國立暨南國際大學資訊工程學系

碩士論文

校園網路下搭配 MySQL 和 LDAP 之垃圾語音防護系統

Countermeasures of Spam over Internet Telephony in SIP.edu Campuses with MySQL and LDAP Support

指導教授：吳坤熹博士

研究生：吳菖育

中華民國九十七年 一月

致謝

在進入暨大資訊工程所之後，經由在研究的過程中，發現很多技術需要去磨練。而在這學習的階段，很感恩吳坤熹老師的教誨，指導了不少有效的做事方式，也不時的討論並指點我正確的方向，使我在研究所的修業中獲益匪淺。老師對學問的嚴謹更是我輩學習的典範，且教導我們在面臨問題時該如何冷靜分析並擁有解決的能力，並直接點出我需改進的缺點。讓我在未來不管在職場或是生活中，不會因這些缺點而再犯下相同的錯誤。

論文名稱：

校園網路下搭配 MySQL 和 LDAP 之垃圾語音防護系統

校院系：國立暨南國際大學資訊工程所　　頁數：82

畢業時間：97 年 01 月　　學位別：碩士

研究生：吳菖育　　指導教授：吳坤熹博士

中文摘要

隨著 VoIP(Voice over IP)的蓬勃發展，網路電話成為了用戶之間一項極受歡迎的溝通管道。而廣泛存在於電子郵件中的垃圾郵件(SPAM)問題，也將帶給 VoIP 類似、甚至更嚴重的困境。垃圾郵件一直被用來成為廣告商行銷的手段。廣告商只要透過一份用戶的位址名單，就可以有效的大量發送廣告信件給各用戶。用戶收到這些不請自來的信，往往需要浪費大量的時間人工過濾刪除。雖然現今有不少的郵件過濾軟體，但仍無法保證能百分之百的過濾掉所有的垃圾郵件，只能降低收到垃圾郵件的機率。垃圾語音(Spam over Internet Telephony，簡稱 SPIT)也即將成為現今 VoIP 上所面臨的嚴重問題之一。在網路電話系統裡將會收到很多不請自來的語音廣告訊息。

由 Internet2 所提出的 SIP.edu 機制結合了網路電話和電子郵件的身份識別方式，用戶將不再需知道每位使用者冗長的電話號碼，這套機制可以很方便地提升以會議初始協定（Session Initiation Protocol，簡稱 SIP）的通訊應用便利性。但 SIP.edu 的便利性卻也成了廣告商發送垃圾語音的工具。本論文裡將會設計一個抵禦 SPIT 的平台，使用黑白名單跟 SIP.edu 的概念來過濾每通電話。另外，在黑白名單的資料庫設計上將會對 LDAP 及 MySQL 這兩種不同的資料庫軟體做效能上的分析。

關鍵字：輕量級名錄存取協定(LDAP)、SIP.edu、垃圾語音(SPIT)、網路電話

(VoIP)

Title of Thesis：

Countermeasures of Spam over Internet Telephony in SIP.edu Campuses with MySQL and LDAP Support

Name of Institute：Department of Computer Science and Information Engineering, National Chi Nan University Pages： 82

Graduation Time：01/2008 Degree Conferred：Master

Student Name：Chang-Yu Wu Advisor Name：Quincy Wu

英文摘要

With the great progress of VoIP (Voice over IP), the Internet telephony has become a popular communication means for users. However, there is a potential threat to this emerging technology. In current Internet, SPAM is a well-known security issue of e-mail systems. Advertisers can send a large amount of junk mails to users chosen from a list. Although there are lots of spam-filtering softwares, none of them can perfectly stop all spam mails, but can only reduce the probability of receiving spam. It is foreseeable that sooner or later SPIT (Spam over Internet Telephony) will also become a serious threat. Users may receive lots of unsolicited audio advertisement messages via VoIP.

SIP.edu is one of several Internet2 new initiatives seeking to promote advanced peer-to-peer communication applications. It provides convergent identities for both voice and email applications. With SIP.edu, user could use the same identity to send emails and make SIP calls, and would not be burdened with a long sequence of digits to make a phone call. However, although SIP.edu brings us lots of convenience, it also becomes an efficient tool for spammers to deliver SPIT. This thesis proposed the design to filter SPIT with *blacklists/whitelists* in *SIP.edu*. In addition, we analyzed the performance of two

different databases, LDAP and MySQL, to support the design of blacklists and whitelists.

Keywords：LDAP, SIP.edu, SPIT, VoIP

目錄

圖目錄

表目錄

1. 導論

1.1 現況

網路的快速成長，帶動了人們生活上的改變。利用Internet Protocol（IP）網路來傳輸語音、視訊等多媒體的服務，已經是未來所面臨的趨勢。而VoIP（Voice over IP）[2] 在即時通訊的領域中扮演著愈來愈重要的角色。現今的VoIP網路電話，已逐漸成為個人或企業通訊生活的一部份。只要透過Internet不僅可使用即時語音傳遞的服務，更能透過視訊會議與世界各地聯繫，這對企業而言是一大福音。在考慮到花費的成本以及通話的品質，網路電話與公眾電話網路（Public Switched Telephony Network，簡稱PSTN），都是可選擇的通訊媒介。網路電話的低成本，更加地讓它成為將來通訊的主流。而網路電話與傳統電話，也只要透過閘道器就能彼此互通。

VoIP主要的通訊協定中，SIP（Session Initiation Protocol）[11] 被廣泛採用。SIP為IETF（Internet Engineering Task Force）所制定的多媒體通訊協定，使用之語法源自於HTTP（Hypertext Transfer Protocol）的文字表示形式，是屬於應用層中的通訊協定。它被設計為建立、管理、終止等各種通話信令的控制。由它所定義的六道指令（INVITE、ACK、OPTIONS、BYE、CANCEL、REGISTER）便能完成呼叫與控制的程序。對VoIP此一應用而言，傳統上所使用的H.323 [10] 本身的複雜性高，子協定也相當多，在通話建立上的速度劣於SIP，擴充性也較差。因此在市場上，已有愈來愈多的網路電話產品採用SIP作為通訊協定。當運用在多媒體通訊系統上時，無論網路交換機（IP-PBX）、電信交換機（Softswitch），或是3G（第三代行動通訊）的行動通訊協定，發展的趨勢都是利用SIP當作信令協定。而設備之間的信令傳遞，則須靠代理伺服器（SIP Proxy server）的轉送，

這其中還包含有用戶註冊、轉發指定位址、通話建立等，都需透過代理伺服器的處理。SIP 可說是IP 網路與傳統電話整合的關鍵技術及標準。

1.2 動機

伴隨著 VoIP 的發展，垃圾語音的問題，也陸續帶給使用者額外的困擾。平時在傳統電話上的電話行銷，常見到的方法都是使用自動撥號機來撥打每通電話。而 VoIP 所使用的 IP 網路架構，只要找出大量的 IP address，再將事先錄製好的語音，一次大量傳送出去。即時性的網路行銷方式，絕對會讓用戶不堪其擾。而這種的傳遞方式，IP 網路所負擔的成本比傳統電話小，因此用它來行銷，對廣告商更具有吸引力。另外對網路上既有的安全機制也是一大挑戰，欲傳送的垃圾語音的檔案大小一定遠大於傳統的垃圾郵件，這都對網路的流量造成很大的負擔。

由 Internet2 組織針對網路應用技術的開發，提出了 SIP.edu [7] 一項校園網路服務的機制，去結合聲音與電子郵件特性來提供 SIP 的網路服務應用。SIP.edu 最主要的概念是利用了 SIP URI 與傳統電子郵件位址的相似，把網路電話的應用帶入校園中。使用者在使用網路電話聯絡朋友之時，不需要知道對方的 SIP URI 或者是校園分機號碼，只要撥打時輸入對方的電子信箱位址，系統就會依據紀錄，找出對方目前註冊的位址進行通話處理。但 SIP.edu 在提供撥打網路電話便利性的同時，卻也容易成為有心人士發送垃圾語音的平台。今日只要廣告商知道某校園內有 SIP.edu 的網路服務，取得了該校園學生及老師們的電子郵件帳號，就可大量的發送廣告訊息出去。而及時性的通話服務處理，不管是校園分機還是 SIP phone 都會響起，這絕對會令用戶不堪其擾。

因 SIP.edu 有這樣的安全性的問題，在本論文裡我們將針對 SIP.edu 這個校園網路服務，以黑白名單的概念去建立出抵禦垃圾語音的防護機制。對陌生的電話

進行通話建立前的過濾動作，以保障老師及學生們在工作或是忙碌時不受到打擾。

2. 背景與相關研究

Internet 為主的通訊技術已漸漸成為未來趨勢，文字、圖形、聲音等各種訊息依靠 Internet 來傳遞已變得迅速且簡便。但它免費與開放的特性，也同時逐步形成廣告商發送廣告內容的新管道。現今以電子垃圾郵件（SPAM） [22] 型式的廣告訊息，已對個人或企業運作上造成極大的困擾。目前雖針對這問題有多項過濾技術被發展出來，但 SPAM 也在各項通訊環境的進步下，衍生出新的傳遞方式。即時垃圾訊息（SPIM） [6] 和垃圾語音（SPIT） [8,15,16] 就是利用即時通訊系統和網路電話的介面來傳送廣告。我們將在這章節仔細描述 SPAM、SPIM、SPIT 的不同，以及目前已有哪些試圖改善這些問題的相關研究。

2.1. SPAM、SPIM、SPIT

SPAM：凡指未經用戶許可就強行發送到用戶郵箱中的任何電子郵件，其內容大都為廣告性質。大部分的用戶對此信件內容不感興趣且頗具反感。而其原由來自

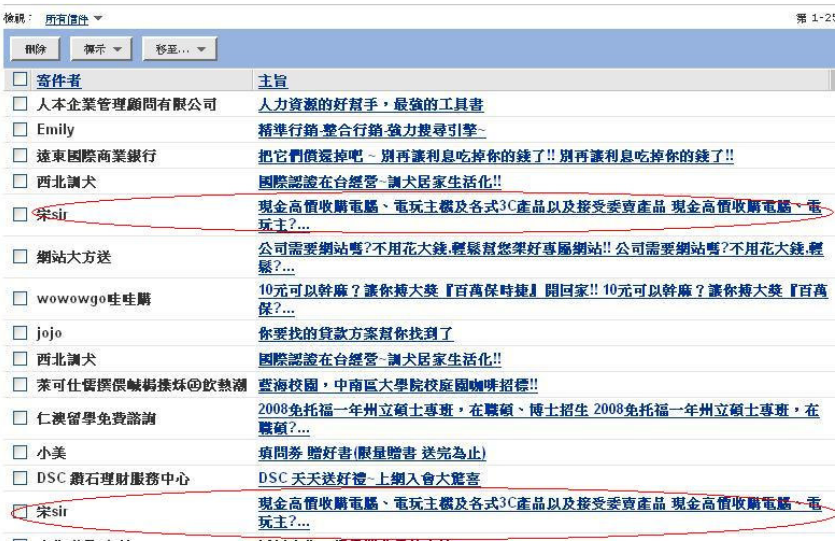

圖 1　　SPAM

有心人士在網路上各論壇或是各大 BBS 等，去收集用戶的電子信箱位址，之後

再轉售給廣告商，透過自動發送信件的軟體就可大量發送電子郵件廣告出去。利用這種行銷方式不但可降低成本且能有效傳達到各用戶。圖 1 為收到垃圾郵件的畫面，可看到紅圈處為重複收到的垃圾郵件，廣告商不斷地發送廣告，而用戶每次在處理這些郵件時，人力或是時間都是無謂的浪費。

SPIM：即時垃圾訊息（**Spam over Instant messaging，簡稱 SPIM**），也就是以 IM（Instant messaging）平台做為傳輸媒介去傳遞 SPAM。即時通訊能讓使用者透過網際網路免費地即時互傳訊息，SPIM 的傳送方法上跟垃圾郵件相似，但不同的是用戶在檢查郵件時才會對郵件內容簡略看過。而 SPIM 是即時性的傳送，用戶往往無法拒絕接收。這對正在工作上的人們無疑是個極為煩人的困擾。而傳送過來的訊息有的還會嵌入惡意的超文字連結，以不實的廣告內容（有獎徵答、愛心公益、特別折扣等等）欺騙用戶去點選，這些超文字連結有時可能是夾雜病毒的網址或是釣魚（phishing）程式。當用戶點選後將導致病毒侵入或被植入木馬，對用戶或企業都是安全上的威脅。更嚴重的情況是大量發送 SPIM 將造成網路壅塞，影響系統的執行效能。

SPIT：垃圾語音（**Spam over Internet Telephony ，簡稱 SPIT**）或稱為 VoIP SPAM，又可被稱為 VAM（Voice SPAM）。它的原理相似於垃圾郵件，透過 IP 網路的媒介，即時性的把語音傳送給用戶，而這些語音往往都是不請自來的垃圾訊息。SPIT 可細分為兩種情況，分別為 Voicemail message、Telemarketing。Voicemail message 的原理就是將預先錄製好的廣告語音傳送給用戶，用戶收到時可自行決定要不要聆聽。Telemarketing 的作法類似於傳統電話行銷，不過 SPIT 的花費成本上比傳統電話便宜，只需負擔上網的費用，就可依事先準備好的名單和錄製好的語音透過軟體發送出去。另外每通垃圾電話所製造的流量，約為 5.1M-byte(11K*8*2*30=5.1M-byte；30 秒的 Pulse Code Modulation (PCM) 編碼的 WAV 音效檔，取樣頻率設為 11KHz，音效類型取單音道 8bit，立體聲雙音道，

時間長度 30 秒)；相較於每封垃圾郵件的幾十個 K-byte，垃圾語音明顯花上相當大的頻寬去傳遞。而它同時也會成為惡意人士作為攻擊他人網路的方法，只要大量發送垃圾語音，就很容易癱瘓了網路。很明顯的例子像是 DoS（Denial of Service）攻擊，只要不斷地發送垃圾語音，就有可能造成網路壅塞。此外對使用者而言，如果每天收到二、三十封垃圾信，影響還不大。因為一般使用者的習慣，是在有空時才坐到電腦前，一口氣把三十封信處理掉。但若是以網路電話的型式，一、二十分鐘進來一通垃圾電話，使用者一定因為工作被頻繁地打斷而不勝其擾。圖 2 為廣告商利用 IP 網路去發送垃圾語音，廣告商透過撥號軟體將預先錄製好的廣告語音經由 SIP Proxy server 傳遞到用戶的語音信箱。

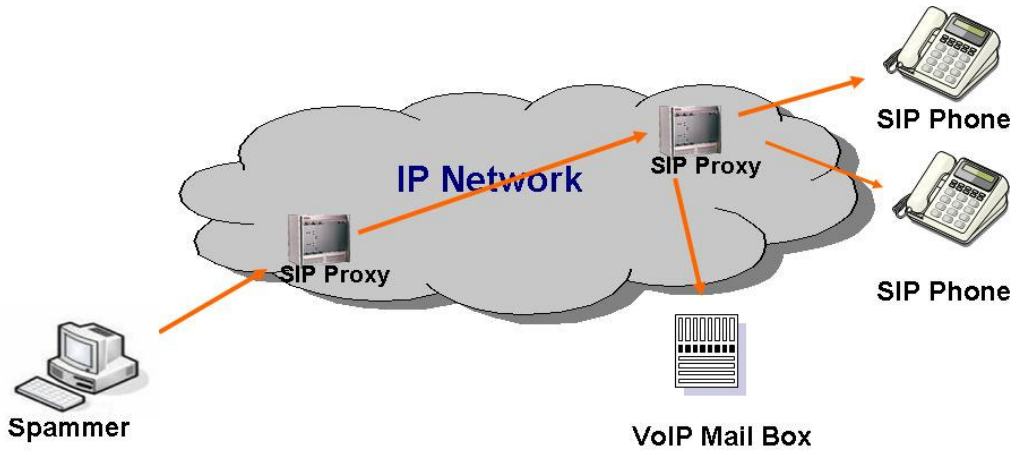

圖 2　　SPIT

2.2. 相關研究

在對 SPAM 或是 SPIT 的防禦研究上，通常可分為幾個作法。1.實作一台專門認證用戶身份的機器。2.建立過濾垃圾語音的 VoIP 網路環境再去對各個用戶去做評分機制。3.大量收集每次通話時所花的時間去評估對方是否為垃圾語音。4.以黑白名單的機制去對來源做過濾。5.針對用戶的回覆信息進行加密以防偽造。接下來我們將介紹抵禦 SPAM 的技術。

2.2.1. SPAM Filtering

2.2.1.1.Content Filtering

關鍵字 [5] ：在傳遞一封信時，有幾個必須的要素，例如信件的標頭、內文、寄件者等。關鍵字過濾就是針對這些要素去做文字的比對。舉例來說，廣告商在發送垃圾郵件時往往都會有常用到的廣告特定字詞，而這些字詞就可作為判斷垃圾郵件的基準。當這些特定字詞出現過多時，系統可依據出現的比率來認定是否為垃圾郵件。所以在信件的標頭或內文去做關鍵字的分析，就會具有過濾的功效。但是如何選定關鍵字，卻是很難界定的關鍵點。若審選不當就會造成誤判情形；另外，有心人士也可在關鍵字中添加空白或其他符號，造成系統的辨識準確度降低。

2.2.1.2.外部資料庫的引用

DCC [5] ：DCC（Distributed Checksum Clearinghouse），此一技術應用於辨識大量郵件。最主要的作法是當具有 DCC 功能的郵件伺服器收到信件時，則會立即產生信件檢查碼(Checksum)，收集好這些檢查碼後，再將這些檢查碼傳送至 DCC 伺服器，而 DCC 伺服器收到回報的檢查碼之後，定期的自動更新內容，並同時告知回報來的郵件伺服器此檢查碼出現的次數。DCC 伺服器及郵件伺服器不停的相互更新回報，之後就可依據收到的郵件去比對檢查碼。因為不同信件所產生的檢查碼會不相同，這些數值代表著這郵件曾在其它郵件伺服器上被傳送的次數。當伺服器發現某一封信的檢查碼的總數過多時，就可判定此一為廣告商大量發送的垃圾郵件。

Razor [5] ：Razor主要的功能為郵件中特定的關鍵字字詞隨機取樣、信件編碼解碼的判斷、附件檔案分析，同時再配合多方搜集垃圾郵件指紋碼(SHA的雜湊演算)，回報資料給Razor線上資料庫，建立即時更新的指紋資料庫。而之後的郵件

