

# 中文域名在網站瀏覽上的研究

曾黎明

國立中央大學資訊工程研究所  
分散式系統實驗室

邱俊穎

國立中央大學資訊工程研究所  
分散式系統實驗室

陳奕明\*

國立中央大學資訊管理研究所\*

## 摘要

由於網際網路的發達，各種語言文字都可在網路上通行，也都能在網頁上展示。但在 DNS，網域名稱的支援上，卻缺乏對非英語系國家語言的支援，使得其他非英語系人口，無法以自己的語言替網站命名，本文提出中文域名在網站上相關的問題；其次討論現有各種作法的相容性；同時亦蒐集中文繁體字體轉換的相關研究，針對中文域名，網頁使用與線上繁體互換需求，配合繁體域名的登錄與對映方法進行解析，使得繁體域名網站可以方便地互訪。

本研究亦實作一個以網路代理伺服器為基礎的系統，Proxy-Web-DNS，可瀏覽中文域名網站及自動繁體互換，讓使用者方便的尋取與瀏覽。文中介紹整個實作方法，並與其他方法加以比較。本系統在架構上具有彈性，可隨標準調整其組成模組，使用上更具相容能力與方便性。

關鍵字：多國語文域名、網路代理伺服器、中文網址、目錄服務、中文 URL、Web-DNS、繁體互換、域名搜尋

## 1 緒論

隨著全球資訊網的普及，網站的數量也日益增多。使用者必須藉由網址 (URL) 來連結到欲瀏覽的網站，網址是由協定名稱 (scheme)、網域名稱 (hostname)、路徑名稱 (path)、及查詢 (query) 所組成 [1]。以往的網域名稱 (Domain Name) [10]，都是由英文的 LDH (Letter Digit Hyphen) 字元組成，如 www.edu.tw 代表了台灣教育部的全球資訊網網頁。由於網域名稱的支援上，缺乏對非英語系國家語言的支援，使得其他非英語系人口，無法以自己的語言替網站命名，所以 IETF 成立了 idn 工作小組來制定國際化域名的標準 [2]。從 1999 年至今，各種的建議不斷的被提出，至今還未定案。不過，現今已經存在許多的系統，以相容於舊有 LDH DNS 架構方式，提供 IDN 的註冊服務，如 i-dns.net [33] 的全漢字域名、VeriSign/NSI [32] 的中文.com 多國語文域名測試平台等。在中文部分，中國互聯網訊息中心 CNNIC [3] 與台灣網路資訊中心 TWNIC [4] 也分別在 2001 年初開放了中文網址的註冊服務。

由於中文常用的內碼又分為繁體與簡體，故同一個網址會以繁體兩種以上不同的形式呈現。而現在的瀏覽環境裡，尤其是繁體不同環境要互訪時，必須將層層關卡突破後，才有辦法讓以中文域名為網址的網站內容正常地完全呈現。

其次習慣繁體字的人，並不習慣閱讀簡體字，反之亦然。所以最好能夠有兩種版本的網頁內容，可是大多數的網站並未提供。中文域名固然易記，但中文域名的輸入比較困難，常需要輸入法的配合，而多半繁體輸入

法無法輸入簡體字，反之亦然。故用中文域名的部分配對 (Partial Matching) 列出各種相關結果以供點選的搜尋功能是需要。

本研究是由繁體中文域名出發，針對繁體網址互訪時的會發生的問題加以分析研究，探討相關可行解法；其次是中文網頁內容的繁體互換支援等，最後實作一個協助系統，讓使用者在使用中文域名時，從繁體域名輸入，名稱查詢到繁體內容顯示能連成一氣，使用上更為方便。

在第二節裡，我們討論相關研究；第三節分析一個中文網站瀏覽過程中可能會遭遇的問題；第四節提出本協助系統的的架構及設計考量；第五節展示系統實作；第六節是本系統與其他系統的比較；最後是結論。

## 2 相關研究

### 2.1 網域名稱系統 (Domain Name System)

網域名稱系統的架構是階層式的分散式資料庫，由許多 DNS server 為節點組成整棵樹。主要的功用是將名稱轉換為 IP 位址。例如：www.ncu.edu.tw 對應到的 IP 位址是 140.115.17.125。每個節點都代表了一個網域 (Domain)，在網域下面還有子網域 (Subdomains)。舉例來說 ncu.edu.tw 就代表屬於 tw 之下 edu 子網域下的一個節點。而上層的網域將某些子網域交給其他 DNS 管理的動作是授權指派 (delegation)。

為了將名稱轉換為 IP 位址，每個主機名稱是用 FQDN 所表示。FQDN (Fully-Qualified Domain Name) 完整指明的網域名稱，如 “www.ntpu.edu.tw.” 從左到右至最右邊的 “.” (代表最頂層 root domain name) 一起完整指明代表的網域名稱，它代表 Internet 上唯一的主機名稱識別。這個名稱被正確的登陸在 DNS 伺服器上時，任何人都可以透過任何一台 DNS 伺服器查詢到這個名稱。域名的查詢過程稱作名稱解析 (Name Resolving)，用戶端 (Client) 會配有解析器 (Resolver)，通常是以函式庫的方式放入作業系統中，各類應用程式經由呼叫這個解析器函式庫可以很容易地向 DNS server 進行遞迴 (recursive) 或反覆 (iterative) 查詢。遞迴查詢通常是解析器將查詢工作送給 local DNS，然後由 local DNS 不斷地查詢到答案為止。反覆查詢是 DNS 對其他 DNS 發出詢問時，都只問到下一步的線索，查詢端再根據線索自行找答案。大部分的 DNS 都可接受兩種查詢方式，但考慮負載問題，高層的 DNS 只接受反覆查詢。

### 2.2 中文內碼介紹

內碼為資訊處理系統內部一種基本表達形式。做為字符儲存、處理等用途。英文系統也有內碼，以一個位元組 8 個位元表示，共有 256 種組合。英文字母與常用的符號加起來，也不超過 127 個，7 個位元就可以表

示。後來 IBM 將 128 之後的位置都編上圖形，所以英文的內碼是 0-255，共 256 個，用 2 的 8 次冪就能包涵所有的字元。中文字的字數較多，以字數較少的大陸 GB 字集來說，約有七千個早已超過了一位元組的 256 個組合。所以中文採用了雙位元組的編碼方法。兩個位元組的組合數是 2 的 16 次冪，即 65536。用這種方法，理論上最多可以收編六萬五千多個中文字。

中文常用的內碼有兩種，一是台灣地區使用的是繁體中文 BIG5 碼，另一種是大陸地區常用的 GB 碼。這兩種內碼都是使用雙位元組的編碼方案。BIG5 碼是通行於台灣、香港地區的一個繁體字編碼方案，其第一位元組的值在 0xA0~0xFE 之間，第二位元組在 0x40~0x7E 和 0xA1~0xFE 之間。因此，其第一位元組的最高位元是 1，第二位元組的最高位元可能是 1，也可能是 0。

GB 碼是國標的拼音簡稱，是大陸地區國家標準碼的縮寫。常簡稱為 GB 碼或國標碼。主要是簡體漢字的編碼，通行於中國大陸地區與新加坡等地。GB 碼的編碼範圍在 0xA1~0xFE 之間，兩個位元組的最高位元都是 1。

GBK [8]為中國大陸的國家標準擴充碼 GB18030 (K 意指擴充，是擴的漢語拼音第一字母)。GBK 規範中收錄了 ISO 10646.1 中的全部 CJK 漢字和符號，並有所補充。GBK 亦採用雙位元組表示，首位元組在 81~FE 之間，尾位元組在 0x40~0x7E 與 0x80~0xFE 之間。它包含了繁體與簡體的字碼，所以同一個字，會有 GBK 簡體與 GBK 繁體之分，如 GBK 繁體「燈」與 GBK 簡體「灯」。

