaning or are linguistically deemed equivalent by some concerned authorities. This issue has also been referred to as "character folding" in the Internet Draft by ADN-TF.[56]

Given that the current recommendations do not advocate character folding at all, then there should be no automatic registration of additional domain names. However, registrars still have the option to provide bundles through their registration mechanisms to meet the requirements of the registrants. For example, the character "ي" is written as "ى" in Egypt. Registrars who are serving the Egyptian market could therefore choose to offer their customers the option to reserve both domain names.[57]

2. *Registry and registrar activities*

In addition to the standard steps for registering a standard domain name (see box 2, above), in this case IDN registration includes an extra step, namely: the generation of Punycode. This process converts the name into an ASCII string with the prefix "xn--" that can be searched at the TLD name server to determine the location of the website.[58] Clearly, there is therefore a need to determine the entity responsible for generating the Punycode.

While there is no apparent standard on this issue, current IDN implementations suggest that the registrar is responsible for carrying the Punycode generation task. Additionally, the registry could provide the registrar with a relevant toolkit that supports valid Punycode registration. The registry is then responsible for verifying that the Punycode sent by the registrar is compliant with the specifications. The recommended task distribution is therefore as follows:

(a) The registrant selects the wanted domain name to reserve;

---

[56] For example, in an Arabic context, it is deemed linguistically wrong to fold different forms of hamza to corresponding vowels.

[57] Note that this is not equivalent to character folding, because these domain names are different and correspond to two different entries inside the registry database.

[58] Punycode ("xn--") was accepted as the IDNA standard by the Internet Assigned Numbers Authority (IANA) on 14 February 2003.

(b) The registrant asks the registrar, usually through an online reservation system, to undertake the registration task;

(c) The registrar can generate possible domain name variants and suggest to the registrant bundle reservation;

(d) The registrar generates the Punycode for the required domain name(s);

(e) The registrar sends a request to the registry to check the availability of the domain name written in Punycode;

(f) The registry checks the availability of the domains, and the conformance of the Punycode sent by the registrar with the legal Arabic character set officially associated with ADNS;

(g) If the registry confirms the availability to the registrar, then the registration is completed. Otherwise the registrar asks the registrant to choose a new domain name.

E. CONCLUDING RECOMMENDATIONS

In terms of the operational aspects of ADNS, the following recommendations can be summarized:

(a) The ".arab" TLD is to be officially registered;

(b) Arabic gTLDs need to be second-level domains of the ".arab" TLD, while ccTLDs can be kept unchanged;

(c) Domain mapping is a necessity and must be supported at the client side;

(d) A non-profit registry must be established to manage the Arabic gTLDs for an initial period, after which the registry can become a commercial entity; and adequate safeguards must be implemented and maintained during this initial phase, thereby avoiding to the greatest extent possible any eventual rush and conflict;

(e) The registry needs to provide the registrars with an IDN compliant toolkit for registration in order to manage the Punycode generation and ensure that it conforms to the specifications;

(f) Registrars are encouraged to provide the clients with "bundles" consisting of several allocated domain names, thereby avoiding ambiguity issues and cyber squatting.

V. GENERAL CONSIDERATIONS RELATED TO THE RELIABILITY OF DNS

A. INTRODUCTION

Most end users do not realize the existence of the underlying DNS service and the effort it takes to transform the domain name string into an IP address. They are so accustomed to using symbolic names in their requests that they take them for granted. There is no doubt that DNS service is the most solicited service among all Internet services. Thanks to such optimizing techniques as DNS caching, the response time of this critical service has been kept within reasonable limits. While performance questions are not frequently raised, the issue of DNS reliability, including the ability to resist malicious attacks using DNS, has become a major issue during the past couple of years.

This final chapter is aimed at identifying reliability problems that could occur, and at exploring ways to avoid such problems or limit their effects. While these challenges are certainly not specific to ADNS and affect the overall DNS architecture, the current efforts aimed at setting up a root server for the Arabic domain name service (eventually several root servers) must necessarily take these reliability and availability issues into consideration.