伺服器可根據Razor線上資料庫的資料作為判斷垃圾信件之準則。而另一套Pyzor的功能與Razor一樣，但不同的是Razor線上資料庫所提供的比對資料庫並不是免費軟體，而是Pyzor則是公開的開放原始碼軟體，使用者可自行去制定過濾條件的資料庫。

2.2.1.3.機率運算

貝氏演算法（Bayesian Filtering） [5] ：貝氏演算法是目前辨識率相當高的一項過濾技術，採用的概念來自機率學中的貝式定理來分析郵件，搭配定量的中央規則，把所有垃圾信件及非垃圾信件通通代入公式後，利用貝氏演算法計算出適當的機率值，利用這機率值就可去分辨出垃圾信件與非垃圾信件。這其中投入分析的郵件量越大，其機率值準確度越高。因為每次算出的機率數值都是當時環境的情況並非永久環境情況，為確保它的攔截精準度，必須時常重新計算。

2.2.1.4.DNS 反查

SPF [23]：Sender Policy Framework，可以保護指定網域下的使用者，防止被惡意人士冒充發信的一種機制。方法是在 DNS 內加入 SPF record，說明這個域名只會透過那些主機發送郵件。公佈這個網域有哪些合法的郵件伺服器。當郵件伺服器收到信件時，經由 DNS 查出 SPF 紀錄，就可確定是否為合法主機，若查出不是，則很有可能是假冒的。

2.2.2. Blacklists、Whitelists、Graylists

主要原理是去收集特定主機或是 IP address 然後收錄在一份名單中，當收到

信件的來源位址出現在名單中，系統會另行處理。這種以名單式的檢查方法，套用在過濾 SPAM、SPIM 或是 SPIT 上都具有相當功效，且容易融入其他協定中進行檢查，例如：SMTP、SIP 等等。在用戶還未收到廣告訊息前，系統就能幫你抵擋下來。在 VoIP 裡，我們所面臨的 SPIT 問題是即時的語音內容，一般而言為了保障通話者的隱私，法律是不允許去任意檢查每通電話內容的。較可行的方式是利用 SIP URI 進行分析，而黑白名單的機制便可運用來檢查 SIP URI 的方法。而我們之後所用到的好友名單，就是針對特定的網域、郵件來源或 IP address 做有效的阻擋，一經比對後就可對來源進行阻擋處理或者接受。

黑名單（Blacklists）：當發送端的來源，存在此名單中，則拒絕接受。目前有很多偽裝來源的方法，來隱藏真實位址，使伺服器在辨別上容易誤判，所以都會同時配合白名單技術來增加分辨率。

及時性黑名單（RBL）：Real Time Black-hole Lists，主要由特定組織經由收集各方的資訊在網路上建立一份完整的黑名單資料庫，提供給各地的郵件伺服器去做即時性的查詢。郵件伺服器本身可依據 RBL 的名單自行判斷是否拒收名單中的信件。但 RBL 的資料庫建立，有時難保它的正確性，原因是過度的制式條件，會產生誤判的情形。好比說有使用者一時覺得某 IP address 為可疑對象則進行舉報。此時系統因舉報就加入黑名單，名單就不正確了，誤攔率則因此升高。必須選擇信任度高的公司。例如 ORDB（www.ordb.org）、MAPS、RBL（www.mail-abuse.com）等，大家所公認的正確性高的資料庫，才能制定好黑名單的機制，對於阻擋垃圾信才會有卓越的功效。

白名單（Whitelists）：唯有名單中所認定的來源，才會去接收，是一種主觀性很強的技術。發送端都需經過接收端的認可才能成功送達。在過濾垃圾郵件的檢查，是常用的技術。

灰名單（Graylists） [20] ：Graylists的工作模式來自於黑白名單的方法，也是由多個E-mail位址及網域名稱所組成的清單。當一封訊息抵達時，它將被隔離。寄件者則必須在某段時間內再次傳送一次。則寄件者才會被系統認定為合法的寄件者，並且未來它傳送的訊息將直接通過而不受阻礙。有限時間內要求重送是Graylists設計上主要的原則。除非一個寄件者在規定的時間內從相同的E-mail位址把第一封的信息傳送給有支援Graylists的使用者兩次，否則過濾器將會阻擋這訊息。使用Graylists功能的系統不需有特別的架構配置，同時對使用者的電腦上的資源不會增加多餘的負擔。

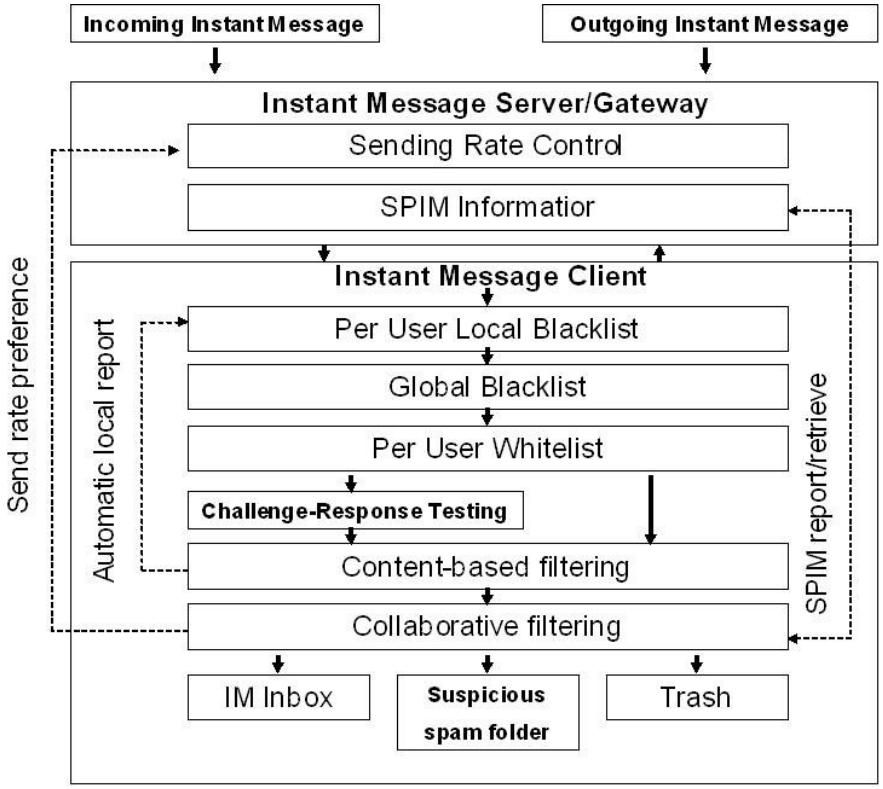

圖 3　　過濾 SPIM 架構圖

圖3是利用了黑白名單的機制和其他過濾的方法去對SPIM的防範架構[17,19] ，其中黑名單的部份，不只是使用了使用者自身設定的黑名單，還用了系統提供的可靠組織的黑名單來進行訊息過濾。Challenge-Response檢查的作法就雷同於灰名單的原理，對寄件者要求在限制時間內做出回應否則對寄件者發出的訊息給丟棄。而Content-based filtering則是以關鍵字的方式對訊息內容做匹配

審核。圖4是過濾SPIM的流程圖 [19] ，先檢查黑白名單之後再去進行其他的過濾檢查。

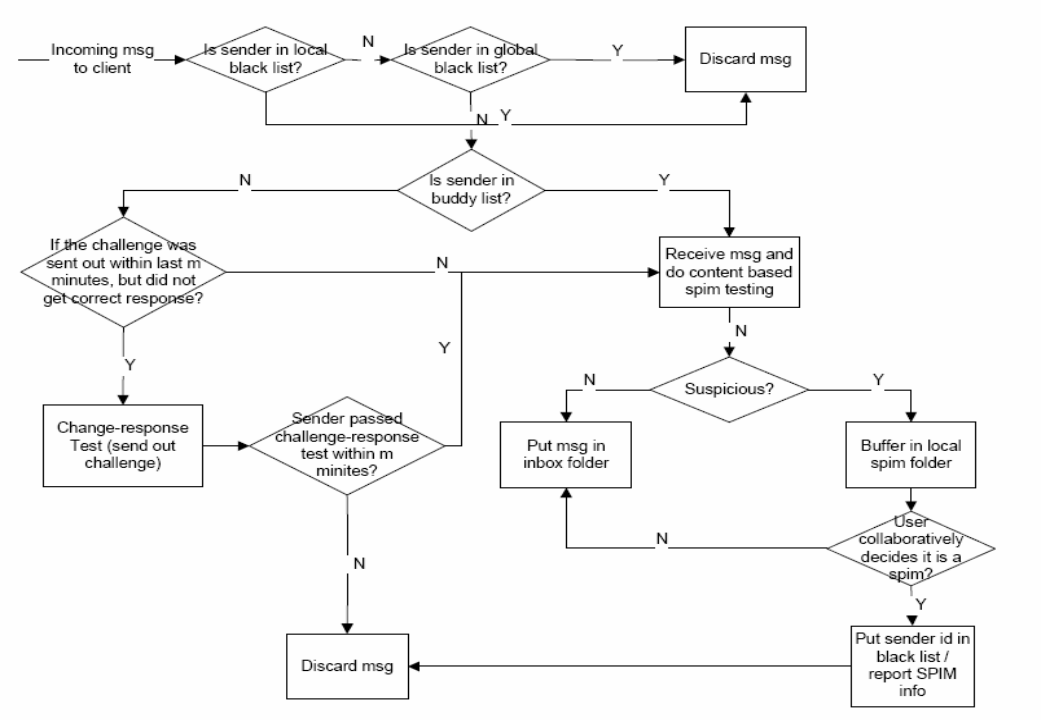

圖 4　　過濾 SPIM 流程 [19]

2.2.3. Anonymous Verifying Authorities

在 IP Telephony Networks 裡，設計一個對用戶進行身分認證的機器。在建立每通電話時，須先彼此檢查雙方的身分，檢查無誤後，才發出一個可通訊的 token 給雙方的 SIP Proxy server 告知可建立 RTP 連線。圖 5 為防禦垃圾語音的模型，當用戶撥打電話出去時，SIP Proxy server 轉送到 Mediator。Mediator 是個中介器。它負責將 Request 以隨機方式傳送給 Verifying Authority（VA） [13]，而 VA 則會對 Caller 或 Callee 進行身分認證。確認無誤後會發出 token 給彼此雙方的 SIP

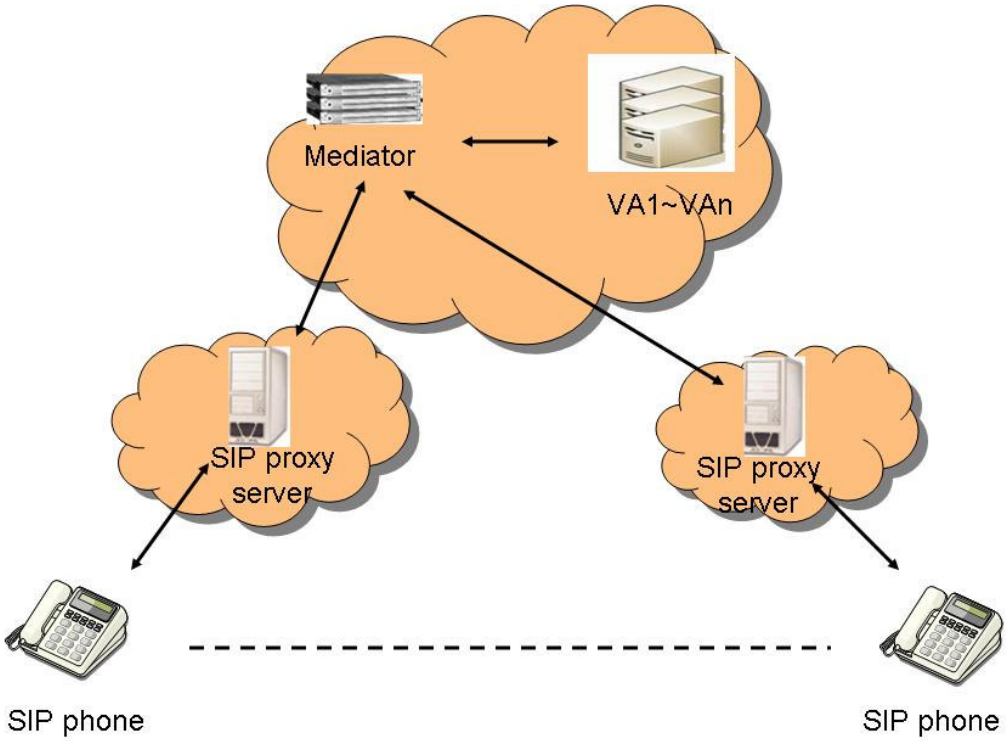

圖 5　　VA 系統模型

Proxy server，SIP Proxy server 收到 token 後，此時才可建立通話的連線。圖 6 為通話建立的流程。舉例來說，當 Susan 要打電話給 Bob 時，此時發出的 INVITE 將會由 proxyA.com 轉送到 Mediator，而 Mediator 會隨機找一台 Verifying Authority（VA）來進行身分認證。所以會將 INVITE 再次轉送到找到的 VA，以隨機的方式選擇 VA，主要原因是不讓專門竊取資料的有心人士，去得知專門認證身分 VA 的固定位址。當 VA 收到 INVITE 時，會發出 Policy request 給 proxyB.com 要求 Bob 的基本資料來進行認證。VA 在確認雙方的資料時，若是確認有問題，則結束這通電話；若確認成功，則予以呼叫 Bob。而這段確認的時間，Susan 的狀態會是 Ringing。當 VA 確認成功後會有三種情況，一個是超過預設的響鈴總時間數，將會通話結束或轉入 Voicemail，另外兩個情況是由 Bob 決定拒絕或是接收，拒絕則一樣會通話結束或轉入 Voicemail。若為接受的話，VA 會收到 Bob 的確認信號。VA 立即發出 token 給雙方的 Proxy 告知將要建立通話。而這段期間，PoxyA.com 必須確認 Proxy.B.com 已收到 token 才能真正開始建立通話。Susan

跟 Bob 的通話建立，都透過了 VA 的認證，以及 Bob 是否接受的 Response 來決定。即使今天 Bob 為已註冊狀態，但若 Bob 表示不予接受這通電話，Susan 將收到 BYE 或是轉入 Voicemail 的指示，Susan 就無法確認 Bob 目前的線上狀態。

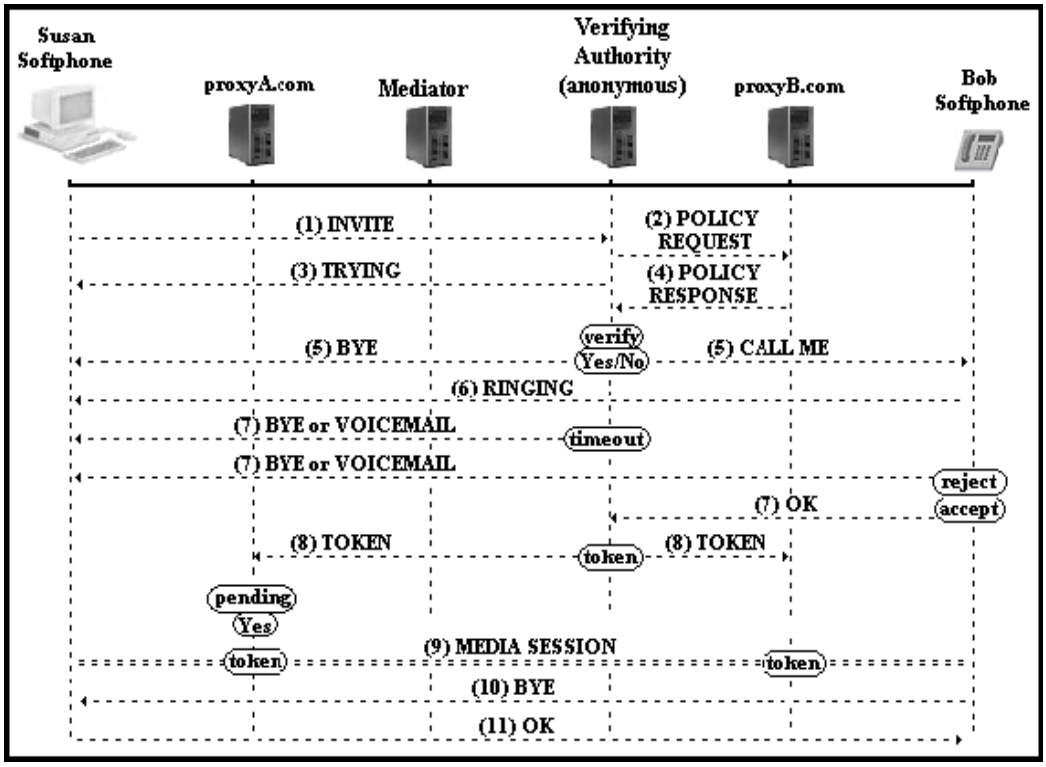

圖 6　過濾 SPIT 流程 [13]

2.2.4. SIP Social Network

SIP Social Network [18] 是在社群網路下，以信譽的方式去評斷用戶來達到過濾的體制。社群網路可視為相互連結的一群人，而不同社群網路中的每個用戶都會有各自的評分值，依據這評分值可讓社群內的用戶來決定是否信任對方，同時也可讓不同的社群參考。舉例來說，Yahoo 拍賣網站，在每次交易完成後，買方或賣方都會給對方與以評價，之後還有其他交易正在進行的話，其他的買家可根據這評價來決定是否交涉。而這種運作方式也可套用於通訊錄中，如圖 7，A、B 兩個網路群，裡面各有一份通訊名單。對這名單中的成員，每個人都會給對方

不同的評分值，假設 A 的網路群裡，對 N_1 有七分的總評價值，N_2 有五分的總評價值，則 N_1 的可信任度比 N_2 高。此時 A 網路群的用戶可根據通訊錄內的評分值去判斷哪些人可信賴。若今日 A 網路群的用戶撥打給 B 網路群中的人，B 網路群的用戶此時就可參考 A 網路群的通訊錄來判斷對方是否可信賴，再決定是否接受通話。所以可透過各網路群通訊錄的內容，提供系統作為辨別評分的依據，最後群跟群之間相連起來，就可形成可信賴網路的集合。