表 1: 中文內碼碼別範圍表 (\* 表示與 ASCII 範圍重疊)

碼別	第一 Byte 範圍	第二 Byte 範圍
ASCII	0x21~0x7E	
BIG5 (CP950)	0x81~0xFE	0x40~0x7E * 0xA1~0xFE
GB2312	0x81~0xFE	0x81~0xFE
GBK (CP936)	0x81~0xFE	0x40~0x7E * 0x80~0xF1

表 1 是將各主要內碼的範圍列表，顯示中文內碼的第一位元組的第 8 bit 為 1，而第二 byte 的則不一定。BIG5 碼便與 ASCII 範圍重疊。重疊時，一些特殊符號如“@”、“|”、“}”等等，在應用程式沒有對內碼的第一 byte 做進一步辨識的檢驗以判定是雙位元組的中文字時，容易被視做一般英數符號而錯誤解讀。

ISO 10646 國際標準化組織 ISO 公佈的一個編碼標準，即 Universal multiple-Octet Coded Character Set (簡稱 UCS) [6]。ISO 10646 的正規形式是有 32 bits，劃分為四個 8 bit。這 4 個八位元，由左而右命名為群八位元 (G-octet)、面八位元 (P-octet)、列八位元 (R-octet) 和格八位元 (C-octet)，分別代表編碼結構中的群組 (group)、字面 (plane)、列 (row) 與格 (cell)。當群八位元和面八位元的值都為 00h 時，稱為「基本多語文字面」(Basic Multi-lingual Plane, BMP)。當電腦系統只使用 BMP 的字元碼時，可以省略群八位元和面八位元，因而將字元碼由 32 個位元縮短為 16 個位元，稱為 ISO10646 字元碼的基本面形式 (可簡稱為 UCS-2)，其實也可視同於 Unicode。UCS-2 幾乎包括世界上各種語言的書面形式以及符號的編碼體系。其中的漢字部分

稱為 CJK 統一漢字 (C 指中國，J 指日本，K 指韓國)。而中國部分，包括了源自中國大陸的 GB 2312、GB 12345、「現代漢語通用字表」等法定標準的漢字和符號，以及源自台灣的 CNS 11643 [9] 標準中第 1、2 字面 (基本等同於 BIG5 編碼)、第 14 字面的漢字和符號。

UTF 為 “a UCS (or Unicode) Transformation Format” 的縮寫，UTF-16 意即把原為 32 位元的 ISO 10646 字元碼轉換為 2 或多個 16 位元的 Unicode [7]。由於現在是 ASCII 世界而非 Unicode 世界，所以當某一個 Unicode 離開了以 Unicode 為內碼的作業系統或應用程式之後，馬上會被網路設備或應用程式、通訊協定切成兩個 8 位元的位元組，並檢查是否有 c0 控制碼，至少數千個 Unicode 將因此變的殘缺不全。因為 Unicode 編碼為連續編碼，不再避開 c0 及 c1 控制碼區 (c0 控制碼區指的是編碼為 0-31 的 32 個控制碼，c1 控制碼區則是意指 128-159 的 32 個控制碼)，而 c0 控制碼是在 8 bit 字元結構中具有控制功能的。為了能讓 Unicode 和 ISO 10646 字元碼能安然通行於 ASCII 世界，Unicode 協會提出 UTF-8 [36] 編碼方式以解決問題。UTF-8 意即把原為 32 位元的 ISO 10646 字元碼或原為 16 位元的 Unicode 轉換為多個 8 位元的位元組。

## 2.3 簡繁互換

由於簡體字的範圍比較小，故繁體轉簡體時，多為多對一轉換，但是由簡體轉換回繁體時，則會遇到許多問題，在「漢字簡繁轉換的複雜性和陷阱」[29] 中有詳述，本節整理節錄其要點部分，並舉例說明之。

簡繁互換的主要問題在於以下四點：

- 1、很多簡體字已經簡化的面目全非，讓人無法聯想到原來的繁體字，如：簡體字「发」與繁體字的「髮」，繁體的「徵」與簡體的「征」。
- 2、繁簡體非一一對應，例如：簡體的「发」對應到繁體字可能是「發」，或是「髮」，必須根據上下文決定哪一個正確。
- 3、簡體對繁體很多情況是一對多的對應，如：簡體「干」對應到繁體的「幹」部、「干」涉、「乾」燥，端看上下文決定。
- 4、繁簡字內碼不同，簡體字使用 GB，繁體使用 BIG5 碼，兩者並不相容，有些字碼沒有相對應的字碼。如：簡體 GB「计」便沒有相對應的 BIG5 碼。

## 2.4 多國語言域名系統 (Multilingual DNS) 簡介

### 2.4.1 背景

網際網路的流行風潮已席捲全球，在世界各國利益競爭的趨勢主導下，讓網際網路賴以運作的 IP 地址及網域名稱已成為「全球重要公共財」之網路資源。目前網際網路上負責指定名稱與號碼的機構稱為 Internet Corporation for Assigned Names and Numbers (ICANN)。由 ICANN 發起的多國語言網域名稱服務系統 (Multilingual Domain Name Service, 簡稱 ML-DNS) 之相關活動與議題，引起許多非英語使用國家注意，而

IETF 也因此成立了 IDN WG [2] 工作群組來討論制定相關的標準。

要建置一個多國語文 DNS 的技術方法大有三種，第一種是直接讓 DNS 能夠正確的傳送 8 bit 字元，DNS 協定本身就可以傳送 8 bit 字元，只是因為相容性的關係被限制在只能傳送 LDH 的範圍。這種方法的好處就是在 server 端做更動，能夠快速的建置多國語言名稱的環境。第二種方式是由協定本身著手，讓協定本身能夠傳送額外的多國語文域名資訊，如 EDNS [18]、DNSII [19] 等。其的優點是名稱可以完全比對不混淆，保有原來 DNS 的效率。但是必須從 Server 與 Client 端到所有的應用程式都必須跟著修改。最後一種方法是維持與現有的 LDH DNS 相容，所有的多國語文域名名都在 Client 替換成 ACE (ASCII Compatible Encoding) 域名再送出查詢 [38][40]，許多的 ACE 方式被提出，如 RACE [17]、LACE [20]、DUDE [37] 等等。採用此法的優點是完全相容於 ASCII LDH DNS，server 端不需要做任何的更動；屬於 tunnel 的技術。缺點則是使用者必須下載軟體才能使用多國語文域名，而下載的 binary Client 程式，有可能不再經過 DNS tree 去查詢，改轉往某些特定的 server 查詢，違反 DNS 授權的唯一性。而且此時的域名的顯示格式 (presentation format) 已經不再與傳送格式 (transportation format) 相同了，而原型傳送是原有 DNS 的一大特色。表 2 是三種方法的比較。

表 2: 三種 ML-DNS 方式比較表

ML-DNS 方法	優點	缺點
修改為 8 bit Clean	Windows 使用者端無須下載軟體。server 端部分更動即可快速建置。	不完全相容於 ASCII LDH DNS。
更改 DNS 協定	維持效率與完整的支援多國語文。	從 server 到 client 所有的程式必須更新，非一夕可及。
將多國語文域名轉為 ACE 再查詢的方式	完全相容於 ASCII LDH DNS。	Client 下載轉碼，病毒，檢驗正確性及版本更新問題。傳送格式與呈現格式不相同。

## 2.4.2 DNS 多國語文化的問題 無版本協商功能

DNS 的協定中，並沒有提供版本協商的功能，新增的功能或協定的更改，不易在現有的系統上運作。然而多國語文便需要對原有的 DNS 協定做修改，但是原本的協定缺乏版本協商功能，使得多國語文的 DNS 與原有的 ASCII DNS 不易完全相容並存。例如多國語文域名的查詢經過一台不知新舊的 LDH ASCII DNS 時，雖 DNS message 格式皆相同，但欠缺版本的回應協商，不能保證是否的透過它能正確地查詢到所要的資料。