B. DNS VULNERABILITIES

Three types of vulnerabilities can be found in the domain name system, namely:

(a) Vulnerabilities intrinsic to DNS hierarchical structure, which links the operation of DNS to a very limited number of root servers and TLD servers. If these servers are attacked successfully, then the whole system is compromised;

(b) Vulnerabilities arising from security gaps in the specifications of DNS protocols. One example of this is the lack of authentication of the name-address information provided by the DNS server. RFC 2535, which defines security extensions to DNS, addresses this vulnerability that can be exploited to implement the well-known DNS poisoning attack.[59] A fully secure DNS system is set to require several years of testing and implementation;

(c) Vulnerabilities owing to implementation bugs in DNS software, particularly the very popular Berkeley Internet Name Domain (BIND) software.[60] A familiar example is the buffer overflow problem that arose under various forms in BIND 4 and 8, and that allowed attackers to gain root access to the server, with all the consequences which can result from such illicit and dangerous access.

While the second and third vulnerabilities cannot be ignored, they constitute the direct responsibility of the system administrator managing the machine and selecting the operating system. The first vulnerability relates to the top level of the DNS hierarchy (the root servers) and, more critically, the top-level domain servers. It is this vulnerability that is the basis for the discussion below.

C. THE DNS HIERARCHY

1. *Root servers*

The Internet DNS functions in a top-down fashion whereby requests begin at the root servers, at the root of the inverted tree of DNS servers. A total of 13 root servers form the top of this tree, with names ranging from A to M, and all are included in the domain "root-servers.net".

---

[59] During such attacks, rogue DNS servers redirect Web pages to different sites.

[60] Berkeley Internet Name Domain (BIND), which is an implementation of DNS protocols, has almost become the DNS server of choice.

Most root servers are located in the United States of America. However, there are also root servers in Japan, Sweden and the United Kingdom of Great Britain and Northern Ireland. Given that DNS lookup start with the root servers, the DNS resolver cannot look for such entries as "a.root-servers.net", rather it uses a hard-coded list for these servers.[61]

In June 2000, RFC 2870 defined the best practices for root servers.[62] The goal was to ensure that the root servers could perform correctly even under extreme conditions. The salient recommendations regarding reliability and availability are as follows:

(a) At any given time, servers must be able to handle three times the peak load of requests for root data under normal operating conditions.[63] This is intended to ensure continued operation of root services in case two-thirds of the servers are out of operation, whether by intent, accident or malice. This requirement extends also to the bandwidth available to the server, and to the multiplicity of connectivity providers;

(b) Each root server must have sufficient connectivity to the Internet in order to support the bandwidth needs of the above requirement. Connectivity to the Internet needs to be as diverse as possible. Root servers must have mechanisms in place to accept IP connectivity to the root server from any Internet provider delivering connectivity at their own cost. Servers must provide authoritative responses only from the zones they serve;

(c) Servers must disable recursive lookup, forwarding or any other function that could allow the disclosure of cached answers. Additionally, they must not provide secondary service for any zones other than the root and "root-servers.net" zones. These restrictions help prevent undue load on the root servers and reduce the chance of their caching incorrect data.

In addition to the above-mentioned requirements, physical and logical network security must be of the level provided for critical infrastructure of a major commercial enterprise.

The global network of root servers has already been attacked. An infamous attack that targeted all the root servers took place on 21 October 2002.[64] The 13 DNS root servers suffered a heavy denial of service (DoS) attack that failed to disrupt service. One particularly important reason for this is the caching technique used by the DNS system, which allows DNS servers to reduce the number of times they consult the root servers. The TLD information usually expires within two days, which means that a DNS resolver will communicate with the root servers only once every two days, thereby minimizing the risk of facing an interruption of service owing to an attack against root servers.

Consequently, the only real risk of interruption of service is expected to occur during the initial operation period while the domain ".arab" is not yet registered and, therefore, not recognized by the existing root servers. In this case, a private root server could be setup and the hint files at the DNS resolvers, which recognize the Arabic domain name, could point to that root server. It is clear that this temporary situation remains much more vulnerable than a network of 13 root servers.

For this reason, it is strongly recommended that, during the initial phase, at least two and preferably three root servers need to be established across the Arab region, which strictly observe the operational rules described in RFC 2870.

Given that these root servers could cease to operate once the ".arab" domain is officially registered, the reliability of the root server system could become a part of the global root servers network. Still, the

---

[61] This hard-coded list is usually known as a cache or hints file.

[62] R. Bush et al., "Root name server operational requirements", RFC 2870 (Network Working Group, June 2000).

[63] This is usually measured in terms of number of requests per second.