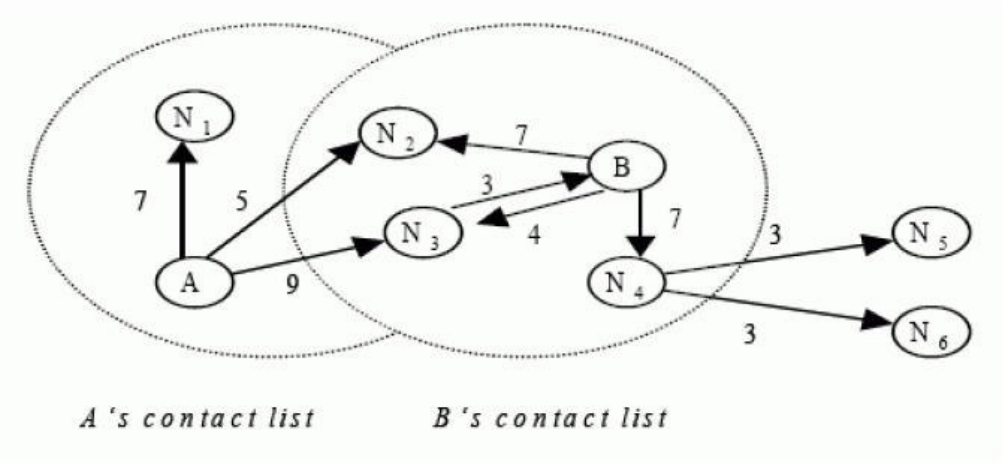

圖 7　　SIP Reputation Network [18]

在 VoIP 的環境中，可利用這用戶信賴機制去對 SPIT 去進行預防動作。當有通電話到來時，此時系統將檢查通訊錄去判斷撥打方的評分值是否可信任。若確認可接受則放行，達不到合格邊緣則拒絕。

2.2.5. Signaling Protocol Analysis

在過濾垃圾語音的方法上，去對語音內容去進行監聽或檢查是不合法的，所以提出以 Signal 的狀態去分析 [14] 。在 Signal 傳送時，去記錄每次所花費的時間都可作為分析的依據。舉例來說，在 2.1 章節中提到垃圾語音中的電話行銷，廣告商一般是將事先錄製好的廣告語音撥打給用戶聽。當用戶接起電話時，聽到的內容為廣告語音。通常用戶對此內容不感興趣，估計從接起電話到掛掉電話所

花費的時間約為 8 秒鐘。因此系統若記錄這通電話所花費的時間，然後去收集某來源所發出的 Signal，依照他與各用戶互動所花的時間是不是剛好都在 8 秒鐘左右。當收集到相當的數量皆為此情形，則可決定發送端是不是真的為廣告商。所以這個方法大量去收集每通電話所發出的信令，從通話建立到結束所花時間，再去分析。對於不管是電話行銷或是傳送垃圾語音郵件，信令分析是可運用的判斷機制。

2.2.6. Response Identity and Authentication

以 Authenticated Identity Body（AIB） [9] 搭配 DAS（Domain-based Authentication Service）去對用戶在回送 Response 時去做身份的確認，以防護有心人士去竄改 Response Header 進行非法行為或是發送垃圾語音等。AIB 為 RFC3893 針對 SIP 協定上的身分認證所提出的專門機制。主要是對 SIP 的 Header 進行加密動作。當要傳送含有 AIB 的 INVITE 信令時，會先複製一份 SIP 的 Header 於 INVITE 的 Request 裡，而這被複製的 Header 會以數位簽章的方式進行加密，以確保在傳送中不被人竄改，進而加強身分辨識的準確度。所以站在 DAS 的角度來看，DAS 能夠處理各種匿名的訊息，是因為回覆方必須在回應的訊息 Header 裡去清楚描述以表示身分，但是無法去隱藏所要傳送的資訊內容。所以 AIB 在回傳的 Response 選擇好所要隱藏的 Header 資訊，再用公開金鑰（Public key）去進行加密動作，以解決 DAS 無法隱藏中間資訊的這項缺點並加以改善。圖 8 [9] 是當系統收到 Response 時，會去檢查 Response 內是否有加密的訊息，解密後確認身分無誤，才繼續其他步驟。

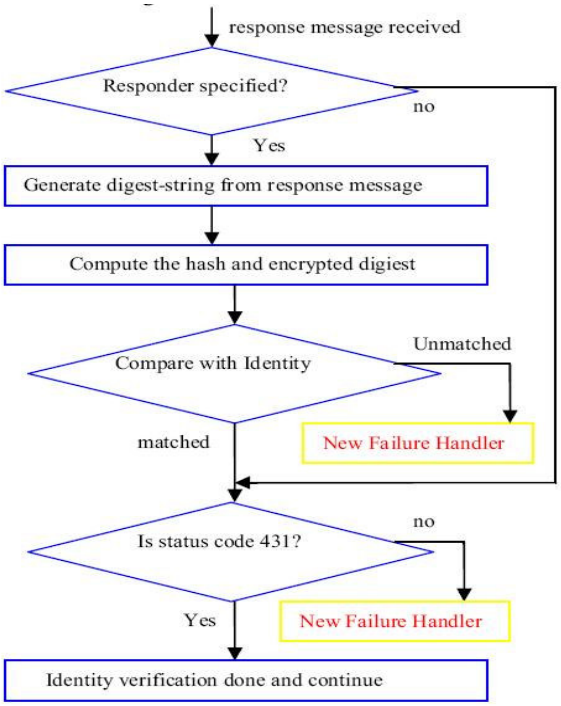

圖 8　Response Identity [9]

AIB 利用私密金鑰（Private Key）在回覆訊息進行簽章的動作來確保資料的完整性及機密性，之後再去對 Response 內容去做身分的認證，不但可以預防假冒的情況，在對很多惡意廣告商利用他人機器作為發送 SPIT 跳板，可用來抵擋 SPIT 被從中攔截而竄改資料，再偽造他人身分的作法。

3. SIP.edu

在西元 1996 年，由美國眾多大學共同提出的 Internet2，主要目標是為各個研究機構建立出穩定並且寬廣的網路環境，以利開發先進的網路技術與應用。而 SIP.edu 則是 Internet2 近期被引進的一項校園網路技術，結合網路電話與電子郵件特性來提供 SIP 的網路服務應用。

平時在校園內常用的聯繫方法有校園分機、電子郵件、SIP phone 等等。但往往使用者在選擇聯絡方法時，都須了解對方帳號或分機號碼。而在處理眾多的帳號時，用戶很難去完全記住所有聯絡人。雖可用通訊錄分門別類去管理，但當聯絡清單被細分為電話簿或是電子郵件通訊錄時，就會在聯絡的過程中有白忙一場的可能。因為每個用戶習慣的聯絡方式有所差異，有的人習慣用校園分機或是 SIP Phone。若是一一去嘗試各方法與對方聯繫，勢必會有徒勞無功的情形。在希望能快速地與對方聯絡情況下，統整以一個帳號來聯繫，將會省掉不必要的嘗試時間，增加更多的便利性。SIP.edu 就是利用了 SIP URI 與傳統電子郵件格式的相似，把網路電話的應用帶入校園中。使用者在使用網路電話聯絡朋友之時，不需要知道對方的 SIP URI 或者是校園分機號碼，只要撥打時輸入對方的電子信箱位址，系統就會依據紀錄，找出對方目前註冊的實際位址進行通話處理。而 SIP.edu 也可對用戶目前線上狀況，再去進行其他的通話處理。接下來我們將再一一敘述 SIP.edu 的架構和運作。

3.1. 架構

SIP.edu 整合了帳號的一致性，所以在描述 SIP.edu 的架構前，先來了解 SIP.edu 在撥打時所使用的帳號。SIP.edu 的帳號是參照 RFC3261 [11] 文件裡對 SIP URI 的敘述，SIP URI 乃是一個用戶的 SIP 的電話號碼，就相似於 email 系統中的電

子郵件位址，格式如下：

sip:XX@YY:port

XX：用戶名稱。

YY：主機名稱(通常可為 Domain name 或者是 IP address)。

Port：所使用的 Port。

範例：

sip:alice@ncnu.edu.tw:5060

sip:bob@163.22.21.82:5060

SIP.edu 利用這個類似電子郵件位址的特性，在撥打網路電話給對方時，直接以對方的電子郵件位址來進行撥打，相較於類似(049)2910960-4763 的複雜電話號碼，自然好記多了。同時不需額外再去了解各用戶的 SIP URI，就能輕易地快速撥打給對方。舉例來說，Alice 的電子信箱為 Alice@ncnu.edu.tw，當 Bob 要打給 Alice 時，Bob 只要撥打 Alice 的電子信箱（Alice@ncnu.edu.tw）即可，不需另外再去查出 Alice 的電話號碼。

在 SIP.edu 的架構裡，最早的概念是當一個 SIP 使用者 Alice，當她撥打給受話端 Bob 時，此時 SIP.edu 系統在處理這通電話時，會把它轉接到受話端的校園分機，如圖 9。

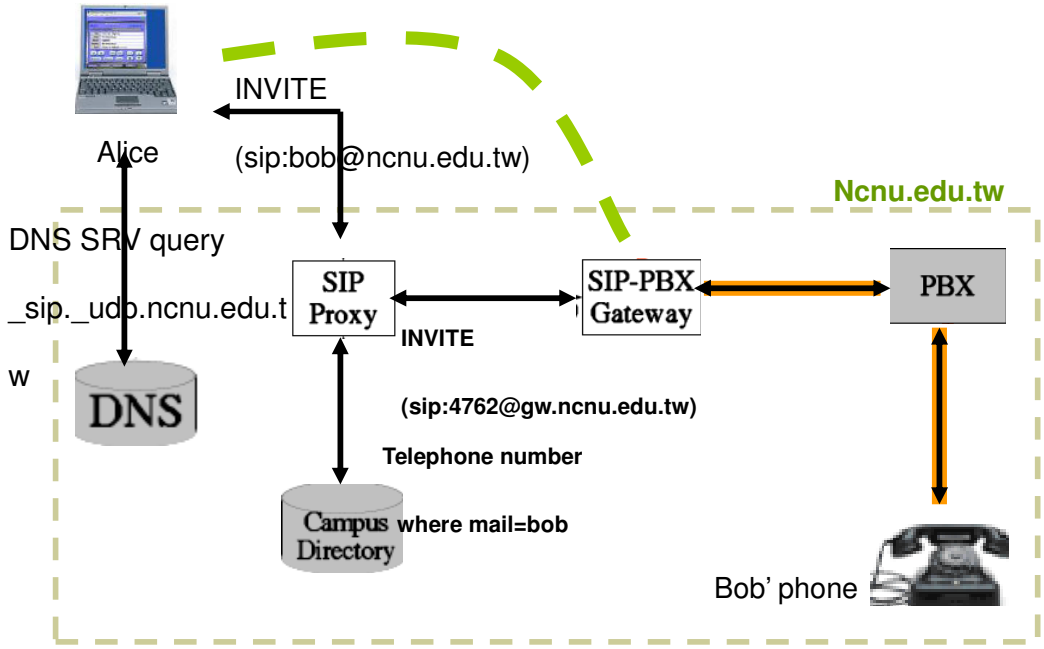

圖 9　　SIP 的初始架構

在圖 9 裡我們可看到 SIP.edu 的架構中有多台伺服器，而各台伺服器都各自負責所屬任務，每個伺服器的功能在此先做描述。

DNS server：DNS 系統必需利用 SRV 的資源記錄類型（Resource Record）來對應各台應用伺服器所提供的服務資訊。在 SIP.edu 裡 SRV 記錄著 SIP Proxy server 的正確位址，進行通話的建立。

SIP Proxy server：專門轉送代理網域的請求跟回應信息，以進行信令的處理。可分兩種運作狀態，分別是狀態性(Stateful)和非狀態性(Stateless)。若為狀態性模式時，在轉送每個信息時，會記錄所有代理伺服器所轉送的請求，以供未來程式處理時所需的依據。而非狀態性則是對每個轉送的信息，其相關的資訊都不會被記錄。

Campus Directory (Location DB)：通常用來儲存每個用戶的分機號碼，以供給

SIP Proxy server需要重新改寫Request URI時所用。

SIP-PBX Gateway：主要作為SIP Proxy server和PBX的橋樑，連通到PSTN網路進行校園分機通話。

PBX：校園內部的電話交換機。PBX對外連到電信局機房的大型交換機，對內則連到校園內部的電話分機。一般所謂的校內分機號碼，就是PBX負責管理的。

接下來以 Alice 撥打給 Bob 的例子來說明 SIP.edu 操作流程。當使用者 Alice 用 Bob 的電子信箱位址（sip:bob@ncnu.edu.tw）撥打給 Bob 時，此時系統去查 DNS server 的 SRV 紀錄，檢查出服務 Bob 的 SIP Proxy server 主機位址。之後 Alice 會發出 INVITE request 到 SIP Proxy server，當 SIP Proxy server 收到 Alice 的 INVITE request 時，會在儲存使用者資料庫中查出 Bob 的分機號碼。當 SIP Proxy server 得到 Bob 的分機號碼時，會把這通電話轉送到 SIP-PBX Gateway，交由校園 PBX 主機進行分機電話處理。

了解SIP.edu的初始架構之後，我們進一步說明SIP.edu在進階架構上所加入的SIP Registrar server（如圖10所示）。SIP Registrar server專門處理用戶的REGISTER request。當UA（User Agent）開機時，UA會先取得可用的IP address，之後會發出REGISTER request給SIP Registrar server告知目前contact address，SIP Registrar server會去更新有關用戶的相關資訊。而以上的動作稱之為「註冊」。之後可依據使用者是否註冊，去進行不同的通話處理，謹以範例說明如下：

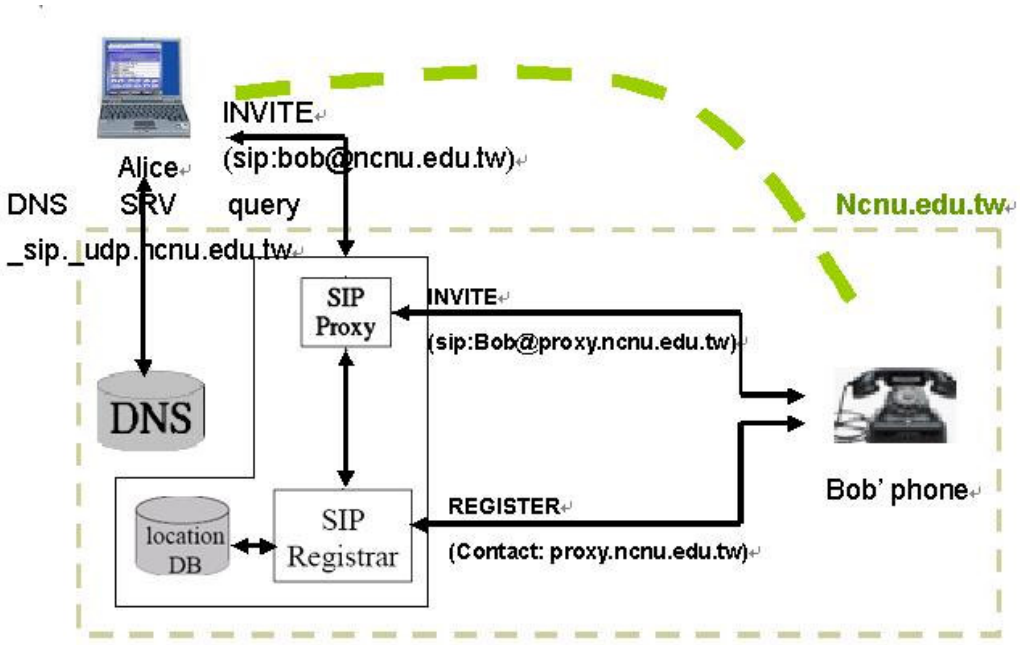

圖 10　支援註冊的 SIP.edu 架構

步驟一：Alice 在她的 SIP user agent 輸入 bob@ncnu.edu.tw 撥打給 Bob。

步驟二：系統此時會去查 DNS server 的 SRV 紀錄查出 ncnu.edu.tw 的 SIP Proxy server 為 proxy.ncnu.edu.tw。

步驟三：Alice 發出的 INVITE sip:bob@ncnu.edu.tw 會傳送到 proxy.ncnu.edu.tw。

步驟四：SIP Proxy server 會從 SIP Registrar server 得知 Bob 現在是否有註冊，再決定是否轉 Bob 的校園分機還是 SIP phone。若是 Bob 的 SIP phone 現在有註冊的話，則 SIP Proxy server 會發出 INVITE 給 Bob 的 SIP phone。

步驟五：若 Bob 目前並沒有註冊的話，此時 SIP Registrar server 會去查出 Bob 的分機號碼給 SIP Proxy server。當 SIP Prxoy server 收到 Bob 的分機號碼時，會重新修改 Request URI，然後再送出 INVITE 給 SIP-PBX gateway。

步驟六：當 SIP-PBX gateway 收到給 Bob 的信令時，將它轉入校園 PBX 主機。

3.2. 衍生問題

SIP.edu所提供的機制允許使用者在僅知道對方電子郵件位址，而不知道電話號碼的情況下，依然能透過此機制以網路電話系統撥通校內分機，提供了相當的便利性。相較於(049)2910960-4763這樣的純數字電話號碼，bob@ncnu.edu.tw這樣的電子郵件位址具有邏輯上的階層性。而在支援使用者有無註冊的情況下去做不同的通話處理，更讓用戶有更多的選擇空間。因此這個機制在Internet2中廣受歡迎，獲得許多大學的採用。但正如同第二章節所提到的垃圾語音問題，許多人擔心SIP.edu的便利性，將會使得廣告商利用手頭上現成的電子郵件清單，將現行以郵件寄送廣告訊息的方式，進一步「升級」為以即時的網路電話，撥打給使用者。這樣的行銷方式對用戶是很困擾的，SIP.edu的好處在這種情形下反而演變成安全性上的缺陷。接下來我們將在下一章中介紹如何對SIP.edu的系統發展垃圾語音的抵禦，為用戶過濾陌生的來電。

4. 系統架構

在第 3 章裡我們提到的 SIP.edu，用戶在撥打電話時，只要以對方的電子信箱位址撥打就可，但它所具有的便利性也容易成了廣告商發送垃圾語音的工具。根據這個問題，我們先以 SIP.edu 的架構作為設計系統的初步雛型，同時加入 Voicemail 的服務於本系統中，提供用戶多項的語音服務。在完成系統的基本環境後，針對垃圾電話的問題，以設計好友名單的過濾機制來預防垃圾電話。用戶好友名單的功能，將會去對撥打給本系統用戶的每通陌生電話去進行分析比對，非名單中的資料將會被轉入用戶的 Voicemail。用戶可根據 Voicemail 的留言內容去決定是否回撥。若用戶有了回撥的動作時，系統將提供一個自動新增好友名單的機制，這功能主要去處理當用戶做出撥打電話的動作時，系統會自動地檢查撥打的對象是否有在用戶的好友名單中，如果不存在名單中，則系統會自動地把對方加入用戶的好友名單中，完成用戶好友名單資料的建立。最後我們再去針對儲存用戶基本資料的資料庫，去評估不同的儲存方式在效能上的差異。