### 欠缺碼別標記

原有 DNS 的協定，對於域名只限制在英文及數字的 ASCII 範圍內。故並沒有什麼碼別標記的問題。但是

在多國語文 DNS 環境時，各種不同的語文，如果使用不同的內碼利用 DNS 傳送，但卻沒有欄位可以標記碼別，就只能由域名本身來的一些特性來判斷，如：字符的數碼範圍、出現的頻次與碼別；某些 ccTLD 域名下收到的查詢資料，其碼別與 ccTLD 有相關性；字符的出現與前後文有關等，但都屬於協助碼別型態快速判斷的輔助而已。如果有碼別標記，則可以避免多碼紀錄並存的記載，此為每一種碼別域名都建立各自對映的紀錄，在管理人員的辨識上較方便。同時多國語文的 DNS 便能透過域名轉碼器，正確的判斷來源碼別，進而轉碼查詢。

### “\”字元的處理錯誤

在最廣泛使用的名稱伺服器軟體 BIND [5] 的實作中，為了 zone file 裡的接續符號 “\line-feed”，跳格 “\tab”，空白 “\blank” 及 Email 帳號 “abc.xyz@foo.bar” 裡的 “\.” 引進了 “\” 的處理規則。但中文字碼的第二位元組，有 8 bits 的，也有位於 ASCII 範圍內的。於是就會碰上中文內碼中第二 byte 的 “\” 字元與英文字母不分大小寫的問題。

嚴格說，“\”是跳脫字元 (escape code)，標示特殊狀況下的某些特定符號需特定處理，但 BIND 實作的測試資料可能因為只有 7 bits data，對 8 bits data 並沒有特別測試，這就造成 “\” 符號錯誤的原因。另一方面，Microsoft Windows 的 resolver 算是 8 bits enable，但過去為了與「有錯的」BIND DNS server 相容，有些模組會將單個 “\” 刪除，有些則會對中文的 “\” 補加 “\”，各有不同的版本。這也是另一個「回溯相容」問題。  
域名不分大小寫

英文域名是不分大小寫的，如 WWW.YAHOO.COM 與 WwW.YaHoO.Com 被視為相同。而多國語文內碼中有，若無法辨識多位元的多國語碼與 ASCII 發生的重疊，如 BIG5 中的「象」(0xB6 0x48) 與「逸」(0xB6 0x68)，0x48 是 ASCII 裡的大寫 H，而 0x68 是小寫 h。在無中文標示或 multi-byte、Multi-lingual 的辨識能力下，這兩筆資料會被 ASCII LDH DNS 視為相同。比對的方法是要能夠辨識這是雙位元組的多國語文資料，取出比對時以一個中文字碼為單位來比對即可。

問題來自 DNS cache。在不分大小寫的情況下，其記憶體的存放位置是相同的，如果中文 DNS 與純英數 DNS 混合使用，純英數 DNS 查詢中文域名後的 cache 就會大小寫不區分，在扮演 Local DNS 的角色時，對查詢的中文域產生不正確的回應。

## 2.4.3 目前發展中的中文域名系統

由於多國語文的需求，以 (1) 直接使用本國域名顯示，(2) 域名的搜尋便利性，(3) 兼顧舊有 ASCII LDH DNS 相容性為主要的考慮。所以多國語文的域名使用問題，大致走向 “多層次的域名系統”。近來 (2001 年 6 月) 的 Internet draft 中，「以搜尋為基礎的 DNS 存取模型」 [27]，提出一個三層次 (layer) 的架構來提供 DNS 的域名搜尋功能，與台灣目前發展中的中文域名系統的多層次架構非常相似。

多國域名系統可分為三個層次，第一層是目前的 ASCII LDH DNS，提供英數域名的查詢功能，第二層是 ML-DNS，提供多國語文域名對照英數域名 [28] 及 IP 位址的對應，第三層才是提供域名搜尋的 Web-DNS 搜尋系統。以實例來說，第一層是舊有的 ASCII DNS，也就是原本的 www.ncu.edu.tw 對應到 140.115.17.125。第二

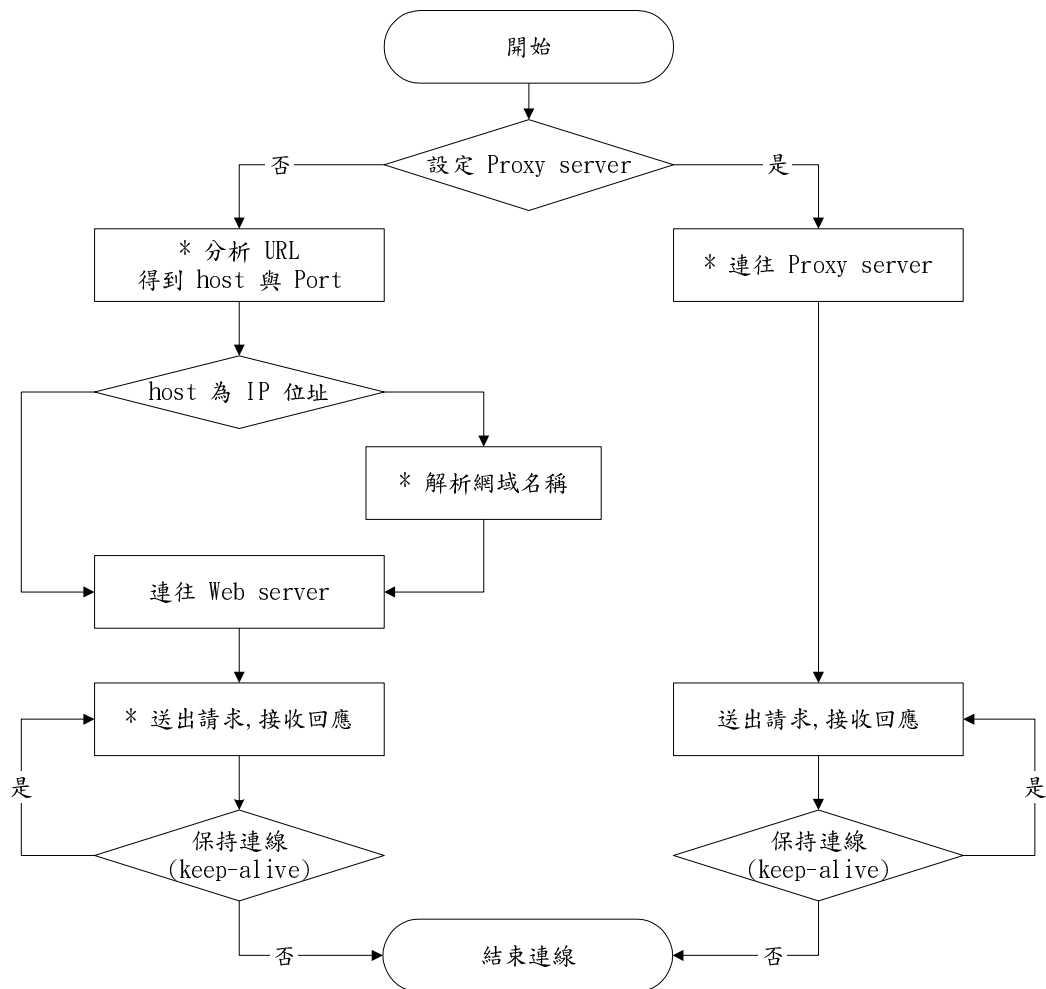


圖 1 網頁瀏覽過程 (\*為中文網址加入後可能會出問題的部分)