[64] See R. Farrow, "Trouble with DNS" (August 2000), which is available at: www.spirit.com/Network/net0600.html.

installation of a root server in the Arab region remains strongly recommended, thereby improving response time of the DNS service.

2. *The top-level domain DNS server*

The DNS server managed by the registry is the second critical point in the DNS operation. While the private DNS servers of ISPs and enterprises are equally important, their failure or interruption of service does not compromise the operation of the whole DNS system.

The gTLDs receive many more requests than the root servers and are more critical to the operation of DNS. The root servers simply point to the gTLDs and ccTLDs, which in turn return the addresses of the authoritative names servers for most domains. While there are no RFCs for specific TLD server requirements, these can largely be inferred from those relating to root servers in general. These are as follows:

(a) Time to Live (TTL) caching, whereby a query is resolved through a cached copy of DNS data on the user's local DNS server, eliminates the need to contact the root and TLD servers. However, the time-to-live of cached information periodically expires, thereby causing a percentage of queries to traverse some of or the entire DNS tree. While long TTLs increase the amount of time that the caches can be used, they reduce the data accuracy for DNS. Typically, a corporate DNS server sets the time-to-live of its information to a few minutes. By contrast, the information at the root server has a TTL of two days;

(b) Redundant authoritative name servers, which were the original specifications for DNS, required that domain owners employ multiple servers configured to answer authoritatively for their domains. This requirement provides for reliability given that failure by any individual server does not cause DNS to fail. Multiple servers are therefore candidates for the query try-retry strategy;

(c) Query try-retry strategy, whereby a query sent to an authoritative server does not return an answer for several reasons. Whatever the cause, the resolver will retry one of the redundant authoritative servers. The retry occurs after a time interval long enough to allow most responses to return, which is intended to reduce the number of unnecessary packet retries. However, these delays cause timeouts, which lead to a poor customer experience.

These classical schemes for DNS reliability are more than enough for normal operation. However, if the goal is to achieve a reliability of 99.99 per cent, which could be a requirement for commercial operations, then a plethora of techniques could be applied. However, these do not involve the TLD and they are out of the scope of this study. Besides, several other hurdles could have a greater impact on the service than DNS, including less-than-perfect telecom infrastructure which is the case in most Arab countries.

D. CONCLUDING RECOMMENDATIONS

In conclusion, the following are recommended:

(a) Until the ".arab" domain is registered, a redundant root server configuration needs to be established during the initial phase, at least two and preferably three root servers;

(b) IETF recommendations for root servers must be adhered to;

(c) Given that these are classic and well-proven techniques, there is a need to rely on redundant authoritative name servers and caching, thereby ensuring the reliability of the top-level domain DNS server.

Critically, DNS reliability and availability are key issues. Consequently, it is essential to assign the registry task to a company that has a proven experience in this domain.

Annex 1

IDN STANDARDS[65]

A. IDN STANDARDS UPDATE

DNS only recognizes the ASCII characters "A" to "Z", "0" to "9", and "-" (hyphen). This limits the number of characters that can be used to build domain names to 37 of the more than 40,000 characters identified within Unicode. In order to create domain names from the wider range of Unicode characters, a character-encoding scheme that uniquely maps Unicode code points to an ASCII representation must be used and standardized.

Within that context, IETF has led the efforts aimed at standardizing the way non-ASCII characters are to be represented within DNS; and has published three standards related to IDNs, namely: (a) encoding scheme; (b) name preparation; and (c) IDNs in applications.

1. *Encoding scheme*

The encoding scheme for IDNs is an ASCII Compatible Encoding (ACE) that is set to encode the local language characters of an IDN into ASCII characters such that DNS can accurately answer a request for an address record. There are several types of ACE. In order to select an ACE as the standard, IETF must consider the difficult balance between compression and implementation. The preferred ACE will allow the greatest number of characters (code points) to be represented and will not be difficult to deploy. Currently, Punycode is the leading candidate for such an ACE.

2. *Name preparation*

The name preparation standard provides the rules that will ensure uniqueness in registering Unicode code points. The rules outline the criteria through which a set of non-ASCII characters will be refined to ensure that there is no ambiguity within the registrations of a specific name space. These rules are mapping, normalization and prohibition.