4.1. 系統基本雛型

針對 SIP.edu 服務可能造成使用者困擾的問題，我們分兩個步驟來加強防禦。首先，我們將 Voicemail 的功能加入到系統裡。Voicemail 原本的主要作用，是用戶的 SIP phone 在未開機的情形下時，若有人打電話過來，就可直接轉入語音信箱裡。若撥打方想留言給用戶時，可進行語音錄製，然後以電子郵件傳送給用戶。用戶從 Voicemail 的內容可以知道誰曾有什麼事情找過他，不是只有未接來電的紀錄。而 Voicemail 的功能不單只有提供用戶使用語音信箱的功能，同時也可去針對每通陌生電話去作過濾的功用。

為了避免在引進 SIP.edu 服務的同時造成使用者遭受到即時的干擾，預設的

方式是當對方以使用者的電子郵件地址當作 SIP URI 撥網路電話進來時， 電話會被轉入語音信箱。

當對方留言後，用戶會收到一封由系統寄來的郵件，內含對方留言的語音檔，同時附有簡單的說明（如圖 11），在紅圈處中將告知用戶可利用下方的超連結決定是否啟用 SIP.edu 的功能，讓別人可以直接利用電子郵件地址打電話給你。

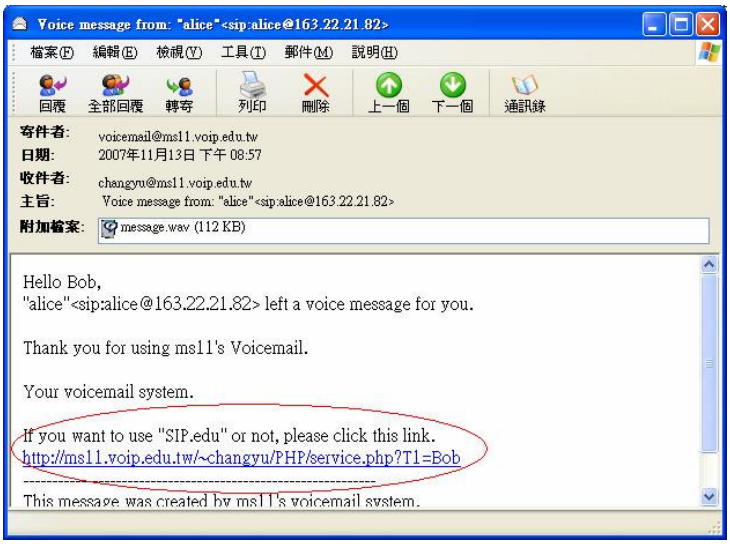

圖 11　Voicemail

當用戶點進郵件中的超連結後，將會看到啟用 SIP.edu 的網頁，如圖 12。用戶可自行選擇要不要啟用 SIP.edu 的功能。啟用之後，將依據用戶目前註冊狀況，將有不同的通話處理。

圖 12　啟用 SIP.edu 畫面

表 1　有無使用系統和註冊的用戶區分

Use SIP.edu? / REGISTER?	YES	NO
YES	SIP phone ringing	SIP phone ringing
NO	Campus phone ringing	Voicemail

何時啟動 Voicemail 的功能或者是轉校園分機或等動作時，規劃四個情況去分辨。在表 1，訂立兩個判斷條件，分別為用戶是否使用 SIP.edu 的功能以及用戶目前是否為註冊的情況。舉例來說，Alice 現為 SIP.edu 的用戶，但是她目前的 SIP Phone 並沒註冊；若 Bob 打電話給她時，則此時系統會把這通電話轉到 Alice 的校園分機。反之若 Alice 的 SIP phone 有註冊的話，當有人來電給她時，則她的 SIP phone 會響起。如圖 13，不管有無使用 SIP.edu 的功能，當有人撥打給用戶時，

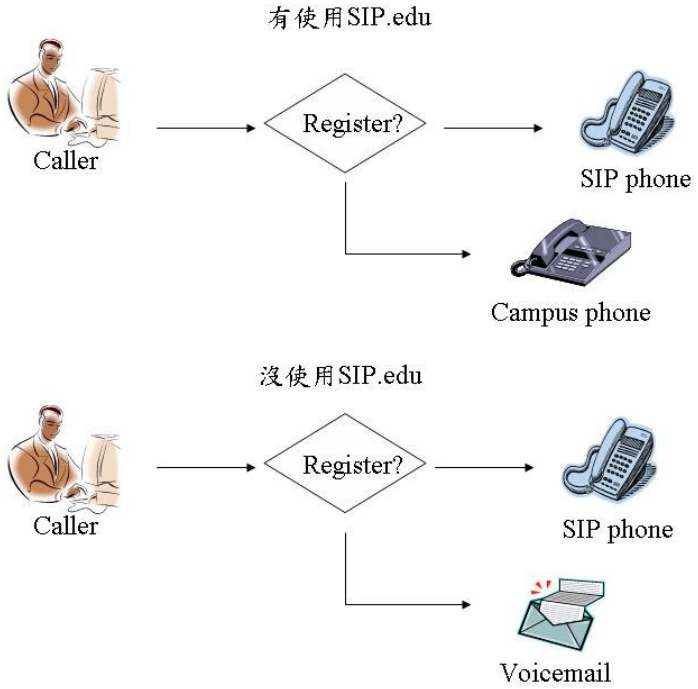

圖 13　用戶撥打流程

只要在有註冊的情況下，用戶的 SIP phone 都會響起，唯一不同的是用戶在沒有註冊情況下，會有轉接校園分機或 Voicemail 兩種不同處理。

圖 14 顯示的是系統的流程圖，針對傳送進來的信令，一般 SIP Proxy server 都會先檢查 Request URI 是否為本端的用戶，不是本端用戶則系統將把它 Relay 出去。確定為本端用戶後，將讀取 Callee 的基本資料（例如：用戶的電子信箱、分機號碼等等）。之後系統判斷用戶是否有使用 SIP.edu 的功能後，再去分析用戶目前註冊的狀況，再決定給予不同通話服務的處理。

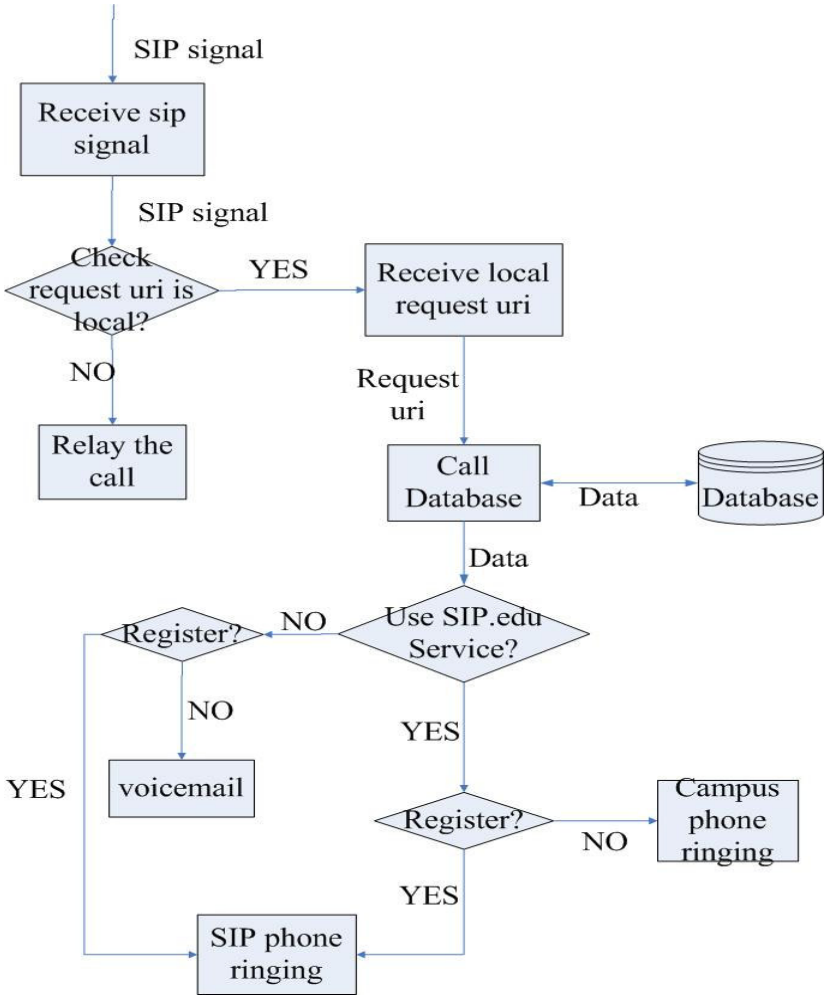

圖 14　系統初始流程圖

4.2. 名單過濾

完成 4.1 章的基本保護措施後，我們進一步在系統裡提供過濾垃圾電話的功能。利用過濾技術中的黑白名單機制，我們運用了白名單的概念，為每個用戶建立專屬的好友名單。這份好友名單可對打給用戶的每通電話，去判斷撥打方是否存在用戶的好友名單中。如果是來自好友的電話，則系統就會執行放行動作；如果沒有，則會轉到 Voicemail 中，讓用戶有空時再聽取留言。表 2 顯示系統針對陌生電話的兩種不同處理方法。

表 2　針對陌生電話的處理

Use SIP.edu / Unknown number?	User of SIP.edu
NO	SIP phone ring
YES	Voicemail

圖 15 描述當 Bob 打給 Alice 時系統處理陌生電話的過程。Alice 是有使用 SIP.edu 的用戶，目前為註冊狀態。Alice 擁有一份好友名單，當 Bob 撥打電話給 Alice 時，系統會找出 Alice 的好友名單尋查 Bob 是否存在 Alice 的名單中。如果有，Alice 的 SIP phone 就會響起，如果沒有則轉到 Alice 的語音信箱。

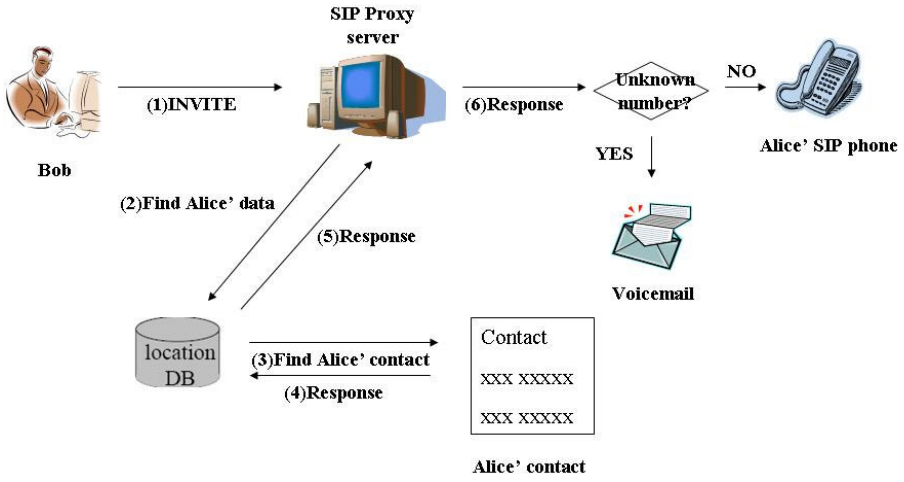

圖 15　陌生電話檢查流程

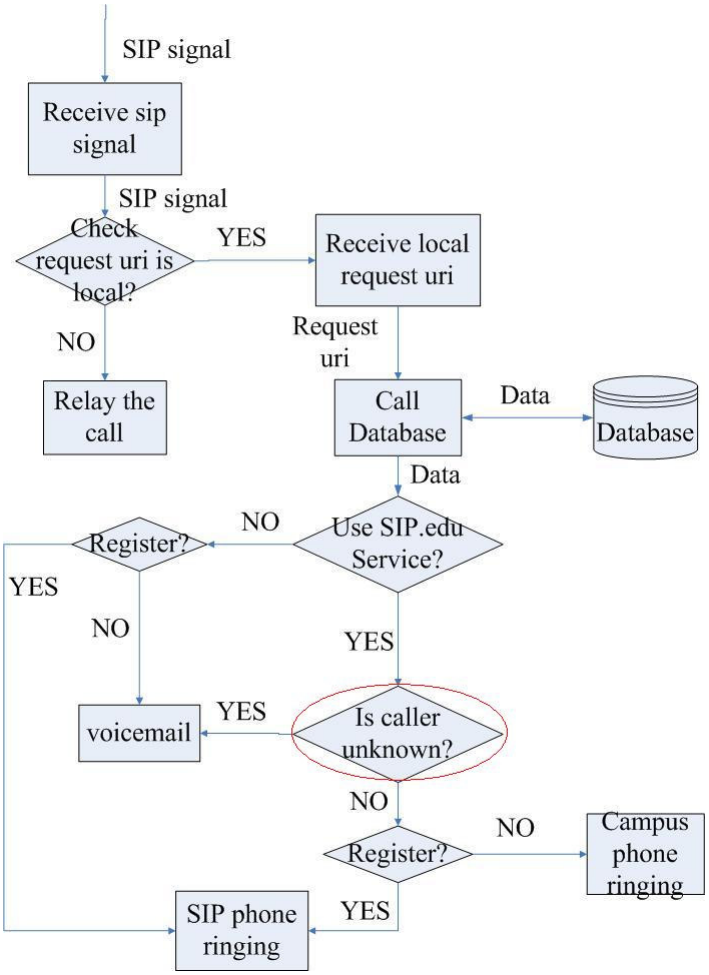

圖 16　陌生電話檢查之流程圖

圖 16 則是系統的流程圖，流程與 4.1 章節的圖 12 相似，不同的是在檢查完用戶是否為 SIP.edu 的用戶後，在紅圈處，多了檢查每通的 Caller 是否有在用戶的好友名單中。

4.3. 自動產生好友名單

好友名單的作用，是可藉以避免陌生電話的打擾。例如 Mozilla 或 Thunderbird 等電子郵件軟體，多半也都提供有 Do not mark mail as junk if the sender is in Personal Address Book 的功能。但若要求使用者以人工方式逐一輸入其好友名單，顯然是件煩雜而令人不快的動作。在這章節則要討論用戶好友名單的產生方

式。當每通陌生電話撥打進來時，系統會直接轉入用戶的語音信箱，用戶此時就可根據語音內容去判斷對方是不是廣告商。如果是廣告商的話，往往都不會回撥回去；若用戶回撥給對方，我們就可以假設對方並非廣告商。系統將自動地把撥打的對象加入用戶的好友名單中，產生出用戶好友名單的資料。圖 17 的流程顯示在 Bob 每次撥打時，系統都會自動地幫他檢查撥打的對象是否有存在他的好友名單中；如果沒有，系統將自動地將撥打對象加入好友名單。

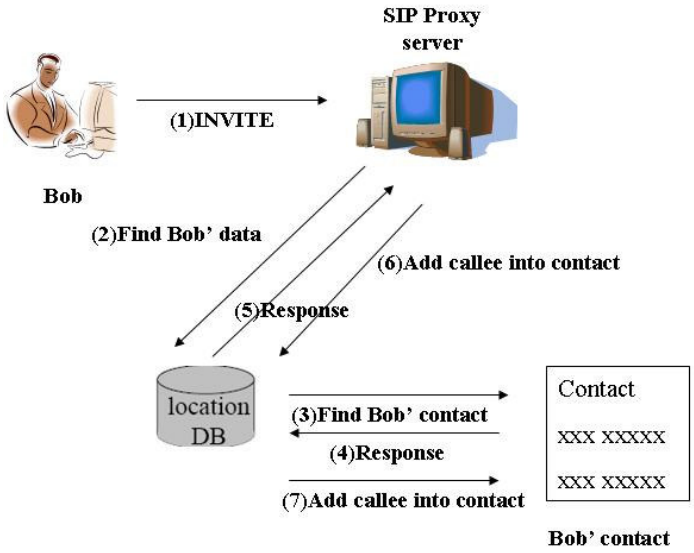

圖 17　自動產生好友名單流程

圖 18 則是好友名單在新增資料的系統流程圖，在系統確認 Caller 是有使用 SIP.edu 功能的用戶後，接著檢查 Callee 是否有在 Caller 的好友名單中，為圖中紅圈處。若沒有則會把 Callee 加入 Caller 的好友名單裡(藍色圈部份)。以上的處理都是針對 Caller 的部份，接下來系統才去處理 Callee 的部份。而圖 18 中綠色圈的部份，則是延續 4.2 章節中圖 16 系統架構的流程。