層是 ML-DNS 層，“國立中央大學.tw”對應到唯一的英數名稱 www.ncu.edu.tw，“國立中央大學.台灣”也對應到可鑑別的唯一名稱 www.ncu.edu.tw，同一層級“中大.tw”是指定給“dns.中大.tw”這台 DNS 管理。而“中大.中大.tw”是“中大.tw”zone file 的一筆記錄，假如“中大.中大.tw”是被視為可辨識唯一鑑別字符 (Unique Identifier)，就可以有對應到最終的 IP 位址 140.115.17.125 這筆記錄。第三層級，是將“國立中央大學”取代為“<http://www.ncu.edu.tw/>”或“台灣中大”取代為“<http://中大.tw/>”的 Web DNS。

目前台灣及大陸所提供的中文域名系統是中英兩用 DNS，在層級上屬於第一層與第二層。其 zone file 中存在 BIG5、GB 及 UTF-8 等多種表示法的域名資料。其中 BIG5 與 GB 的繁簡對應，是使用者在註冊登錄時原始登錄的域名為主要域名，再以「買繁送簡」或「買簡送繁」的方式，由使用者對繁簡對映中一對多的字體部份自行選取一個對應的繁簡網域 [39]。為了域名搜尋輔助，TWNIC 也利用 local root DNS 提供 Error trap 的輔助，將查詢不到的中文域名指向某一台 Web server。該 Web server 提供域名的 partial matching 搜尋功能，再由使用者點選其中的結果，這是屬於第三層級的 Web-DNS。

多層級的中文域名系統可以兼顧回溯相容 (Backward compatible) 及前推相容 (Forward compatible)。由於目前大多數的 AP server 都不認得 ML 域名，所以我們必須透過 UNAME [28] 來顧及回溯相容性。UNAME 是 Unique name/Identifier 的意思，被拿來

做唯一性區分辨識之用，在使用上就不能讓該 identifier 違背可辨識性與唯一性。它可以是個不受轉碼影響的英數名稱、ACE 名稱，甚至唯一的亂數也是可以的。使用者透過第二層的 ML-DNS 將中文域名「替換」為 UNAME，再以 UNAME 去第一層的 ASCII DNS 查詢出 IP 位址，以替換之後的名稱來稱呼原有的 AP server，AP server 可以辨識這個替換後的名稱，就可以相容於原有系統。未來當 AP server 可以辨識 ML 名稱之後，多國語文名稱在唯一可辨識的條件下，也是可以當 Unique-name/identifier，這時候使用者透過第二層 ML-DNS 就能從 identifier 直接拿到 IP 位址，以 ML 名稱去存取 ML-aware AP，就能兼顧未來需求的前推相容。這些有辨識中文名稱的能力的 Web server，中間的 Proxy、firewall 等，我們稱之為 ML-aware Application。目前的實驗系統是以 CNAME 暫當 UNAME 來使用。但 CNAME 有很多限制不能與其他 RR 紀錄並存。

### 3 問題分析與相關解法

本節主要探討中文域名在瀏覽網頁時的問題，特別是因繁簡體不同而衍生的困難。

#### 3.1 使用中文域名繁簡互訪的問題

我們以圖 1 說明以中文域名瀏覽網頁所可能產生的問題，整個瀏覽步驟可分為兩種情況。一是不透過代理伺服器，另一種是透過代理伺服器。透過代理伺服器時，URL parsing、中文域名解析、送出請求都是由 Proxy server 代理。

## URL 分析與名稱解析問題

首先，使用者必須在瀏覽器的「位址列」輸入，或點選一個中文的 URL。當瀏覽器收到一個非英文的網域名稱時，瀏覽器可能是直接將它阻擋掉、或經過某些轉換才送出 DNS 查詢。在遇到中文的 URL 時，主要可分為下列幾種行為：

- (1) 直接送出：按照原形送出透過下層的解析器或中間的 proxy server 向 DNS server 查詢。
- (2) 轉換成 UTF-8：目前的 browser 實作上來說，這是麻煩最多的部分。它通常牽涉到瀏覽器應用程式與作業系統核心之間錯綜複雜的溝通轉碼。有些情況下，字串被轉換了兩次，有些 ending byte 被捨棄掉了，有些轉換成另一種形式之後，便無法恢復。有的送往核心前會轉成一種格式，送上 DNS 查詢時又以 8 bit 原形送出。如將繁體中文版的 Windows 及 IE 5.x，將「永遠以 UTF-8 傳送」選項打開，非 ASCII 字元就無法以原形送出。
- (3) 轉成某種形式的 ASCII 字串：就如同轉換成 UTF-8 的問題一樣，不同的 browser 有不同的行為。如 Netscape 6.0，會將 8bit 字元轉換成 %HH 的表示法，送給 proxy server。而 IE 就會不會轉換成 %HH 的格式。
- (4) 阻擋不送出：傳統的 BIND 程式在 UNIX 系統建立的解析器，都是不允許 8 bit 資料通過。即使是 Windows IE Browser 也有很多碼值範圍檢查後阻擋的現象。

更多詳細的測試及問題可以在 OpenIDN 的網頁取得 [11]。

在送出 DNS 查詢之前，URL 必須先被分析成 Host 與 port 等部分。URL 的 Host 部分，本來是依照 DNS 協定的規則，只通行 A-Z、a-z、0-9\_，中文域名加入之後，URL 中便會有非 ASCII 的字碼。由於大部分的字元剖析函式 (parse routine 或 function) 在設計之初，並沒有考慮到 multi-byte 的問題，故中文字碼中若有“@”字元時，如「一」、「作」的第二位元組，就會被解釋成 mail URL 或 ftp URL 中的帳號與主機的分界字元。在瀏覽器解析之後，送出的查詢就少了“@”前面的部分。如“<http://統一大學.tw/>”就成了“<http://大學.tw/>”。還有些瀏覽器 (如：Opera 5.1) 會先警告說這是無效的 URL。目前針對 Microsoft IE 5 的解決方法是在含有“@”的中文域名 URL 前面加上兩個“@@”。

英文域名是大小寫不分的。若不經分辨就將域名部分全部轉為小寫，則有些中文域名便送錯字了。如大「象」便成了大「逸」。而大部分的瀏覽器都會將域名轉為小寫才送出，所以即使利用 URL escape 方式，將中文域名改成 %HH 的 ASCII 表示法，遇上與 ASCII 範圍重疊的碼，還是行不通。

### 網頁伺服器及虛擬主機

由文獻探討中，我們知道以域名為基礎 (name-based) 的虛擬主機，是同一個網站伺服器，以域名來區分不同的網頁的方式。然而由於這種方式，變得

網頁伺服器對於 Client 如何稱呼它敏感。就以廣泛使用的 freeware apache httpd [13] 來說，它便有一段程式碼檢查請求的網域是否正確。這段檢查會將含有非 ASCII 字元的域名稱呼視為錯誤 (malformed)，使得請求被拒絕。同樣的情況，也發生在位於 web server 與 client 中間的 firewall、proxy cache 等上。所以現存的解法中，必須將 web server 與 proxy cache 做小部份的修改，放寬對 hostname 檢查的限制，才能讓 browser 以中文名稱請求回應。

一個網站有了中文域名的時候，存取該網站的名稱也就變多了。如“台灣大學.tw”是繁體的表示法，也有簡體的表示法“台湾大学.tw”，也有英文的表示法：www.ntu.edu.tw 與 w3.cc.ntu.edu.tw，甚至 IP 位址的表示法：140.112.8.130。這幾個表示法都代表同一個站。在這一台網頁伺服器上，若只有一個網站的時候，那不論如何稱呼，都代表同一個網站。但如果它是一台提供多個網站網頁的虛擬主機伺服器的話，加入中文的虛擬主機之後，就必須增加中文域名的辨識能力與名稱設定。