(a) *Mapping*: Characters can be mapped to nothing, to a single character or to multiple characters based upon their usefulness in terms of text or sense. For example, the soft hyphen (u+00AD) is invisible or ignored in terms of mapping, and is largely used within text to clarify meaning. The more common example is the mapping of a capital letter to a lowercase letter such as "B" (u+0042) to "b" (u+0062). This is to ensure that a registration such as "ibm.com" does not have a conflict with other registration such as "IBM.com" or "iBm.com";

(b) *Normalization*: Once a set of characters has been mapped, the set is normalized. Some input method editors (IME) enter characters that look exactly like another character, but have different code points. Normalization also ensures predictable results through ordering where characters have a number of combining diacritics;

(c) *Prohibition*: After normalization, the mapped and normalized set of characters is checked against a table of prohibited characters. These characters are prohibited for a variety of reasons. The most common prohibited characters are the space character that could lead to confusion, and control characters that cannot be displayed.

3. *IDNs in applications*

The IDN in applications standard focuses on the location where the Unicode to ASCII mapping will take place. The approach by IETF makes the applications that send and receive traffic from DNS, including browsers and e-mail clients, encode and decode the Unicode characters.

---

[65] The IDN standards are available at: www.verisign.com.

All these above-mentioned issues are outlined in the IETF Internet draft, entitled "Preparation of internationalized host names", which is currently being reviewed and will be updated by VeriSign IDN Testbed. However, enhancing the current DNS to include more than just English characters is not a simple undertaking. There are a number of outstanding issues surrounding the deployment and use of IDNs that need to be resolved by IETF.

B. CHARACTER VARIANTS

The majority of domain name registrants register domain names that have meaning for them in their language. Specifically, the domain name can be a name, word or phrase. While these words or phrases have meaning in the registrant's language, they could have different meanings or connotations in other languages and cultures.

The domain name registration process was designed without consideration of a language context. Technically speaking, the registrant registers a domain name using a set of characters within a script, rather than any specific language. For example, the Latin script is used by many languages including, among many others, English, French and German.

The overlap between scripts and languages define the variant issue. The IDNA protocol enables the translation of all Unicode code points into unique ASCII strings. This broader range of characters has the potential to cause end-user confusion due to characters with similar appearances or interpretations, which are equally known as variants. There is a strong need therefore to address the variant issue in order to reduce confusion and improve the end-user experience.

While there are different types of variants, character variants are not covered by recent IDN-related RFCs. Communities across the world, particularly in the Asia-Pacific region, have asked TLD registries to address character variant issues in their domain spaces. Implementing its character variant solution helps to improve the end-user experience.

C. IETF AND ICANN RECOMMENDATIONS[66]

IETF is a large and open international community of network designers, operators, retailers and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual.

The premise of this paper is that it is a mistake for ICANN to pursue a burdensome and/or intrusive approach to IDN implementation by, for example, putting ICANN in the position of approving a character-equivalence table for each language, and of maintaining such tables. The deployment of IDNA within existing top-level domain registries is fundamentally a registry responsibility; and the registries will be in the best position to make appropriate implementation decisions themselves, and need to have the freedom to make adjustments as experience dictates. Just as DNS registries embrace a wide diversity in registration policies and administrative procedures by reflecting the diversity of local Internet communities, it is apparent that the vast diversity of human character sets and the languages from which they arise compels a language-by-language, registry-led approach to the development of detailed registration policies and administrative procedures.

[66] These recommendations are available at: www.icann.org/riodejaneiro/idn-topic.htm#5.

Annex II

IMPORTANT TECHNIQUES TO CATER FOR THE CHALLENGES FACED BY A NEW REGISTRY[67]

A. THE "SUNRISE" APPROACH

This approach was adopted by Afilias Limited, the registry for the ".info" gTLD. The ".info" start-up plan consisted of two major phases, namely: the "sunrise" period and the "land rush" period. The ".info" sunrise period opened on 25 July 2001 and closed, after four rounds, on 31 August 2001. During this period, trademark or service mark owners whose registrations had been validated before 2 October 2000 could register a domain name identical to the textual or word elements of that mark.[68] Afilias charged registrars $5.75 for each sunrise domain name registration per year, with minimum and maximum terms of 5 and 10 years, respectively.

Any third party could challenge a suspect sunrise registration by invoking the Sunrise Dispute Resolution Policy before 26 December 2001, and by paying $295 to the World Intellectual Property Organization (WIPO). In order to succeed, a challenger had to establish one of the following: (a) the registrant did not own a current mark; (b) the registration was not of national effect; (c) the second level of the domain name was not identical to the mark; or (d) the trademark registration was issued after the cutoff date. The Sunrise Policy was intended to operate more quickly and be less costly than a Uniform Dispute Resolution Process (UDRP).