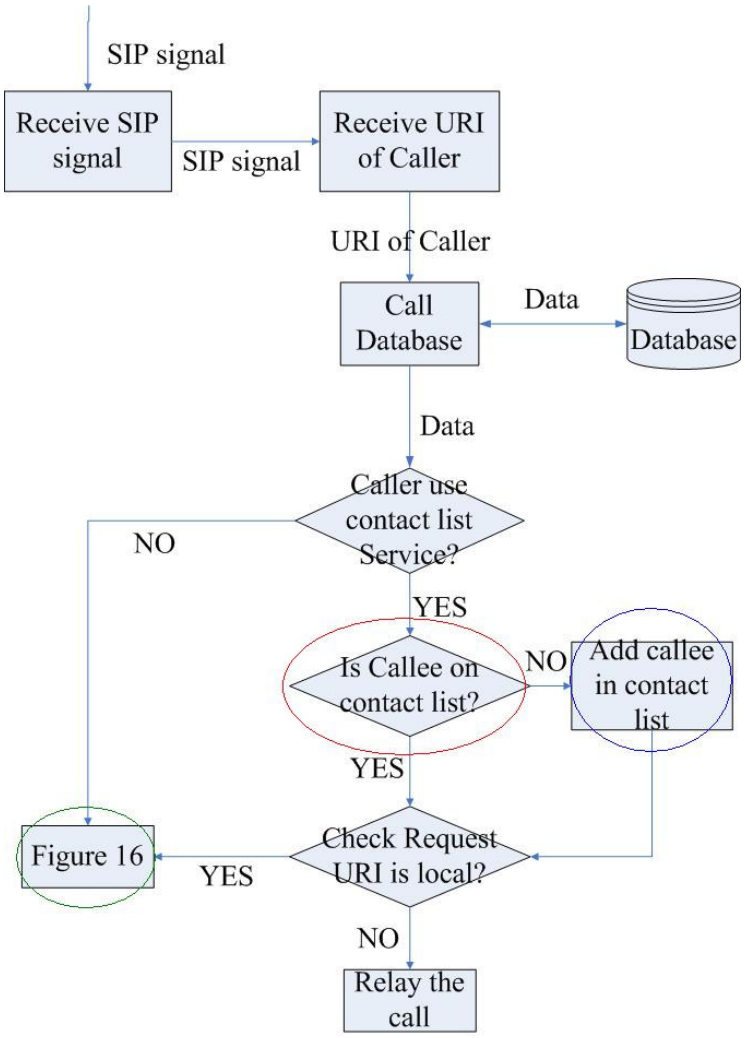

圖 18　新增好友名單流程圖

而好友名單內容的增加，主要是針對陌生電話進來時，用戶會根據 Voicemail 的內容判斷是否為廣告商再決定是否回撥。但有時用戶依 Voicemail 的內容仍不能正確判斷對方的身份時，仍可能會回撥以進行確認。若對方為廣告商，這一個回撥的動作，將導致系統把對方加入好友名單中，則這份名單就會不正確了。所以我們附帶設計了一個讓用戶可刪除名單中資料的 Web 介面（以 PHP 開發 [4] ），可刪除被不小心加進好友名單中的廣告商。圖 19 是某用戶登入管理好友名單的 Web 介面，裡面存放五筆資料，若想刪除某位用戶，就輸入該用戶的 SIP URI 進行刪除動作。

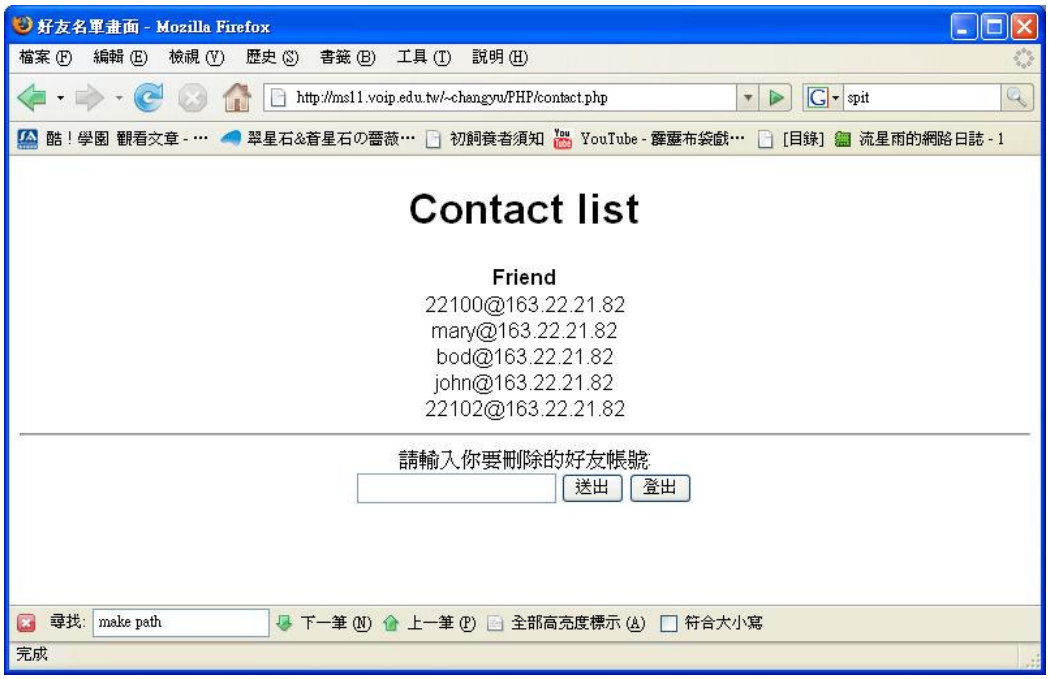

圖 19　好友名單管理畫面

5. 實作與效能測試

本系統以 SIP Proxy server 和 Media Proxy server 作為網路電話的基本開發環境。同時在儲存用戶資料，選用的資料庫軟體將有 MySQL [24] 和 OpenLDAP [26] 兩種。

5.1. 實驗環境

伺服器的硬體設備為 Intel Pentium®4 CPU 3.40GHz，768MB 的 RAM，作業系統為 FreeBSD5.4-RELEASE。在選定開發系統的 SIP Proxy server 軟體為 OpenSER（The Open source Source SIP server） [25] ，是一個以 C 寫成的 Free software。而架設 Media Proxy server 來提供 Voicemail 服務的部份，選用由 Iptel 所開發的 SEMS (SIP Express Media Server) [21] 。而在做為儲存用戶基本資料的

資料庫選擇了兩種，分別為 MySQL 和 OpenLDAP。之後在 5.2 章節效能測試裡，會去分析這兩種資料庫在系統運作時，將會對系統產生的負擔會有多少。

5.1.1. OpenSER

OpenSER 是一個可以用來架設 SIP 代理伺服器（Proxy server），或者註冊伺服器（Registrar Server）與轉向伺服器（Redirect Server）所用。OpenSER 與 Iptel 所提出的 SER 相似，但在 OpenSER 裡有些模組的應用函式，SER 並不支援（例如:AVPops 模組裡，SER 並無提供儲存資料的函式），所以開發本系統的 SIP Proxy server 選定為 OpenSER。安裝方式請參考附件 A。

OpenSER 的設定檔稱為 openser.cfg，它是一個純文字檔，算是個 lex 和 yacc 的輸入格式，在 OpenSER 的 source 裡可以找到 cfg.lex 和 cfg.y 這兩個的 lex 和 yacc 的 source，因此 OpenSER 也不算是一個 shell script，而是一個必須符合 OpenSER 它自訂的 Token rule 文字檔。另外 OpenSER 搭配使用 dlopen 來動態載入模組，在 openser.cfg 中決定所要載入的系統模組或自己撰寫的模組。

設定檔主分四個部份 1. Global parameter 2. External module loading 3. Module parameters 4. Routing blocks。了解這四者的用法，就能讓輕易架設 SIP Proxy server 的功能。

Global parameter：主要是設定整個設定檔的執行模式與相關環境設定。例如設定 Port=5060，在決定執行 OpenSER 的信令處理在 Port 5060 的位置。

External module loading：載入所需的 modules，以提供程式去處理。

例如：

loadmodule "/usr/local/etc/lib/openser/modules/usrloc.so"

上面敘述就是載入 usrloc 的模組，使用者可自行在模組目錄（/usr/local/etc/lib/openser/modules）中，找出所要的模組。

Module parameters：指定每一個模組的相關參數，針對載入的模組去做參數設定。

例如：

modparam("usrloc", "db_url", "mysql://user:passwd@host/dbname")

表示對 usrloc 模組的參數”db_url”內容設定為"mysql://user:passwd@host/dbname"。指定要連結的資料庫路徑和可連線的使用者帳號及密碼。

Routing blocks：在 OpenSER 接收到 SIP message 時候去做對應的處理。所有的 SIP message 都會先由 route{}開始進入程式中。而 route{}內依據收到的信令去做指定的處理，例如收到 BYE 的信令，則進行通話結束的步驟。同時可自行設定其他特別處理的程式區塊，這類似於平時撰寫程式時的副程式。通常命名方式為 route[1]、route[2]等，可做其他延伸的應用。

例如:

route {

if (method==”BYE”){

.....route[1]其他處理或判斷.....

};

}

route[1]{

.....應用處理.....

}

每一個 SIP message 都代表要做某一個動作，因此判別每一次的 SIP message 所要求的 method，在 OpenSER 裡給予適當的流程處理，才能架構好 SIP Proxy server。

5.1.2. SEMS

由 **Iptel.org** 提出的 SEMS，是個專門用來架設多媒體服務伺服器的軟體。它實現了多項 VoIP 的技術，像是語音信箱，多方會議等等功能。都可讓使用者去佈置所要的多媒體環境。在它的程式核心中，把各個服務給模組化，使管理者在系統開發上能夠輕易的進行程式撰寫。所以為目前架設 VoIP 多媒體服務伺服器相當受歡迎的一套免費軟體。SEMS 的安裝請詳見附件 B。

SEMS 主要常用的功能有 Voicemail、Announcement、Conference、IVR 等等。每個功能該何時啟動，都是依據 SIP Proxy server 傳來的信令，再去進行處理。本章謹描述 Voicemail 的功能及運作過程；其他進階功能，請參考 SEMS 手冊。

Voicemail（語音信箱），管理者可以藉由這個服務，設定偵測到當受話方無註冊的情況下，啟動 Voicemail 的功能。如圖 20 所示，這是一封由 Alice 傳送給 Bob 的一封 Voicemail，Bob 的信箱位址為 Changyu@ms11.voip.edu.tw，紅圈處的

附檔”message.wav”則是 Alice 給 Bob 的語音內容。

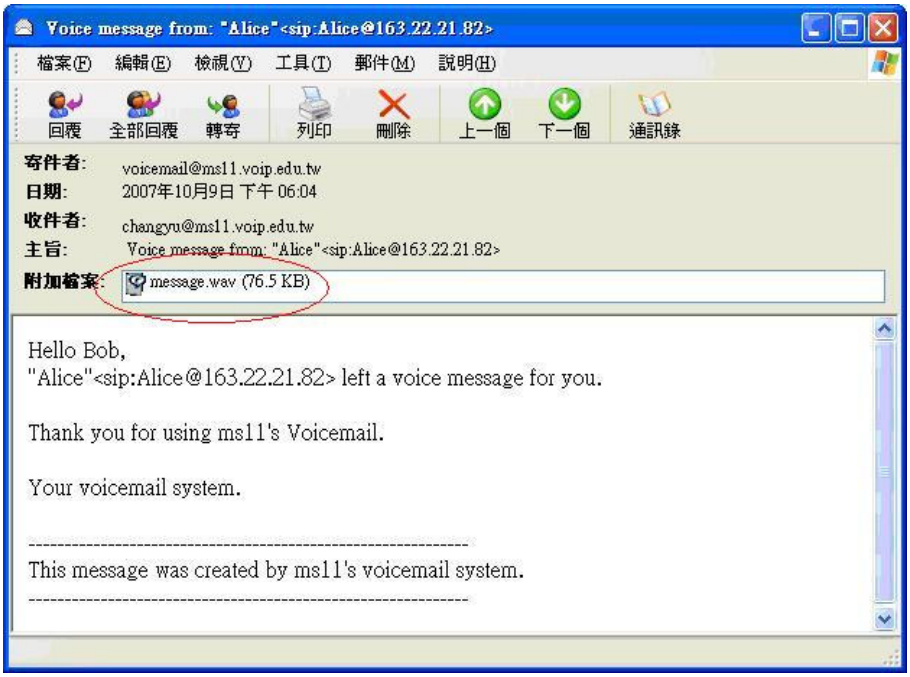

圖 20　Voicemail 範例

由 SIP Proxy server 傳來的信令送給 SEMS 時，SEMS 會啟動 Voicemail 的功能進行語音錄製，錄製好後才進行發送。圖 21 中描述 Voicemail 的發送過程，當 SIP Proxy server 收到 Bob 的 INVITE 請求時，SIP Proxy server 將會把 INVITE 導向給 SEMS 去處理。而 SEMS 根據傳送過來的信令，去執行 Voicemail 的功能，在 Bob 和 SEMS server 之間建立連線來儲存語音訊息。之後 SEMS 將收到的語音訊息存成 wav 檔，夾帶在信件裡，再利用 SMTP server 傳送出去。

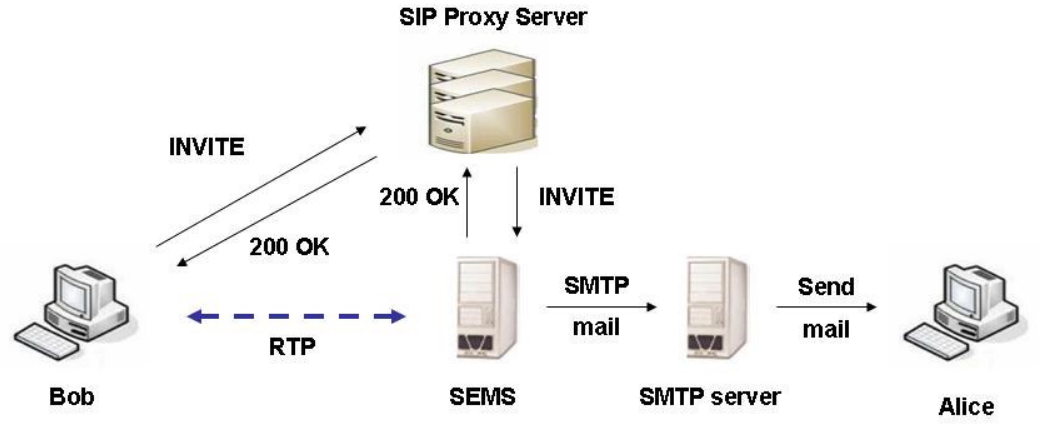

圖 21　SEMS 發送 Voicemail

SEMS 與 SIP Proxy server 之間傳遞信令主要是利用 Sockets，當 SIP Proxy server 收到 UA 的 INVITE 信令時，會將信令經由 Sockets 傳達給 SEMS。圖 22 則是信令傳遞的簡圖。

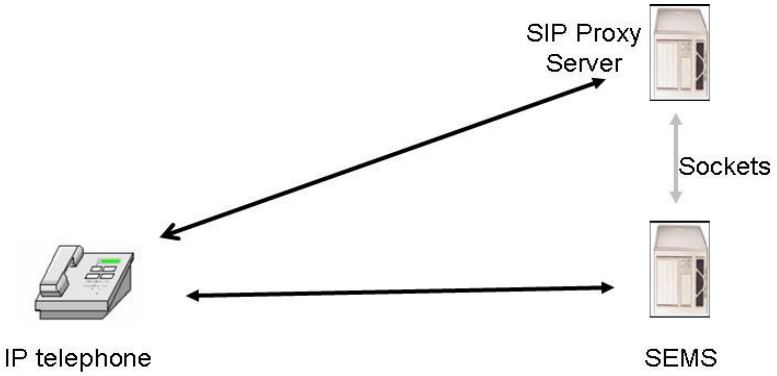

圖 22　SEMS 的信令傳遞

5.1.3. MySQL

MySQL 具有高階商業資料庫軟體的功能，主要以 Client 和 Server 的架構去管理大量資料。它的設計很適合管理網站資料庫程式所需，且支援多種 SQL 查詢，包括有結合查詢字串、多重資料表的同時更新或刪除、巢狀查詢。所謂 SQL

（Structured Query Language）是一種為了存取或操作關聯式資料庫所設計的語言。主要解決以往程式與資料庫相依性過高的問題，透過 SQL 存取資料庫使得後端資料庫較容易更新，達成資料庫的獨立性。而前端的介面也可獨立使用不同的開發工具，例如 PHP、ASP.net 等。如此將資料層分離出來再儲存到資料庫伺服器，對於維護與安全都更有保障。而 MySQL 在目前在操作關聯式資料庫的 Open source 是很受歡迎的。在本論文中，在選擇儲存用戶的基本資料及好友名單時，MySQL 這套軟體將是選擇之一。

5.1.4. LDAP

由 IETF 所提出的標準協定 Lightweight Directory Access Protocol（LDAP）[3,12] ，是一個以目錄化做為資料型態的資料處理系統。而目錄服務（Directory Service），廣義的說，資料的儲存機制是以目錄結構概念去設計。經由通訊協定的存取，再藉由目錄服務資訊的提供，可將網路上原本互不相關的兩個端點，進行連結及資訊或服務交換。它同時可為各式的資源與資料去做命名、說明、搜尋、存取、以及保護該資源等等，使各個網路服務、應用程式與使用者間進行相互運作。舉例來說，舊式的電話簿就是很典型的目錄服務例子。LDAP 起源自 X.500 標準，是個屬於「輕量級」版本的 DAP（Directory Access Protocol）；會有輕量級的稱呼，主要是 DAP 本為網路目錄服務標準 X.500 的一部分，將目錄定義成階層式架構以存放大量的資訊。而 LDAP 簡化 X.500 的傳輸方法， 捨棄 X.500 的 OSI 的七層架構而只採用了 TCP/IP 該層，如圖 23。

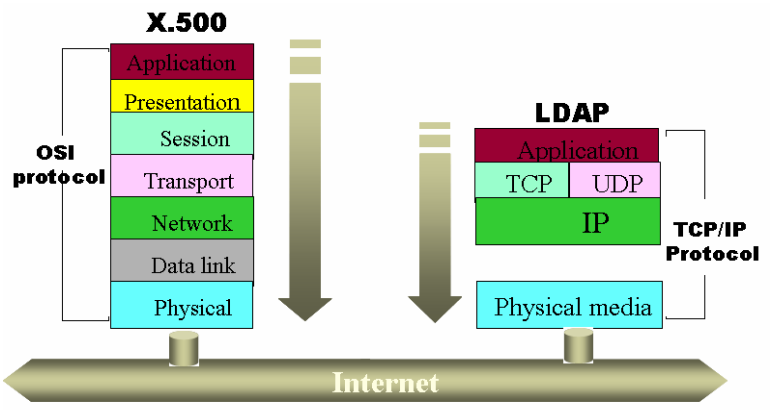

圖 23　X.500 和 LDAP [3]

這樣的改變能有效的快速查詢，並以樹狀結構的資訊管理方式和分散式的佈署架構，方便組織資料的整合。而 LDAP 的目錄，就像是樹狀圖般的分支，從根目錄（root）開始，國家、組織、個人，分散於多個伺服器中。當伺服器接收到查詢或者修改等指令時，針對適當路徑去執行，就可成功地完成任務。圖 24 是 LDAP 名錄資訊樹的一個例子。以樹狀結構依序將單位排序下來，各點都含有相關的屬性說明及資訊。圖中的 cn 表示顯示名稱，ou 代表所屬單位，dc 則是所屬區域，dn 是紀錄的位址。從這些資訊可瞭解到 Alice 屬於 ncnu.edu.tw 的網域內，並且為單位學生中的一人。