### 中文網址繁簡互換問題

當 Web server 認得中文域名的表示法之後，我們便能夠以繁簡中文域名去瀏覽該網站。但如果以繁體域名去瀏覽內容為簡體的網站時，或簡體 GB 碼域名來訪問繁體內容的網站時，我們可能會看到因為圖片沒有下載完整而殘缺不全的網頁，或甚至看不到網頁。這是因為 Web browser 會依據目的端內容碼別，將網址自動轉碼，但其碼值範圍又非本身語碼範圍所致。

舉例來說，在輸入繁體 BIG5 域名訪問簡體網站時，如“<http://中山大學.cn/>”，我們可以從 DNS 上查詢到這個站的 IP 位址，很快的找到網頁下載。該網頁內容是以簡體中文表示的，而且網頁中的圖片都是以相對路徑存放的。當瀏覽器要存取該圖片時，會根據所查詢網頁的編碼方式 GB，隨之自動轉碼為“<http://笈利湮.cn/xyz.gif>”或“<http://笈利湮惺.cn/xyz.gif>”。如果因為這個繁簡轉換與註冊時商定好的字碼不同，則無法查得該轉換後對應網址的 IP 位址時，該網站圖片就會因找不到 IP 位址而無法顯示。還有些情況，自動轉換之後的碼因為無法通過轉碼模組的碼值範圍檢查 (range check) 而產生“/?”的亂碼，則無法抓取到正確的圖片。如鄭州大學的「鄭」，轉成簡體之後是「郑」(0xE0 0x8D) 第二 Byte 0x8D 超出 BIG5 的範圍，就是最好的例子。

另外，Windows 的輸入框到底應不應該隨著碼別轉換，應該如何轉，字碼何時要視做相同，是有關名稱準備 (NamePrep) [16] 的問題，目前尚無標準定案。

### 繁簡網頁內容顯示

台灣的網站大多都是以繁體為主的，而大陸的網站大多數都是以簡體為主。少數的網站有同時提供兩種版本的網頁顯示。雖然只要瀏覽器及字型支援，在繁體環境下，也可以顯示簡體的網頁。不過，繁體的使用者看不習慣簡體字體，習慣簡體字的使用者，不見得看得懂繁體字。所以繁簡互換是必要的，將網頁內容繁簡互換，可以在三個地方著手，一是在用戶端，二是在 Proxy server，三是在 Web server 端。

- (1) 在用戶端的解決方案最常見，通常是由使用者安裝應用程式，該應用程式提供了繁簡字型及轉碼功能，繁簡輸入法等，可以將視窗上出現的字碼直接做繁簡互換的動作。如南

極星 [14]、RichWin [34] 就是在 Windows 平台下提供中日韓文顯示及輸入系統。

- (2) 在 server 端提供繁簡轉碼的方式，有利用一張字對字的繁簡對照表，以其中一種碼別為主，轉換產生或動態產生另一種版本的網頁。也有利用專門的繁簡轉換伺服器，或 reverse proxy 的方式，專門對某個站的另一種版本語言做轉換服務，如「看漢在線出版伺服器」 [31]、「信使繁簡通」 [35] 等就是這種性質的系統。目前還有的是利用單一 CGI 程式，對單一網頁轉碼，我們將之歸類在 server 端轉碼。如：網擎 Openbar [15]，蕃薯藤的繁簡互換服務 [16] 等。大都是以使用者鍵入欲轉換的 URL，讓 CGI 程式代理抓取，轉換後顯示。Openbar 利用其下載程式，提供了按鈕，方便使用者將目前瀏覽 URL 送往特定 CGI 程式繁簡互換。
- (3) 由代理伺服器來提供繁簡互換的系統，也是非常方便的。使用者只要將 proxy 設定在可以繁簡互換的 proxy server 上，即可一路用同樣的碼別瀏覽網頁。但目前並無公開服務的繁簡互換 proxy server。

### 3.2 問題總結與相關解法

總結目前中文域名繁簡互訪瀏覽環境中，存在的問題可分為以下幾類：

- 隨著中文域名帶來的問題：
  - (1) 中文 URL 解析問題：中文 URL 的問題是因為以往 URL 的 hostname 部分，並沒有非 ASCII 字元，於是中文網址中的中文含有特殊字元，如“@”或“\”，在程式不能區分中文字的情況下被視為英文，導致瀏覽器及 DNS 無法正確解析這些中文網址。
  - (2) 網頁瀏覽各組成成員有 Web Server、Proxy cache server 等，也因為對網址的非 ASCII 做檢查，而認不懂中文域名。另外是中文域名的 Virtual Host 問題，必須增加中文名稱的設定。
  - (3) 中文網址繁簡互訪時，由於網址會根據網頁內容碼別做自動轉碼的動作，導致網頁中的物件可能下載不全。
- 繁簡中文使用習慣：網頁內容繁簡互換問題。
- 協助輸入使用問題：中文網址輸入容易打錯，需要一些如網址搜尋的輔助功能。

針對中文網址帶來的問題，在 Web server 沒有全面性的認得中文域名時，目前的解法就是要透過第二層的 ML-DNS 使之回溯性相容。就是將中文域名取代為 UNAME，再用該 UNAME 的名稱去瀏覽。如“<http://湖南大學.cn/>”被其 UNAME 替換成“<http://www.hunu.edu.cn/>”。則繁簡網址的拜訪便不會有問題。

目前存在回溯相容的方式有 Client 端的解法、與 proxy 端「替換代理 server」兩種：

Client 端下載轉碼就是在 Client 端「替換」中文域名為英數符號 (ACE 域名)，並假設這些出示的英數域名都因相容而全被放行。它的問題如下：

Q1. 只有回溯相容，將不可能有未來的 8bit multi-byte 字元域名的 DNS。

Q2. 每一個 Client 端都要下載轉碼。

Q3. 轉碼表是建立在一個不確定的 UNICODE 聯盟的維護上，而且變動率不低。

Q4. 下載轉碼隱藏 tunnel「額外的轉向查詢」，不同產品來源相互攻伐不止。

Q5. 所有沿途相關 server 皆需「設定 ACE 域名」，而非不受影響的放行。

反之，替換代理 Server 可以兼顧前推相容，如果沿途的相關 Server 都已被設定且能辨識註冊的中文域名，替換為 ACE UNAME 就無必要。此時的做法如下：

M1. 「替換代理」server 不再居間替換，而是告訴 Client 逕自用中文域名通行。

M2. Client 端不再需要設定指向中間的「替換代理」Server，而是比照原有的英數 DNS 協定就可通行。

表 3 是 Client 端 ACE 轉碼及替換代理伺服器兩種方法的比較。

表 3: 回溯相容方式實例與優缺點比較

回溯相容方式	實例	優點	缺點
Client 端 ACE 轉碼	VeriSign/NSI [32]、I-dns [33]	不更改 BIND DNS server 軟體	下載程式的缺點 只有回溯性相容
替換代理 server	TWNIC 利用 proxy.pac 將中文域名導向替換代理 server。	可兼顧前推相容	只提供替換轉向，造成中文域名沒有顯示。
	本系統，以為「替換代理」server。	真正的 Proxy server，可以以回溯或前推相容方是代理詢問可涵蓋 proxy.pac 轉向方法	需要 Proxy server 的頻寬支援

關於繁簡網頁內容顯示問題，在 Client 端，Server 端與 proxy 端三種解法裡，以屬於 Proxy 端的解法最簡單而有效。從網站提供者的角度來看，他們不用再多準備另一版本的網頁或添購專用繁簡互換伺服器，而從使用者的角度來看，也不需要安裝一套功能強大的視窗應用程式。而繁簡內容互換在 Proxy 端即可處理。表 4 是三種方法的比較。