During the "land rush" period, Afilias applied a "round robin" system to provide domain names to the registrars. Each registrar submitted a list or queue of names that it was seeking to register on behalf of its customers. Afilias randomized each queue, as well as the order of the draw for each round, so that the registrar selected to go first in each round would be unlikely to be the same in subsequent rounds. The round robin proceeded until each name in every queue had been checked for availability. A total of 306,017 domain names were awarded in this fashion. A second land rush period was provided and allowed the registration of a further 1,500 domain names.

B. THE INTELLECTUAL PROPERTY (IP) PROTECTION APPROACH

NeuLevel, which is the registry for the ".biz" gTLD, designed a system for intellectual property (IP) protection that involved several phases of an "IP claim service". During the first phase, all trade and service mark owners interested in protecting their mark could enroll in the service by completing an "IP claim form". This form had to be for a ".biz" domain that was identical to their trademark and include a description of the goods and services for which the mark was being used; the date of first use of the mark in commerce; and the country and registration number. Any IP owner could file an IP Claim, irrespective of whether its rights derived from trademark registration or common law.

The first phase consisted of accepting IP claims, which totaled 80,008 forms after an 80-day period. The registry pointed out in its instructions to claimants that completing an IP claim was not the equivalent of registering that name; rather it was a way to put others on notice of an impending registration that may infringe on their rights.

In the second phase, the registry compared all domain name applications against all IP claims. For each match, the applicant for the domain name and its registrar were notified that another party had claimed intellectual property rights in the domain. The notification included information regarding the trademark claim to help the domain name applicant decide whether to proceed with registration. If the applicant decided

---

[67] This section is based on Summit Strategies International, "Evaluation of the new gTLDs: policy and legal issues" (10 July 2004).

[68] That cutoff date was selected because it was the date of the application to ICANN by Afilias to operate the TLD, which described the planned sunrise mechanism.

to proceed to register the name notwithstanding the IP claim, then subsequent to a successful registration, the name was placed "on hold" for 30 days.

In the third phase, during the 30-day hold, the registry notified all IP claimants of the identity of the registrant and its Whois information. Once notification of a registration was given, the claimant had 20 days to decide whether to contest it by filing a Start-up Trademark Opposition Policy (STOP) action, which could result in the transfer of the domain to the claimant. During the "land rush" period, the registry adopted a round-robin method similar to the one applied by the ".info" registry.

C. THE DEFENSIVE REGISTRATION APPROACH

The Global Name Registry, which manages the ".name" gTLD, allowed intellectual property (IP) holders with trade or service marks of national effect to submit a defensive registration (DR) for the corresponding domain name. These registrations were not the equivalent of a "live" domain name registration, given that they did not resolve within DNS. Rather, they blocked a particular name and precluded others from obtaining certain registrations. During the first phase of the process, a DR had to match the textual elements of the mark; the mark had to have national effect; and the mark had to have been registered before the registry operation started. During the second phase of defensive registration, it was not necessary to meet these requirements; any entity could apply for a DR to protect any name or combination of names.

A registrant seeking to register a name that was already protected defensively received a notice through its registrar to that effect. If the prospective registrant wished to register anyway, it would have two options. First, it could seek consent directly from the defensive registrant. Alternatively, it could challenge the defensive registrant's eligibility under the Eligibility Requirements Dispute Resolution Policy (ERDRP). A party was not permitted to receive any compensation in connection with a decision to grant consent. If a challenge succeeded, the party could proceed to register the domain or e-mail address and the defensive registrant received a "strike". After a certain number of strikes, the DR was cancelled. The registry also offers protection in the form of a "NameWatch" Service, under which subscribers are notified if a third party registers a particular name.

During the land rush period, the registry used a "random queue system" that filtered each registrar's queue to eliminate duplicate submissions. The "unique entries" from each registrar's queue were merged into a single pool. For any duplicates, one unique entry was selected at random and entered into the same pool. There was therefore no inherent advantage to being on the list of a smaller registrar or on the "preferred" list of a larger registrar, given that all preferences were combined into a single pool. There was still, however, an advantage to be gained by being in as many registrar queues as possible.