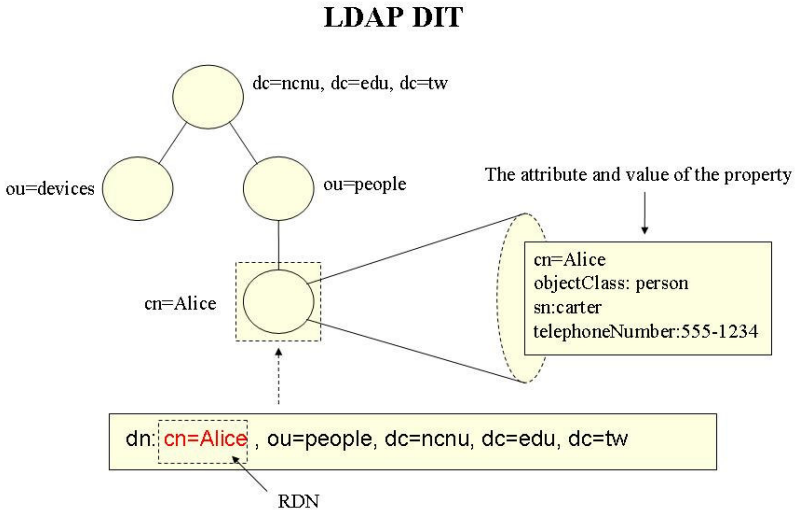

圖 24　LDAP 的 DIT Tree

LDAP的應用已成為各家廠商如何管理的重要理念，例如利用LDAP建構出

公司的郵件共同通訊錄功能，讓員工能夠透過關鍵字查詢，就可以輕易快速搜尋同事、客戶、或特定其他公司的聯絡人的聯絡資訊，可省去不必要的人力和時間。而在這篇論文中，除了用到MySQL作為後端儲存資料庫，我們也將利用OpenLDAP軟體，去儲存用戶資訊以及他們的好友名單。

5.1.4.1.OpenLDAP

OpenLDAP 是個被廣泛使用、開放原始碼、與 LDAPv3 [12] 相容的伺服器。能輕易的去實作 LDAP。使用在多種平台上，包括有 Linux、Solaris、Mac OS、FreeBSD 等。在新的版本（OpenLDAP-2 系列）支援了 LDAPv3 的實作，這可讓用戶端引薦至其他 LDAP 伺服器的能力，且具備有經由 LDAP 操作發佈伺服器之綱要（schema）的功能；這可讓用戶端得以在進行搜尋之前輕易得知伺服器的綱要。而在字串上使用 UTF-8 字符（RFC2253）去符合國際標準。同時改進認證所用的憑證的安全性與彈性，以及經由 SASL 和 SSL/TLS 傳送資料。

在 OpenLDAP 的設定檔裡分為幾個部份，Schema、Log record、SSL/TLS、Database area、ACL。

Schema（綱要）：在設定 LDAP 伺服器的第一步就是決定名錄應該支援哪種綱要。名錄該儲存哪些資訊、伺服器環境該如何命名這都需去定義。而針對應用程式的需要，在為名錄選用綱要時就須納入考量，所有必要的 attributeType 與 objectClass 都會定義在該屬綱要檔。表三舉出常用的綱要檔範例。

表 3　　OpenLDAP 綱要檔

Schema	Descript
Core.schema	OpenLDAP core (required) RFC 2251~2256

Inetorgperson.schema	InetOrgPerson (useful) RFC 2798
Cosine.schema	Cosine and Internet X.500 (useful) RFC 1274
Misc.schema	Assorted (experimental)
Nis.schema	Network Information Services RFC 2307

Log record（記錄檔）：log 紀錄可以將執行期間的資訊記錄下來。參數 loglevel 所對應的設定值是個整數，用來表示哪些類型的資訊應該記錄在系統日誌中。而這些數值是由 2 的指數值去組成。例如 8+32+256=296（參考表 4），296 表示將連線管理、搜尋過濾器的處理和連線操作的統計紀錄。管理者可根據除錯資訊進行程式的 Debug。

表 4　OpenLDAP 登錄等級

等級	所記錄的資訊
-1	所有登錄資訊
0	不登錄任何資料
1	追蹤函式呼叫
2	封包處理除錯資訊
4	大量追蹤除錯資訊
8	連線管理
16	封包的送收
32	搜尋過濾器的處理
64	組態檔的處理
128	存取控制清單的處理

256	連線、操作及結果統計
512	結果傳回用戶端的統計
1024	與 shell 後端的通訊
2048	印出項目剖析除錯資訊

SSL/TLS：OpenLDAP 在連線時可經由 SSL/TLS 傳送簡易認證。在進行任何操作之前，LDAP 可以協商出一個經過加密的傳輸層。另外 OpenLDAP 提供幾個選項，可用來設定與 SSL 及 TLS 有關組態，表 5 中是針對 TLS 三個參數去加以描述。

表 5　OpenLDAP 的 TLS 參數說明

參數名	描述
TLSCipherSuite	指定伺服器將會接受哪些密碼或加密法
TLSCertificateFile	設定 LDAP 伺服器的私有金鑰檔案
TLSCertificateKeyFile	設定 LDAP 伺服器的公開憑證

Database area（資料區塊）：OpenLDAP 組態檔中，需去定義資料庫區段，每個資料庫區段是用來定義其目錄分割區。表六是對資料庫的參數 ldbm、bdb、passwd、shell 的說明。

表 6　OpenLDAP 資料庫參數

資料庫參數	描述
ldbm	使用 GNU Database Manager 或 Sleepycat 的 Berkeley DB 軟體。
bdb	使用 Berkley DB 4 資料庫管理程式，廣泛用於索引與快取技術提昇效能。

passwd	系統密碼檔。
shell	使用替代或外部的資料庫。

ACL（Access control list）：存取控制清單，基本上它們用來定義『誰』(who) 有『權力』(right) 存取『什麼』(what)。

舉例來說：

Access to *

By * read

表示允許任何使用者都可進行讀取動作，*代表任何人。

Access to attrs=userPassword

By self write

By * auth

表示使用者可變更自己在名錄中的密碼，並且限制 userPassword 只用來認證。

5.2. 資料存取

防禦垃圾語音的系統裡，儲存好友名單的資料或是用戶資料的查詢，如圖25所示。SIP Proxy server與資料庫之間如何傳遞，是系統中關鍵的處理部分。如何把啟用應用服務時所需要的資料（例如:用戶分機號碼），透過資料連線從資料庫取出後，再回傳給SIP Proxy server；或者是SIP Proxy server所提供的資訊，如何藉由資料連線去執行資料庫存取，這些動作的效能，是垃圾語音防禦系統能否有效運作的重要關鍵。在資料庫的套件上，我們選用MySQL和OpenLDAP兩種不同協定的資料庫去設計，接下來就是設計如何建立存取連線。5.2.1和5.2.2兩節描述不同的資料連線存取方式。

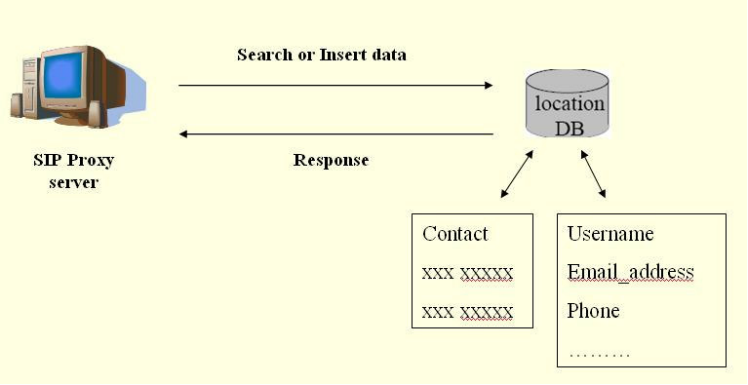

圖 25　資料存取

5.2.1. OpenSER 的 module

在 OpenSER 的眾多模組中，AVPops 模組用在 SIP Proxy server 和 MySQL 之間互相傳遞資料或是修改資料上。事先在模組的參數設定裡，設定與 MySQL 建立連線的路徑，之後透過 AVP 指標將從資料庫得到的資料去做特定的處理，例如 Request URI 的修改、新增或修改資料於資料庫中等等。有關此模組的詳細說明可見附件 C。

在本系統中，將利用 AVPops 模組中專門操作資料存取的 Function 去完成系統中要求的功能。而各功能所使用到的 Function 說明如下：

avp_db_load(“source”, “name”)：查詢指定的資料。用來讀取用戶的電子郵件信箱跟分機號碼，同時也用來讀取用戶好友名單的內容。

avp_ db_query(query,[dest])：執行 SQL 的動作。用此 Function 去完成新增資料於好友名單裡。

avp_pustto(destination,name)：修改 Request URI。當要來電轉至分機號碼時，

可利用此 Function 去進行 Request URI 修改為指定目的地。

avp_check(name,op_value)：檢查資料內容，運用在進行陌生電話的來源與好友名單中內容去比對時，做為過濾用。

5.2.2. 外部程式的呼叫

在OpenSER裡，目前並未提供與OpenLDAP資料存取的模組，須以外部程式呼叫去執行OpenLDAP。利用OpenSER中專門呼叫外部程式的模組Exec（詳見附件D）去執行事先撰寫好Unix Shell Script。處理的方法是當SIP Proxy server需要呼叫OpenLDAP去取得資料時，會連同OpenSER的環境變數傳送給外部程式。所謂環境變數是由OpenSER所提供針對SIP messages內的一些欄位值，例如:Request URI、Caller的username或contact address等等。而這環境變數我們將只會用到Caller和Callee的SIP URI，原因是在呼叫資料庫時，欲讀取某用戶的資料時，需要有用戶的名稱才能進行查詢。而在系統裡，外部程式所用程式語言為Unix Shell script。當Unix Shell script收到Callee的SIP URI時，SIP URI的username將為查詢用戶資料的參考依據，讓程式可決定回傳所要的資料，例如是要callee的分機號碼或是電子信箱等。也就是說Unix Shell Script作為OpenSER和OpenLDAP之間溝通的橋樑，如圖26。而同時也撰寫一份以MySQL作為資料庫的垃圾語音防禦程式，以比較其執行效能。在後面效能測試的部份將評估MySQL和OpenLDAP兩種不同的資料儲存型態，分析系統的負荷量。

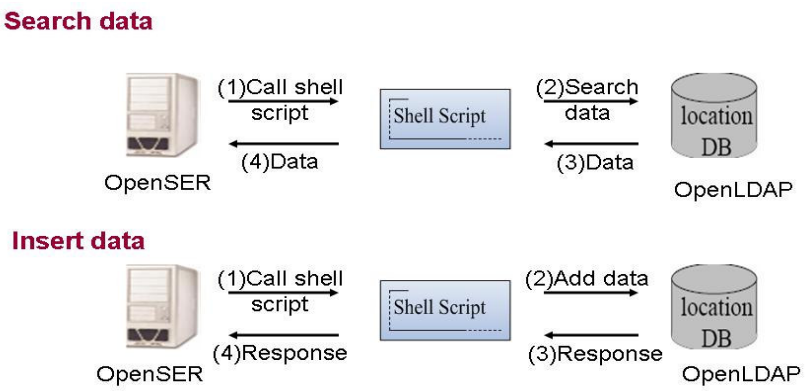

圖 26　外部程式呼叫

5.3. 效能測試

抵禦垃圾語音的 SIP.edu 系統裡，將對儲存資料所用的兩種資料庫軟體進行效能測試，評估對系統的負擔。

5.3.1. 測試工具

對系統的效能分析，使用SIPp [27] 這套軟體來進行壓力測試。SIPp是一套open source的測試軟體，可以大量發送SIP訊息給終端設備，再根據UAC（User Agent Client）和UAS（User Agent Server）通話的建立和結束，來計算所要測試的通話數，其中成功或失敗的連線。如圖27，UAC和UAS之間可互傳信令。另外可自行設定所要測試的參數去完成所要的實驗目標。同時它也可支援XML scenario去操作更高階的實驗。

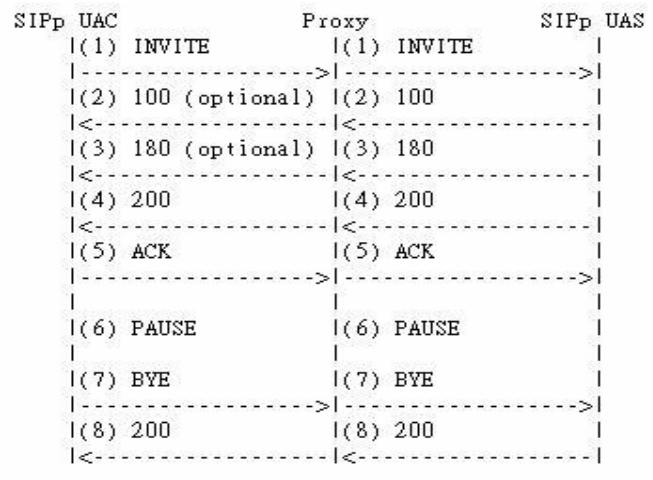

圖 27　SIPp 信令傳遞 [27]

5.3.2. 實驗環境與條件

在實驗環境部分，架設各服務的軟體如下:

- SIP Proxy server：OpenSER1.2.1
- Voicemail server：SEMS-0.10.0 rc2
- Database：MySQL-5.0.22、Openldap-server-2.3.27
- Operating System：FreeBSD 5.4

硬體的環境上,因架設 SIP Proxy server 和 Voicemail server 及 Database server 三種服務皆為同一台機器，所以硬體規格相同。

- CPU：Intel Pentium®4 CPU 3.40GHz
- RAM：768MB

在 5.2 章節提到資料存取方式有所不同，以外部程式呼叫或是 OpenSER 的

模組處理，系統效能必將有所不同。所以在不同的資料呼叫方式，規劃四個實驗環境。分別如下：

1. 無修改過的 SIP 設定檔
2. 以 OpenSER 所提供的模組呼叫 MySQL
3. 使用 Unix Shell Script 呼叫 MySQL
4. 使用 Unix Shell Script 呼叫 OpenLDAP

第1個方式，主要是用來測量無使用任何外加服務，也無資料庫的支援或語音信箱功能時的系統效能，作為比較的基準。接著進一步觀察不同的系統實驗環境後，每個測試出來的結果將會有多少的負荷。而每個實驗所要測試的SIPp的參數如下：

1. Total calls：20,000 calls
2. Max Allowed Rate：10,000 calls per second
3. Duration time：30,000 ms

上述的數值，主要目的是測試每段實驗在改變Concurrent Calls數目時，通話的失誤率變化情形，進而瞭解每個機制的效能。每次實驗固定的參數是總通數為兩萬通，Call Rate為每秒一萬通，每個通話持續30秒。唯一要改變的是Concurrent Calls的部份，我們從零開始，以200通的單位慢慢累加，直到實驗到8000通為止，並紀錄每段實驗所得出的結果。

5.3.3. 效能比較

在支援抵禦垃圾語音的系統中，會不斷的去查詢資料庫中的資料，包括有無註冊情況時的服務選項、用戶是否啟用本系統的功能、好友名單的檢查等。每次的查詢或是新增動作，對系統而言，都是負擔。為了準確分析每次將進行幾次的查詢或是新增資料時，通話數的量對系統承受度將會是多少，這是需仔細分析。所以除了三種不同存取方式的實驗環境，還必須再去設立幾個實驗條件。

實驗1: 初始的 OpenSER 設定檔。

實驗2: 用戶無註冊情況下，轉校園分機服務（一次的查詢動作）。

實驗3: 用戶無註冊情況下，依是否有使用 SIP.edu 功能轉校園分機或 Voicemail 服務（兩次的查詢動作）。

實驗4: 用戶有註冊情況下，使用好友名單查詢功能（兩次的查詢動作，若撥打方不在好友名單內將會是三次查詢，因為會啟用用戶的 Voicemail 服務，將需要用戶的電子郵件信箱）。無註冊則只會回「404 Not Found」訊息。

實驗5: 用戶有註冊情況下，使用好友名單查詢功能並擁有自動新增資料至好友名單的服務（兩次的查詢動作和一次新增動作，若撥打方不在好友名單內將會是三次查詢。因為會啟用用戶的 Voicemail 服務，將需要用戶的電子郵件信箱）。無註冊則只會回「404 Not Found」訊息。

實驗6: 用戶有註冊情況下，使用好友名單查詢功能並擁有自動新增資料至好友名單的服務（兩次的查詢動作和一次新增動作，若撥打方不在好友名單內將會是三次查詢，因為會啟用用戶的 Voicemail 服務，將需要用戶的電子郵件信箱）。而用戶無註冊時，則擁有實驗 3 的服務功能。

實驗 1 的作用也是用來作為比較的基準，其他五個實驗則是測試在不同的資料存取次數對系統的影響。接下來針對每個階段的實驗再以不同的資料存取方式去進行分析。

圖 28 是以三種不同實驗環境去進行實驗 2。由圖看出，只做一次查詢的資料處理，外部程式呼叫 OpenLDAP 對系統負擔比同樣以外部程式呼叫 MySQL 少很多。且在 Concurrent Calls 為 6400 通左右時，外部程式呼叫 MySQL 的方法已經到達失誤率 50%，失誤率的上升起伏劇烈且呈現階梯狀。相較其他兩個呼叫方式，效率差異甚大。

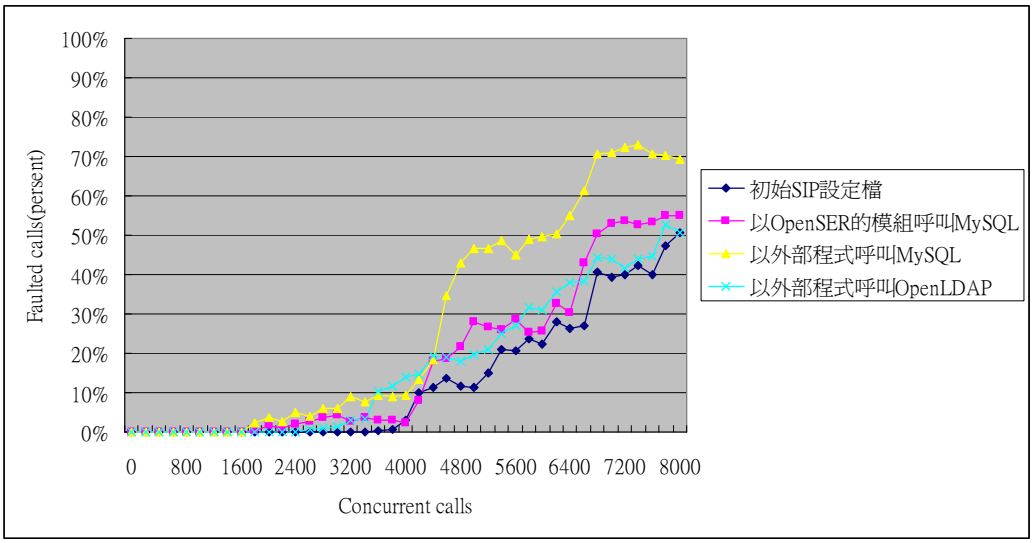

圖 28　實驗 2 的效能測試圖

圖 29 和圖 30 是實驗 3 和實驗 4 的壓力測試圖，是針對兩次或三次以上的資料查詢動作時，伴隨著 Concurrent Calls 增加而紀錄失敗的通話數。以外部程式呼叫 OpenLDAP 可由圖中可看出，失誤率的起伏並不像使用 MySQL 呈現階梯爬升，而是平緩的增加。直到測試 Concurrent Calls 為 8000 通時，失誤率到達 50% 左右。