要提供使用者域名輸入的輔助，送往第三層 Web-DNS 搜尋，攔截點可以在 DNS 端、Client 端、以及 Proxy 端。Web Client 端攔截的 Keyword 系統，會使得所有無協定字串都通往特定的搜尋引擎查詢，並非只是搜尋域名為主。使用 DNS 端的方式，只有高層 (Top Level Domain) 的 DNS 適合提供這項功能，目前也只有 TW ccTLD 的 TWNIC 有提供。這些功能在 Proxy 上做輸入錯誤輔助，有免下載的優點，更可以配合其他的搜尋引擎，讓使用者更方便的能夠找到欲瀏覽的中文網站。三種方式的比較如表 5。

表 4: 各種繁簡內容轉碼解決方案比較

網頁內容繁簡互換解決方案	實例	優點	缺點
Client 端應用程式	南極星 RichWin	可提供視窗環境下多國語文的顯示，輸入。功能不只針對網頁。	使用者必須安裝應用程式。
Server 端 (或 reverse proxy 方式)	漢網系統 信使繁簡通	專做繁簡互換，效率及正確率高。能處理輸出入問題。	只對特定主機提供另一版本網頁。
	Cgi 程式	只針對某一頁或少頁的需求。簡便。	陽春、使用不易，最好有 browser plugin 配合使用。
Proxy 端	本系統	使用容易，對於所有中文網站皆可。	必須指定 proxy server。

表 5: 各種域名輸入的輔助的攔截位置與優缺點比較

域名輸入輔助	實例	優點	缺點
DNS 攔截	TWNIC 的 local root 攔截給 Web-DNS	完全達到域名的輔助 trap 的功能。擁有為正式 DNS 的中文域名註冊資料	利用“*”的做法，使得 ping 等測試失誤。
Web Client 攔截	3721.com [30] IE 與 Realname [21]	配合 keyword 系統，一步就可以找到對應的網頁 URL，而非網址。	必須下載，所有資料攔截到特定的搜尋引擎查詢。IE 免下載，但自訂功能不完整，查詢非 user 所自願。針對其 Directory 服務，而非域名。
Proxy 攔截	本系統	免下載，使用方便由 Proxy 端攔截，可專司 HTTP 查詢。配合其他搜尋引擎搜尋網頁。	必須設定 Proxy。

綜合所述，在 Proxy server 上進行輔助，是目前較單一且容易的方案，可一併解決使用中文網址會遇到的一系列問題、對繁簡內容作轉換、與 Web-DNS 輔助搜尋等一併整合，也較獨立於多數的 DNS resolver 與

server 軟體，可獨立發展。也可快速更換不同的轉換法與使用方式，也能指定有中文處理能力的 DNS server 而不必大幅修改 Client 端已有的設定。所以本論文提出並實作一個 Proxy-Web-DNS 的協助系統，來解決上述問題。

#### 4 系統架構

本輔助系統是以代理人伺服器為基礎 (Proxy-based)。主要是由 Proxy server、Web server 與 DNS 三個部分組成 (簡稱 PWD)，架構如圖 2。使用者是透過 HTTP 協定連上 Proxy 使用。Web 提供了使用者介面，DNS 則是當作域名輔助搜尋的資料來源及中文域名的解析。

#### Proxy 的功能

分為三個模組：一、是前端模組，二、是 Proxy 核心模組，三、是轉碼模組。

一、前端模組，其主要功能是：

- (1) 辨識使用者是否已經設定參數
- (2) 代理使用者查詢 ML-DNS 的 UNAME 資料
- (3) 根據使用者參數，決定中文 URL 取代方式，送往下一個 Proxy 核心模組，或著發出重導訊息。

使用者可以設定的參數主要有兩個，一是喜歡的碼別，繁體或簡體中文，或不轉換。

二、是中文 URL 的取代模式，詳細參數及功用如

表 6:

表 6: 中文 URL 取代模式說明

URL 取代模式	動作	適用
1、代理查詢	保持瀏覽器位址列顯示中文，但 Proxy 將之取代為 CNAME/IP 再代理抓取。	回溯相容，輸入端維持中文原形。
2、查詢轉址	同上，但將 CNAME/IP 取代後回應「轉向」訊息。	回溯相容，輸入端域名被替換。
3、基本代理查詢訪問。	保持中文網址的原形查詢訪問。	假設 Web Server 認得中文域名稱呼。
4、自動	中文網址查詢到 CNAME 時，則為代理查詢；查詢到 IP 時，則以中文網址原形去瀏覽。	假設有登陸 CNAME 的為回溯相容。直接登陸 IP 者為可辨識中文域名 server。

使用繁體網址拜訪大陸簡體網站時，適用 1、2、4 的設定。由於簡體網址的登陸較完整，都含有 CNAME，且大多數不認得中文的稱呼。拜訪繁體網址的網站時，適用 3、4 的設定。但有時還是會碰到 Web server 不認識繁體域名的情況，可以換成 1、2 的方式瀏覽。

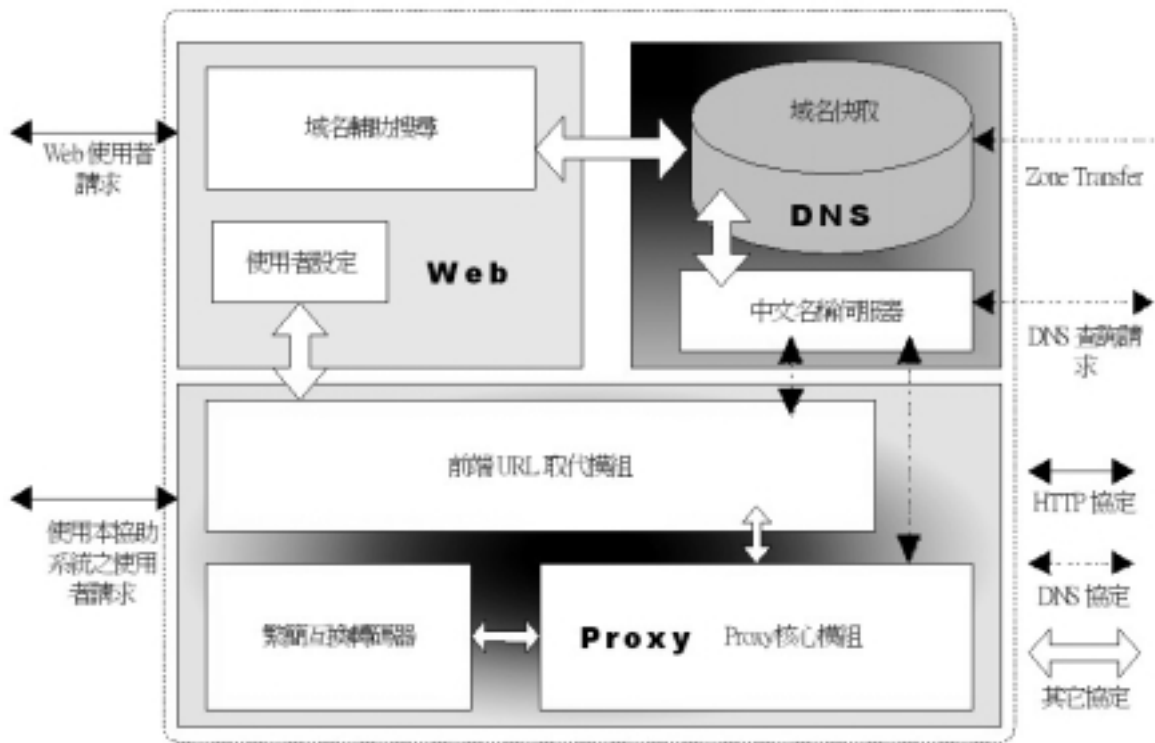


圖 2: 系統架構圖

此外，本模組還攔截 IE 5.5 位址列無協定，無 ". "，或無 "www " 的字串會轉往 "auto.search.msn.com " 的查詢的程式，使用者可以省去 "http://" 直接鍵入中文域名，進入本系統的輔助搜尋功能。由於可以攔截到完整字串，我們便在上面加了一些輔助，如 "telnet 中文域名 "、"mail user@中文域名的功能 "。我們透過 Web-DNS 的查詢替換，將中文域名替換成英數 CNAME、或 MX RR，達到便利使用的功能。另外我們也提供搜尋引擎的使用輔助，如在位址列鍵入 "g 昆蟲記 "，則此命令將會攔截被重導至 "http://www.google.com/search?q=昆蟲記"。