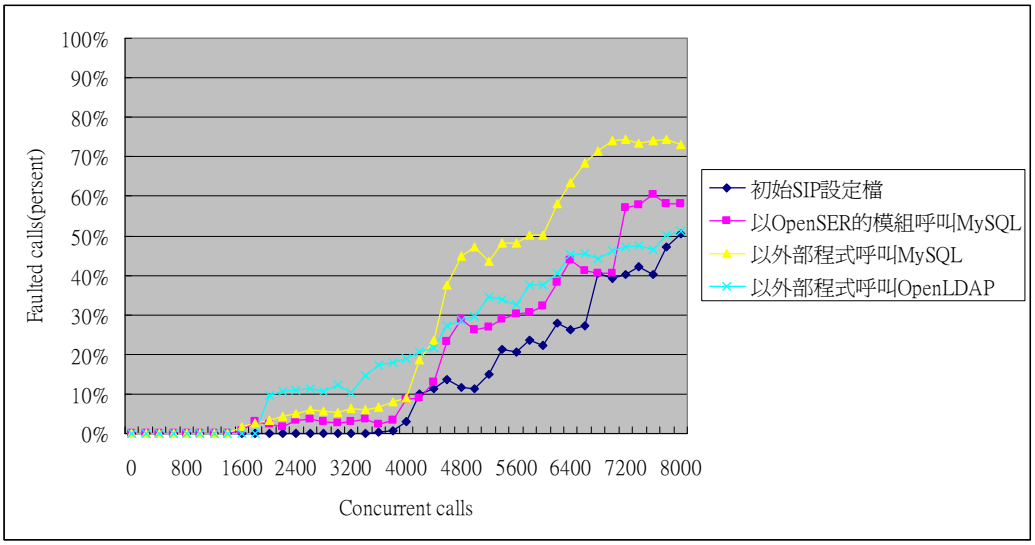

圖 29　實驗 3 的效能測試圖

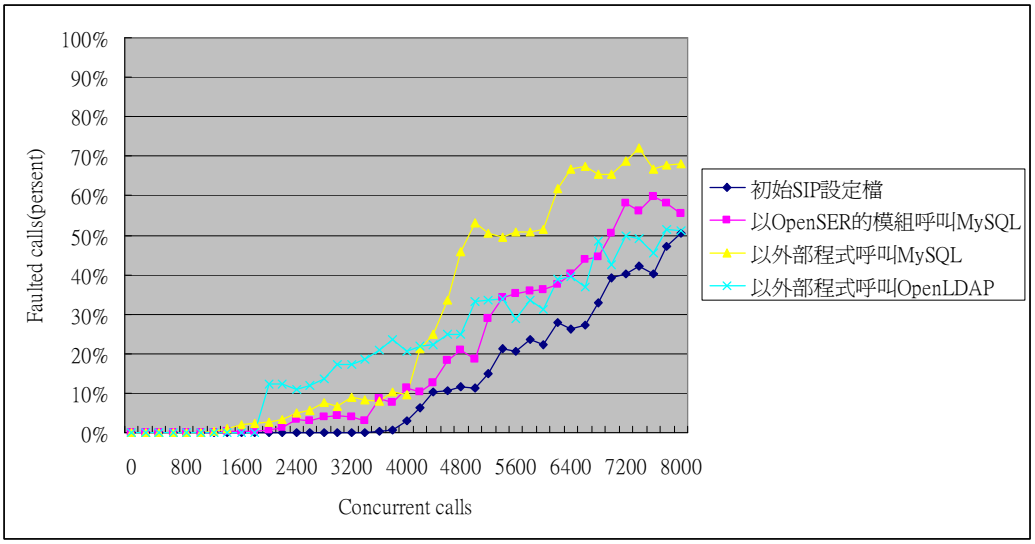

圖 30　實驗 4 的效能測試圖

圖 31 和圖 32 各代表實驗 5 和實驗 6，這兩個都有新增資料的動作。除了以 OpenSER 模組呼叫的方式在 Concurrent Calls 為 7000 通左右時，失誤率才衝破 50%。其他兩個以外部程式去呼叫的做法，在 Concurrent Calls 為 1000 通左右就到達 50%。很明顯有了資料新增的動作後，對系統具是很大的負荷。

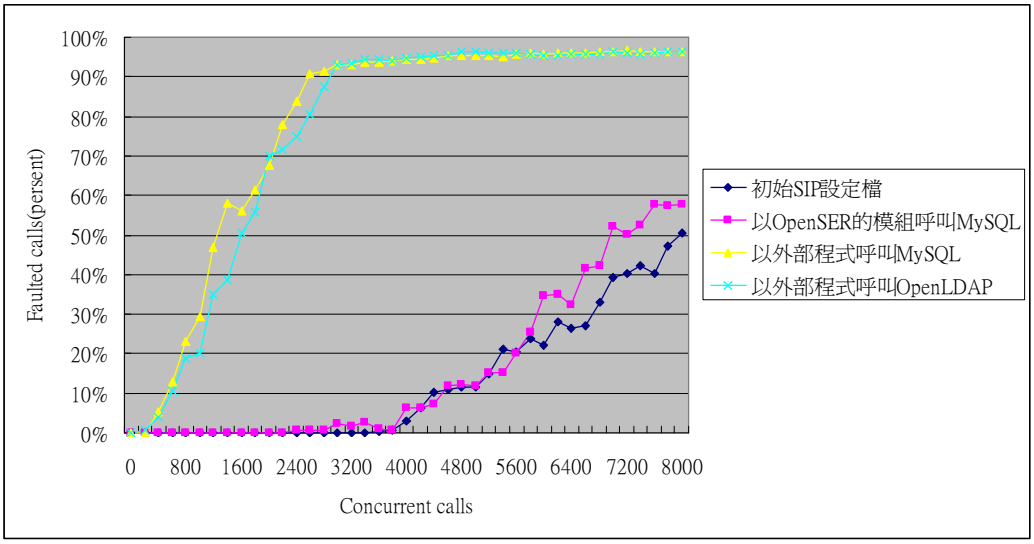

圖 31　實驗 5 的效能測試圖

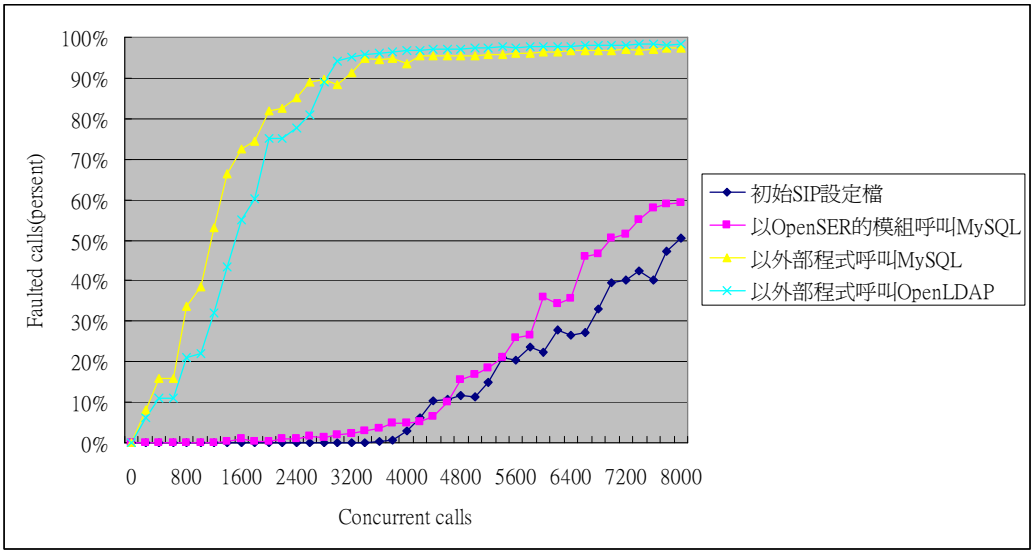

圖 32　實驗 6 的效能測試圖

最後根據不同的實驗環境，從 6 個實驗彙整出每個不同存取機制的壓力測試圖。圖 33 是以 OpenSER 所提供的模組去呼叫 MySQL 的壓力測試，而圖中的六個曲線分別是不同的資料存取次數所得出的結果。初始的 SIP 設定檔在 Concurrent Calls 設定為 8000 通左右時，失誤率達到 50%。而其他存取動作的失誤率皆落在於 6600~7200 通左右。

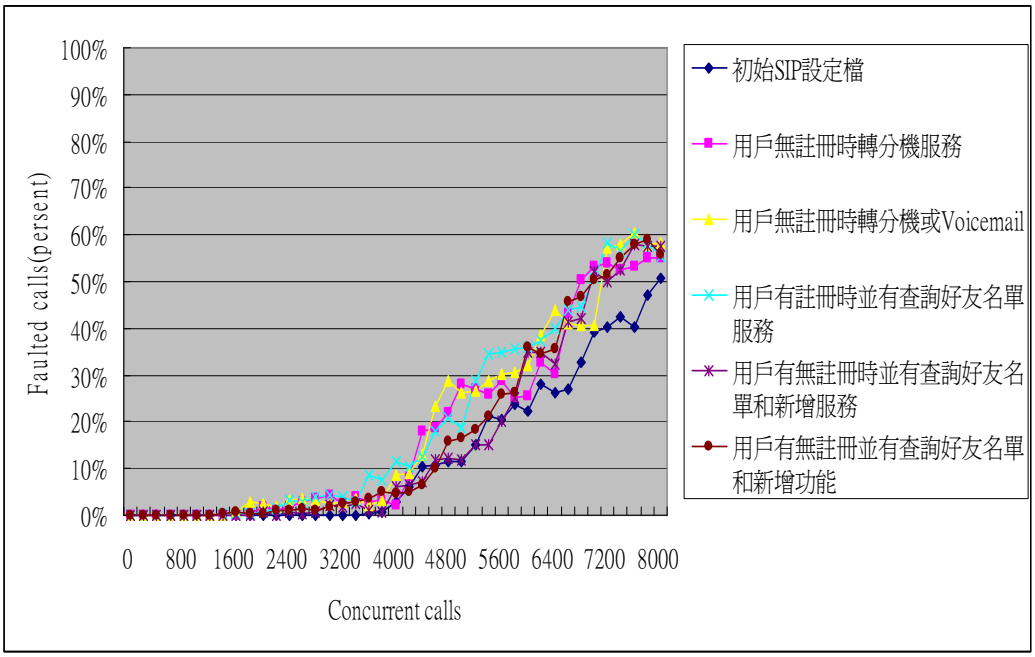

圖 33　OpenSER 模組存取 MySQL 效能測試圖

圖 34 是以 Unix Shell Script 去執行 MySQL 資料存取動作，圖中六個曲線可看出單純只有查詢動作時，當 Concurrent Calls 為 4000 通前，失誤率還不算高，但是一過 4000 通後，在將近 5000 通時，失誤率就到達 50%。以外部程式呼叫 MySQL 進行只有查詢動作時，系統在 4000 通時，失誤率不高。但是若加入新增資料的動作，不到 1000 通，就打破 50%的失誤率。可見新增資料的動作，對系統而言是很大的負荷。

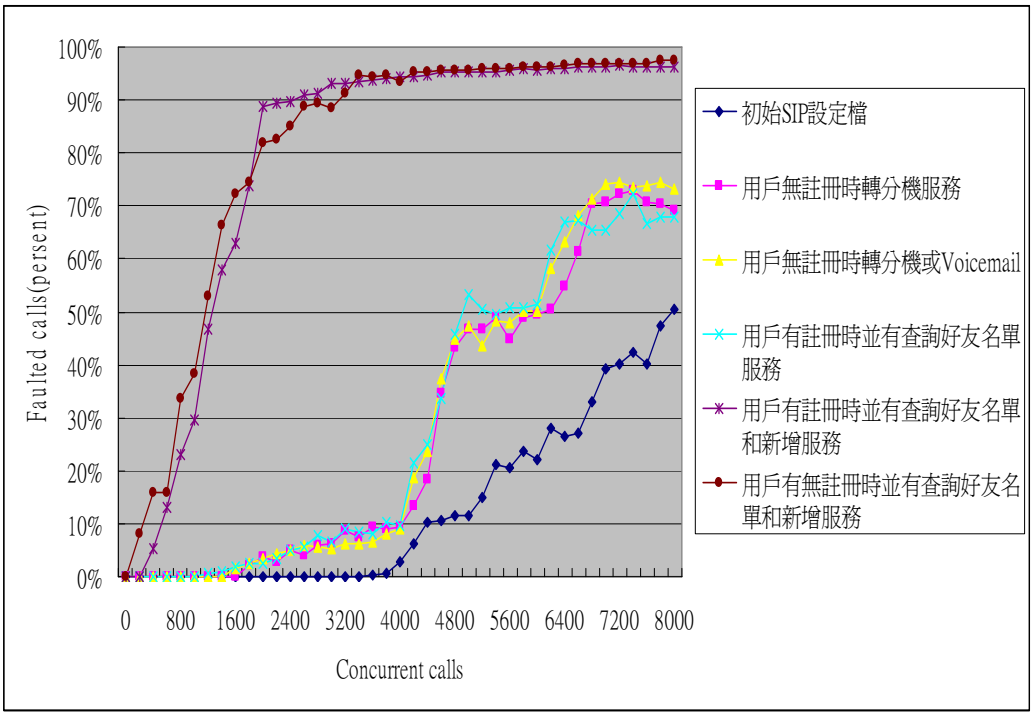

圖 34　以外部程式存取 MySQL 效能測試圖

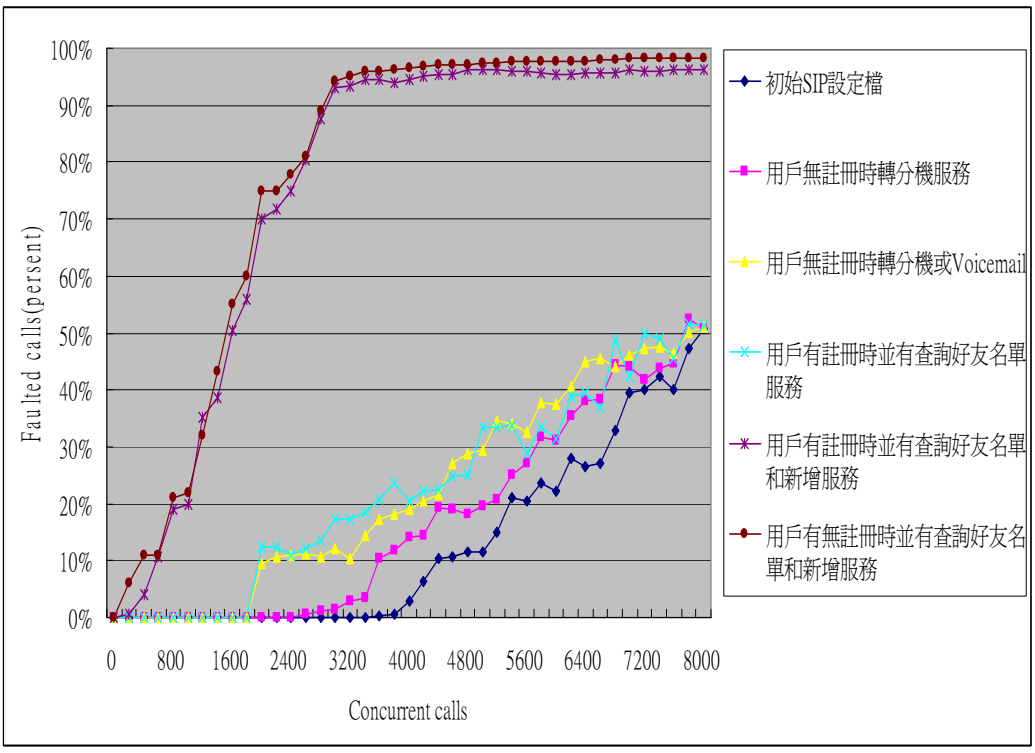

圖 35　以外部程式存取 OpenLDAP 效能測試圖

圖 35 則是以外部程式呼叫 OpenLDAP 的壓力測試圖，圖中可看出系統只有查詢資料的動作上所產生的失誤率在 Concurrent calls 8000 通，才到達 50%。而

一次查詢資料的動作跟兩次查詢資料的動作，這兩者之間雖有差距。但是隨者 Concurrent calls 慢慢增加，逐漸接近。比起以外部程式去呼叫 MySQL 不到 5000 通則就打破 50%的失誤率，OpenLDAP 效能上比較優勢。但是在新增資料方面，雖然比起 MySQL 在 1000 通時到達 50%，而 OpenLDAP 在 1600 通時到達 50% 失誤率較好。但是對系統而言，都是負擔過重的。

從圖 32 來分析，三種不同的資料傳送方式去測試本系統的抵禦垃圾語音環境下。惟獨以 OpenSER 的模組去存取 MySQL 是效率最好的。但是只從去執行資料查詢動作來判斷，圖 28、29、30 顯示出 OpenLDAP 的失誤率明顯比 MySQL 還要低很多。雖然 LDAP 名錄服務具有高效能的查詢功能，但是新增資料的處理，以外部程式去執行，會顯的較差。MySQL 則是不管在新增或是查詢，效益上都不如 OpenLDAP。

6. 安全與隱私議題

利用好友名單抵禦垃圾語音，可減輕陌生電話對用戶的即時打擾。而依據用戶的回撥動作自動增加好友名單的資料，對用戶具有操作上的方便性。SIP Proxy server 和資料庫之間，這兩者在傳送資料的同時，尚需要進行資料傳遞時安全性的保護。若用戶的基本資料在傳送的過程中一經竊取，則對系統安全是個漏洞。同時用戶的好友名單一經知曉，則用戶不再存有任何隱私。

另一問題就在 Voicemail 的使用，系統對陌生電話的處理，是轉入用戶的語音信箱。但是若有人將大量預先錄製好的語音，不斷地傳送給各用戶。當用戶收到後，就啟動語音信箱服務。之後只要不斷地傳送語音給用戶，就相同於 DoS 的攻擊。對 Voicemail 的使用，將需要有較嚴格的限制來保護使用者。

7. 結論

7.1. 優點

針對垃圾語音的問題，我們提出一個過濾平台，事先參考了SIP.edu 的概念去結合IP 網路與傳統電話網路，使得用戶在聯絡上更能有效操作。而採用白名單的機制去設計出好友名單，讓每個用戶都擁有一份自己的名單，再根據名單內容，來降低收到垃圾語音的機率。對SIP.edu所衍生的問題，所尋求解決方法是利用好友名單的過濾，能確實的解決即時性的陌生電話打擾，讓用戶不再受干擾。在設計存取好友名單的方面，用LDAP 和MySQL 這兩種資料管理軟體。之後對資料的存取和呼叫，細分出幾個存取方法，去評估各方法對本系統的負擔。找出適合的存取機制。

7.2. 限制

在圖 32 中我們可看出，兩個以 Unix Shell Script 呼叫的方式，在接近 Concurrent Calls 設定為 3000 通時，失誤率就高達 90%以上。雖然 LDAP 理論上是個輕量級的資料存取方式，對系統的負擔較輕。但在每通電話都以 Unix Shell Script 呼叫的方式來處理時，失誤率跟以 Unix Shell Script 呼叫 MySQL 差不多。由此可知，利用外部程式來處理資料庫上的運作，必定會給系統很大的負擔。最後還是以 OpenSER 提供的模組才可完成效能好的系統。

8. 未來方向

我們得知以外部程式的方式去呼叫LDAP，會造成系統很大負擔。所以在未來，將檢討該如何修改LDAP的指令執行以及呼叫方式，來減輕系統的負荷，降低失誤率，實現LDAP的資料存取的高效能。另外，對Voicemail的使用需增加一項門檻，將參考IVR（Interactive Voice Response）技術，在別人想留言給用戶時，須先輸入系統要求的一串數字，這數字時有系統隨機產生，若有輸入成功，則才可做語音留言的動作。

參考文獻

[1]. Cameron Newham, Bill Rosenblatt, “Learning the bash Shell,” O'Reilly, Jan. 1998.