Proxy 核心模組主要工作代理使用者去存取 Web 資源，並提供快取功能。它可以直接存取遠端的 Web server，也可以與其他的 Proxy peer，共享快取區。轉碼

模組會依據使用者參數，決定轉換目的碼別，並且分析網頁的來源碼別作是否轉換的依據。透過此模組，網頁內容將會繁簡互換。

### Web 的功能

Web 主要有兩個功能，一是提供使用者設定介面，二是提供 Web-DNS 的域名搜尋輔助功能。搜尋的請求是由 Proxy 向 DNS 查詢不到中文域名時轉向而來。搜尋的順序是先查詢 DNS 再查詢資料庫。輔助功能的搜尋提供了 AND "；" 與 NOT "~" 的運算功能，可以使搜尋的範圍縮小，如使用者可以輸入 "苗栗;國小;南 " 很快的找到 "苗栗縣縣立南庄國小 "。另外，此介面還提供將搜尋字串轉向到其他搜尋引擎的功能。以方便使用者查詢域名以外的目錄或全文檢索服務，圖 3 為本系統的搜尋結果圖。

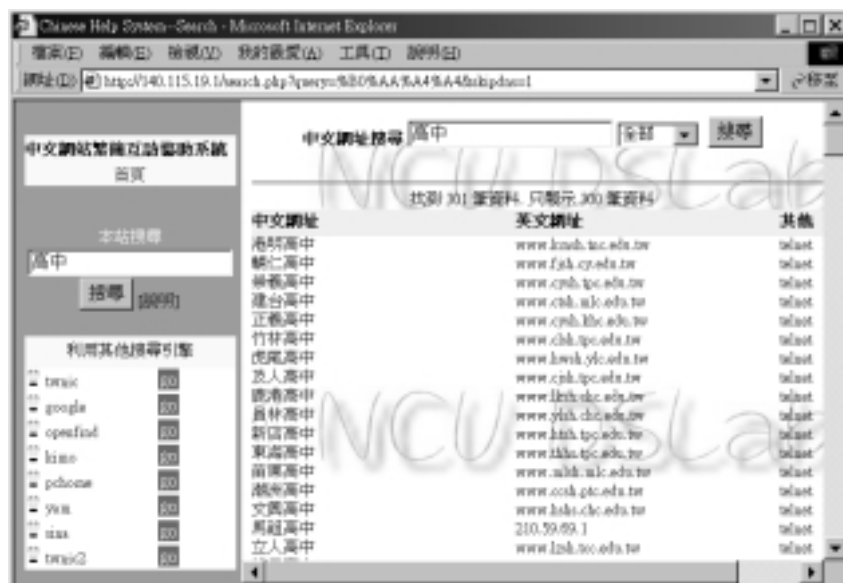


圖 3: 中文網域搜尋結果





圖 4: 本系統的設定畫面

另外我們也在 Web 介面提供 URL 取代 server 的重導功能。使用者若不想更改 Proxy 設定、而又無可使用中文網址的環境時，我們提供另一種選擇。直接利用 PWD 的 Web 介面。如 User 使用 ”<http://pwd.ncu.tw/>中央大學.tw“ 時，PWD 將會到 DNS 中查詢，將 ”中央大學.tw“ 替換成他的英數域名 ”[www.ncu.edu.tw](http://www.ncu.edu.tw/)“，回應使用者重導至 ”<http://www.ncu.edu.tw/>“。

### DNS 的功能

DNS 本身是一台中英兩用的 ML-DNS，主要功能有兩個：一是提供中英文網址的解析，二是當其它 ML-DNS 的 Cache DNS，將域名資料蒐集起來，供給 Web-DNS 的搜尋使用者介面使用。

## 5 系統實作

本系統安裝於一部個人電腦，其具有雙 Pentium III 800Mhz CPU，512M 主記憶體，100M 乙太網路卡。作業系統為 Linux 2.2.17，中文 DNS 使用 cBIND 8.2.3 版本，它改自 BIND 8.2.3 [5]，支援中文域名環境。並且由 .tw 的 DNS 授權 zone transfer 中文域名 zone file。

Web 部份使用 apache 1.3.20 加上 virtual host 的 patch，將 hostname 的檢查限制拿掉，內嵌 mod\_php4 模組，並使用 Mysql 3.23.36 作為資料庫。本系統除了與 proxy server 相關模組外，皆以 PHP [22] 撰寫。Proxy 採用的軟體是 Squid [23]，擁有改寫 URL 的能力，藉由提供一個 redirector 的前端模組，Squid 可以讓所有的請求均透過 redirector，讓 redirector 決定是否改寫這個 URL。我們改寫一個能提供 squid 很完整過濾模組的 squid filter module [24] 來實作文件轉碼的功能。最後繁簡互換轉碼函式是使用 autoconvert 0.3.9 [25]，一套利用字頻判斷的智慧型繁簡互換程式。

欲使用本系統，只需三個步驟：(1) 若使用 IE，請關閉 URL 以 UTF-8 傳送。(2) 透過 Proxy 自動組態或手動設定，將 Proxy server 指向本系統，(3) 瀏覽網頁。在第一次尚未設定參數使用時，Proxy 接收到請求會自動轉往本系統的設定 Web 使用者介面，如圖 4。設定完成即可照正常的方式瀏覽。如將碼別設定在繁體，則簡體網頁會自動轉成繁體顯示，如圖 5。



圖 5: 繁體 <http://北京大學.cn/>

## 6 系統評估與比較

我們在繁體中文的 Windows 環境下，測試與 PWD 相關的系統及產品。相關的系統可分為兩類，一類是多國語文域名系統；另一類則是 Keyword 系統。Keyword 系統不是以 DNS 為基礎的目錄服務，故多半搭配 Client 端下載、攔截位址列送往特定的搜尋引擎尋找。這類系統比較擅長的是在名稱的搜尋，以一個關鍵字能夠查詢到相關 URL。

多國語文域名系統又分為兩大類，一是以 Client 下載轉碼成 ACE 再到 DNS 詢問的方式，以 "中文.com" 為代表。另一種是台灣與中國大陸採用的中英兩用 ML-DNS 環境。ACE 域名系統的 Client 端，通常只針對 http:// 之後的域名做攔截轉換動作，其它部分都不處理，甚至有的要配合將 URL 以 UTF-8 傳送的選項，否則就無法使用。CNNIC 與 TWNIC 的中文域名系統碰上繁簡域名，都得使用 URL 取代的方式。而 CNNIC 選擇 Client 端下載，而台灣這邊選用免下載的 proxy.pac 方式，與我們的 PWD 系統設計類似，都是屬於輔助 "中文域名.tw" 的系統。表 7 是本系統與其他系統的比較。

類別	代表公司/系統	使用方式	多國語文域名支援	名稱輔助
Keyword (Directory)	Realnames & IE	免下載	支援	實名資料庫搜尋 搜尋不到將域名轉成 UTF-8 而無法使用。
	3721.com	下載	無法使用	提供拼音等多種方式輔助。
Web Address	CDNR.com 網名.cc 多國語文.nu	免下載	支援	無、導向註冊畫面。
ML-DNS (多國語文.com)	VeriSign I-DNS.net Walid.com [26]	Client 下載	支援，轉成 RACE。使用 UTF-8	無名稱輔助
ML-DNS 中文域名.cn	CNNIC [12]	Client 下載	可支援，將中文域名轉成 CNAME	無
ML-DNS 中文域名.tw	TWNIC	設定 Proxy.pac	中文域名 CNAME/IP 取代，部分代理冒充。	Directory 搜尋 提供 partial matching 功能。
	PWD 本系統	設定 Proxy	中文域名 CNAME/IP 取代，代理	域名提供 partial matching、及搜尋其它 Directory 功能。

## 7 結論

本研究是由繁簡中文域名出發，針對繁簡網址互訪時的會發生的問題加以分析研究，探討相關可行解法。其次是中文網頁內容的繁簡互換支援等。最後實作一個以代理人伺服器為基礎的協助系統，能夠協助使用者在使用中文域名時，從繁簡域名輸入、名稱的查詢、繁簡內容的顯示，一路讓使用者方便使用。

發展中的多中文域名系統是不相容於原有的 ASCII DNS 的，在大部分的 DNS resolver 與 AP server 都還沒過渡到可辨認多國語文域名時，網頁的瀏覽就會發生問題。我們的輔助系統是可以兼顧中文域名的回溯相容與前推相容的中間過渡系統。Proxy 提供了中文域名的替換與繁簡內容互換功能，能夠解決繁簡中文網址互訪時的種種問題。而 Web-DNS 輔助系統，能夠讓使用者以部份的輸入，找到想瀏覽的中文網域名稱。

由於 Proxy 的位置為提供多國網址輔助的最佳地點，它不僅可以維持使用者習慣、兼顧 ML-DNS 的回溯相容與前推相容，而讓使用者及系統管理者以最少的更動達到最佳的效果。故在多國語文 DNS 的標準尚未定案前，或者新的規格標準尚未普及到所有的 AP 時，PWD 都是最佳解決之道。最後，本系統未來將朝向分散式的 PWD 發展，能夠與其他的 PWD 互通，讓域名資料更齊全，使得使用者更加方便使用。

### 誌謝

感謝行政院國家科學委員會支持，計畫編號 NSC 90-2213-E-008-045.

### 參考文獻

- [1] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 2396, August 1998.
- [2] Internationalized Domain Name Working Group of the Internet Engineering Task Force (IETF), <http://www.ietf.org/html.charters/idn-charter.html>
- [3] 中國互聯網絡信息中心, <http://www.cnnic.net.cn/>
- [4] 台灣網路資訊中心, <http://www.twnic.net.tw/>
- [5] Internet Software Consortium, BIND "Berkeley Internet Name Domain", <http://www.isc.org/bind/>
- [6] ISO, ISO/IEC 10646-1993 (E) Information technology - Universal Multiple-Octet Coded Character Set(UCS) - Part 1: Architecture and Basic Multilingual Plane, Geneva, International Organization for Standardization, 1993.
- [7] The Unicode Consortium, <http://www.unicode.org/>
- [8] 中文資料找尋與資料彙編, 參考資料 GBK 碼規範 <http://www.haiyan.com/stealk/navigator/b5home.htm>
- [9] 國家中文標準交換碼, <http://www.cns11643.gov.tw/web/index.jsp>
- [10] P. Mockapetris, "Domain Names - Implementation and Specification", RFC 1035, Nov. 1987.
- [11] Edmon Chung, David Leung, Jim Lam, Wilson Chow & Ken Lee, OpenIDN, "Preparation for Migration of Multilingual Domain Names and Email Addresses", <http://www.openidn.org/www10.html>, Mar. 2001.
- [12] CNNIC 中文通用系統用戶端下載程式, <http://www.cnnic.net.cn/cdns/clientexp.shtml>
- [13] Apache HTTP server project, <http://httpd.apache.org/>
- [14] 南極星軟件公司, <http://www.njstar.com/>
- [15] 網擎 OpenBar, <http://www.openbar.com.tw/>

- [16] Paul Hoffman, IMC & VPNC & Marc Blanchet, ViaGenie, "Preparation of Internationalized Host Names", <http://www.ietf.org/internet-drafts/draft-ietf-idn-nameprep-03.txt>, Feb. 24 2001.
- [17] Paul Hoffman, IMC & VPNC, "RACE: Row-based ASCII Compatible Encoding for IDN", <http://www.ietf.org/internet-drafts/draft-ietf-idn-race-03.txt>, Nov. 2000.
- [18] Marc Blanchet, Viagenie & Paul Hoffman, IMC & VPNC, "Internationalized domain names using EDNS (IDNE)", <http://www.ietf.org/internet-drafts/draft-ietf-idn-idne-02.txt>, Mar. 19, 2001.
- [19] Edmon Chung & David Leung, Neteka Inc., The DNSII Multilingual Domain Name Protocol, <http://www.ietf.org/internet-drafts/draft-ietf-idn-dnsii-mdnp-02.txt>, Feb. 2001.
- [20] Paul Hoffman, Mark Davis, "LACE: Length-based ASCII Compatible Encoding for IDN", <http://www.ietf.org/internet-drafts/draft-ietf-idn-lace-01.txt>. 2001-Jan-05
- [21] RealNames, [http://web.realnames.com/Eng/Eng\\_Corporate\\_RealNamesHomepage.asp](http://web.realnames.com/Eng/Eng_Corporate_RealNamesHomepage.asp)
- [22] PHP-Hypertext processor, <http://www.php.net/>
- [23] The Squid Web Proxy Cache, <http://www.squid-cache.org/>
- [24] Squid filter module, <http://sites.inka.de/bigred/devel/squid-filter.html>
- [25] Autoconvert, 漢字內碼智能識別轉換程序, <http://banyan.dlut.edu.cn/~ygh/>
- [26] Walid.com, <http://www.walid.com/>
- [27] John C. Klensin, "A Search-based access model for the DNS", <http://www.ietf.org/internet-drafts/draft-klensin-dns-search-00.txt>, May 2001.
- [28] Li Ming TSENG, Jan Ming HO, Hua Lin QIAN, Kenny HUANG, "Internationalized Domain Names and Unique Identifiers/Names", <http://www.ietf.org/internet-drafts/draft-ietf-idn-uname-00.txt>, Mar. 2001.
- [29] Jack Halpern, Jouni Kerman, "The Pitfalls and Complexities of Chinese to Chinese Conversion", <http://www.basistech.com/articles/c2c.html>
- [30] 3721.com, <http://www.3721.com/>
- [31] KanHan.com, <http://www.kanhan.com/>
- [32] VeriSign, "Internationalized Domain Name Testbed", <http://www.verisign-grs.com/idn/index.html>
- [33] Sen, J., Yap, J., "iDNS - The Next Big Step in the Internet Saga, ", Dec 1999, available from I-DNS.net, <http://www.i-dns.net/>
- [34] RichWin, <http://richwin.sina.com.cn/>
- [35] 信使繁簡通, <http://www.infoscape.com.cn/>
- [36] F. Yergeau, "UTF-8, a transformation format of ISO 10646", IETF RFC-2279, Jan. 1998.
- [37] Mark Welter, Brian W. Spolarich, Adam M. Costello, "Differential Unicode Domain Encoding (DUDE)", <http://www.ietf.org/internet-drafts/draft-ietf-idn-dude-02.txt>, Jun. 2001.
- [38] Patrik Falstrom, Paul Hoffman, "Internationalized Host Names In Applications (IDNA)", <http://www.ietf.org/internet-drafts/draft-ietf-idn-idna-02.txt>, Jun. 2001.
- [39] XiaoDong LEE, HSU NAI-WEN, GuoNian SUN, "Traditional and Simplified Chinese Conversion", <http://www.ietf.org/internet-drafts/draft-ietf-idn-tsconv-00.txt>, Jun. 2001.
- [40] Yoshiro Yoneya, "Evaluation of various ACEs with existing Japanese Domain Names", <http://www.ietf.org/internet-drafts/draft-ietf-idn-ace-eval-jp-00.txt>, Jun. 2001.