[2]. Daniel Collins, “Carrier Grade Voice Over IP,” McGraw-Hill, 2003.

[3]. Gerald Carter, “LDAP System Administration,” O'Reilly, Mar. 2003.

[4]. Hugh E. Williams, David Lane, “Web Database Applications with PHP and MySQL,” O'Reilly, May 2004.

[5]. William Stevenson, “SPAM,” Crossroads (The ACM STUDENT MAGAZINE) Issue 11.2 winter. 2004.

[6]. Paulson, L.D., “Spam hits instant messaging,” IEEE on Computer Society, Apr. 2004.

[7]. Dennis Baron, Jeremy George, Ben Teitelbaum, “The Internet2 SIP.edu Initiative,” http://www.internet2.edu/sip.edu/, Jun. 2003.

[8]. Fei wang, Yijun Mo, Benxiong Huang, “P2P-AVS: P2P Based Cooperative VoIP Spam Filtering,” IEEE on Wireless Communications and Networking Conference, Mar. 2007.

[9]. Feng Cao, Cullen Jennings, “Providing Response Identity and Authentication in Telephony,” IEEE on Availability, Reliability and Security, Apr. 2006.

[10]. Gary A. Thom, “H.323: The multimedia communication standard for local area networks,” IEEE Commun. Mag., Volume 34, pp.52-56 Dec. 1996.

[11]. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP : Session Initiation Protocol,” IETF RFC 3261, Jun. 2002.

[12]. M. Wahl, T. Howes, S.Kille, “Lightweight Directory Access Protocol (v3),”

IETF RFC2251, Dec. 1997.

[13]. N.J Croft, M.S Olivier, “A Model for Spam Prevention in IP Telephony Networks using Anonymous Verifying Authorities,” Information Security South Africa Conference (ISSA), Jul. 2005.

[14]. Robert MacIntosh, Dmitri Vinokurov, “Detection and Mitigation of Spam in IP Telephony Networks using Signaling Protocol Analysis,” IEEE on Advances in Wired and Wireless Communication**,** Apr. 2005.

[15]. S. Dritsas, J. Mallios, M. Theoharidou, G. F. Marias, D. Gritzalis, “Threat Analysis of the Session Initiation Protocol Regarding Spam,” International Performance Computing and Communications Conference (IPCCC) 2007, Apr. 2007.

[16]. So Young Park, Jeong Tae Kim, Shin Gak Kang,“Analysis of Applicability of Traditional Spam Regulations to VoIP Spam,” International Conference on Advanced Communication Technology (ICACT) 2006**,** Feb. 2006.

[17]. G. Gonzalez-Talavan, “A simple, configurable SMTP anti-spam filter: Greylists,” in Computers and Security, vol. 25, 2006, pp. 229–236, Feb. 2006.

[18]. Y. Rebahi, D. Sisalem, “SIP Service Providers and The Spam Problem,” In 2nd Wksp. Securing Voice over IP, Washington DC, Jun. 2005.

[19]. Zhijun Liu, Weili Lin, Na Li, David Lee, “Detecting and filtering instant messaging spam - a global and personalized approach,” IEEE on Secure Network Protocols **,** Nov. 2005.

[20]. Graylisting, http://www.greylisting.org/.

[21]. SEMS - SIP Express Media Server, http://www.iptel.org/sems/.

[22]. SpamArchive, “SpamArchive,” http:/ /www.spamarchive.org.

[23]. SPF, “Sender Policy Framwork,” http://www.openspf.org/.

[24]. MySQL, http://www.mysql.com/.

[25]. OpenSER - the Open Source SIP Server, http://www.openser.org/.

[26]. OpenLDAP, http://www.openldap.org/.

[27]. SIPp, http://sipp.sourceforge.net/index.html.

附件

A. 安裝OpenSER

1. OpenSER 的簡易安裝（simple installation）

本實驗的作業系統環境為 FreeBSD，所以接下來的安裝流程只說明如何在 FreeBSD 5.4 的環境下去進行安裝。其他的不同作業的安裝流程將不會介紹，有興趣的話可參考 OpenSER 的網站（http://www.openser.org）。

1.1 安裝 OpenSER 的方法:

1.1.1 使用 ports 安裝 OpenSER

在 FreeBSD 的作業系統下，有提供 ports 的安裝方式，使用者可以 /usr/ports/net/openser 的目錄下，輸入 make install 的指令後，系統會根據 distinfo 檔內的內容，去抓取 OpenSER 的軟體來進行安裝。

1.1.2 下載 OpenSER 檔案

可直接從 OpenSER 的官方網站直接下載檔案進行安裝，到下列網址去取得 OpenSER 的安裝檔案。

http://www.openser.org/pub/openser/1.2.1/src/openser-1.2.1-notls_src.tar.gz

按照下列順序進行安裝:

1. 首先切換到/tmp 的目錄下。
2. 輸入下列指令

   #fetch source file(上列網址)。
3. 下載完畢後進行解壓縮動作，輸入下列指令。

   #tar zxvf openser-1.2.1-notls_src.tar.gz
4. 解壓縮完畢後會產生一個叫做 openser-1.2.1-notls 的目錄，接下切換到此目錄下。

   #cd openser-1.2.1-notls

5. 接下來 compile source file。

   #gmake

6. compile 過後則進行安裝，安裝指令如下:

   #gmake install

1.2 安裝後的設定

安裝完畢後，則會出現幾個檔案，分別是:

執行檔

/usr/local/sbin/openser

設定檔

/usr/local/etc/openser/openser.cfg

模組檔

/usr/local/lib/openser/modules/*

1.2.1 複製 OpenSER 的設定檔

OpenSER 的設定檔存放在/usr/local/etc/openser 底下，我們可以先複製原始的 openser.cfg.default 成 openser.cfg，在進行修改，輸入下列指令來進行複製。

#cp openser.cfg.default openser.cfg

之後可以自行利用文字編輯器(如 vi)去對設定檔進行修改。

1.2.2 設定 OpenSER 的環境

修改好 OpenSER 的設定檔後，接下來要設定好啟動環境才能啟動 OpenSER 的運作。首先必須設定環境變數 SIP_DOMAIN，OpenSER 的環境變數的設定檔存放在/usr/local/etc/openser 底下，檔名為 openserctlrc。我們使用 vi 去編輯這份檔案。找到 SIP_DOMAIN 的參數後，把它修改成 Domain Name 或是主機的 IP address，在此我們設定為 Domain name。

SIP_DOMAIN=ncnu.edu.tw

1.2.3 啟動 OpenSER

在 OpenSER 裡有個專門支援啟動 OpenSER 的指令，叫做”openserctl”。它的使用方法如下:

openserctl start => 啟動 OpenSER

openserctl restart => 重新啟動 OpenSER

openserctl stop => 停止 OpenSER

啟動後，可以檢查是否有真的成功啟動。

#ps ax | grep openser

1.3 開機後直接執行 OpenSER

想要每次開機就會直接啟動 OpenSER，可以在/etc/rc.conf 下加入下行

/usr/local/sbin/openserctl start

1.4 參考文獻

OpenSER 網站

http://www.openser.org/mos/view/OpenSER-Documentation-Repository/

B. 安裝SEMS

1.SEMS 的安裝 (simple installation)

1.1. 下載檔案安裝 SEMS

從 Iptel.org 的網站直接下載檔案進行安裝，可到下列網址去取得 SEMS 的安裝檔案。

ftp://ftp.iptel.org/pub/sems/sems-0.10.0-rc2.tar.gz

接下來按照下列順序進行安裝:

1. 首先切換到/tmp 的目錄下。
2. 輸入下列指令

   #fetch source file(上列網址)。
3. 下載完畢後進行解壓縮動作，輸入下列指令。

   #tar zxvf sems-0.10.0-rc2.tar.gz
4. 解壓縮完畢後會產生一個叫做 sems-0.10.0-rc2 的目錄，接下來切換到此目錄下。

   #cd sems-0.10.0-rc2
5. 接下來 compile source file。

   #gmake

   compile 過後則進行安裝，安裝指令如下:

   #gmake install

1.2 安裝後的設定

安裝完畢後，會出現幾個檔案，分別是:

執行檔

/usr/local/sbin/sems

設定檔

/usr/local/etc/sems/sems.conf

Plug-in 檔

/usr/local/lib/sems/plug-in/*

音效檔

/usr/local/lib/sems/audio

1.2.1 複製 SEMS 的設定檔

SEMS 的設定檔存放在/usr/local/etc/sems 底下，我們可以先複製原始的 sems.conf.default 成 sems.conf，再進行修改。輸入下列指令來進行複製。

#cp sems.conf.default sems.conf

之後可以自行利用 vi 去對設定檔進行修改。

1.3 修改 SEMS 的設定檔

接下來就需要去修改 SEMS 的設定檔。因為我們架設 SIP Proxy server 的軟體是 OpenSER，而 SEMS 的預設是以 Iptel.org 提供的 SER 為預設設定，所以須對 SIP Proxy server 溝通的路徑做修改。

ser_socket_name=/tmp/ser_sock 把它修改成

ser_socket_name=/tmp/openser_sock

為了提供發送 Voicemail 的功能，也必須去更改 SMTP Server 的設定，修改下行。

smtp_server=mail 修改成

smtp_server=IP address(如 163.22.21.83)

1.4 修改 Voicemail 的發信範本

使用 vi 去修改 default.template，會發現下面內容:

subject: Voice message from: %from%

from: voicemail@%domain%

to: %email%

Hello %user%@%domain%,

%from% left a voice message for you.

Thank you for using <your domain>'s Voicemail.

Your voicemail system

把粗體字的%domain%的部份修改成要處理發送 Voicemail 的 Mail server。可以是 Domain Name 或是 IP address。在此我們修改為 163.22.21.83

subject: Voice message from: %from%

from: voicemail@163.22.21.83

to: %email%

Hello %user%@163.22.21.83,

%from% left a voice message for you.

Thank you for using <john.ipv6.club.tw>'s Voicemail.

Your voicemail system

1.5 啟動 SEMS

啟動 SEMS 之前，須先知道該台主機的網卡編號，可用 ifconfig 查出。

```
ms11[~]$ ifconfig
rl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=8<VLAN_MTU>
        inet 163.22.21.82 netmask 0xffffff00 broadcast 163.22.21.255
        inet6 fe80::240:f4ff:fee9:14ae%rl0 prefixlen 64 scopeid 0x1
        ether 00:40:f4:e9:14:ae
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
```

圖 1. 查網卡編號

我們可以看到這台主機的網卡編號是 rl0，接下來則是啟動 SEMS，輸入下列指令。

#sems –d rl0

參數-d 的意思是所我們所要使用的網路介面。下面圖 1.2 則是成功啟動的畫面。

```
ms11[~]$ sems -d rl0
Sip Express Media Server (0.10.0-rc2 (i386/freebsd))


Configuration:
        configuration file:  /usr/local/etc/sems/sems.conf
        Ser's unix socket:   /tmp/openser_sock
        our unix socket:     /tmp/sems_sock
        reply unix socket:   /tmp/sems_rsp_sock
        plug-in path:        /usr/local/lib/sems/plug-in
        daemon mode:         1
        local IP:            163.22.21.82
```

圖 2. 啟動 SEMS 的畫面

1.6 參考文獻

SEMS 網站

http://www.iptel.org/sems

C. AVPops module

AVPops 模組提供了各種 Script function 去實作與資料庫的存取控制或是查詢等動作。利用了 AVP 的指標去取得 SIP messages 所提供的 header，而 header 中所會用的資訊例如 Request URI、From URI、撥打方或受話方的使用者名稱等等。利用這些得到的資料可與資料庫去進行其他資料的查詢或存取。在操作資料庫上是個很強力的工具。

1. AVP naming format

在 AVPops 模組裡，都會宣告 AVP 指標參數的型態，之後才能給 Script function 使用。而宣告的語法格式如下：

$avp(avp_flags:avp_name)：

avp_flags：定義變數的資料型態，主要有兩種，分別為整數或是字串。如果設定為 i 或是 I 則表示為整數型態；若是設為 s 或 S 則為字串型態。

avp_name：設定好 avp_flags 之後，avp_name 就是設定變數的名稱。

範例：

$avp(i:12)：宣告此 AVP 變數指標為整數資料型態，而名稱為 12。

$avp(s:test)：宣告此 AVP 變數指標為字串資料型態，而名稱為 test。

2. Exported Parameters

宣告 OpenSER script 中 AVPops 模組的環境變數設定。

2.1. avp_url(string)：

宣告 AVPops 模組欲連結資料庫的路徑。

語法格式：

modparam("avpops","avp_url","mysql://user:passwd@host/database")

範例：

modparam("avpops","avp_url","mysql://openser:openserrw@localhost/openser")

2.2. avp_table(sting)：

宣告 AVPops 模組欲使用的資料表

語法格式：

modparam("avpops","avp_table","avptable")

範例：

modparam("avpops","avp_table","openser")

2.3. uuid_column(string)：

宣告 AVPops 模組去啟用資料表中的使用者 ID

語法格式：

modparam("avpops","uuid_column","uuid")

範例：

modparam("avpops","uuid_column","uuid")

2.4. username_column(string)：

宣告 AVPops 模組去啟用資料表中的使用者名稱

語法格式：

modparam("avpops","username_column","username")

範例：

modparam("avpops","username_column","username")

2.5. db_scheme(string)：

定義一個綱要來存取資料表中特定的指定資料。

語法格式：

db_scheme=name:element

name：$avp(name)

element：uuid_col=string

username_col=string

domain_col=string

value_col=string

value_type=(integer|string)

table=string.

modparam("avpops","db_scheme","scheme1:table=subscriber;uuid_col=uuid;val
ue_col=first_name")

範例：

modparam("avpops","db_scheme","phone_scheme:table=subscriber;uuid_col=u
uid;value_col=phone")

3. Exported Functions

3.1. avp_db_load(source,name)：

從資料表中以 source 的內容為索引去載入特定資料於 name 中。

語法格式：

source：username|domian|uri|uuid

name = avp_spec['/'(table_name|'$'db_scheme)]

avp_spec = $avp(avp_name)

範例:

avp_db_load("$ru","$avp(i:123)/$some_scheme")

說明：

以 Request URI 為索引，把摘要 some_scheme 所設定的指定資料載入 $avp(i:123)中。

3.2. avp_db_query(query,[dest])：

對資料庫做指定的存取動作，所得到的資料可存到 dest 中。

語法格式：

query：為 SQL function，例如 select、update、delete 等動作。

dest：$avp name

範例：

avp_db_query("select phone from subscriber where username=`alice`",$avp(user))

說明：

將資料表 subscriber 中 username 為 alice 的 欄位 phone 的值放入$avp(user)。

3.3. avp_pustto(destination,name)：

把 AVP name 的值放入 destination，而 destination 為 SIP message。

語法格式：

destination='$ru' ['/'('username'|'domain')]

範例：

avp_pushto("$ru","$avp(i:678)")

說明：

把$avp(i:678)的值放入 SIP message 中的 Request URI。

3.4. avp_check(name,op_value)：

檢查 name 和 op_value 兩值是否相同。

語法格式：

name=AVP name

op_value = operator '/' value

operator = 'eq' | 'ne' | 'lt' | 'le' | 'gt' | 'ge'

value = pseudo-variable | fix_value

fix_value = 'i:'integer | 's:'string | string

其中 operator 的意義如下：

eq =等於

ne =不等於

lt =小於

le =小於或相等

gt=大於

ge=大於或相等

範例：

avp_check("$fd","eq/$td/")

說明：

檢查$fd 和$td 是否相同，是的話回 true，否則 false。

D. Exec module

Exec 模組可在 OpenSER 的 script 裡去執行外部指令或者是外部程式。而外部程式可以為各種程式語言，例如 C、C++、Unix shell script、Perl 等等，以供管理者做為其他服務的應用。同時 OpenSER 也提供了環境變數可讓給 Exec 的 Exported Function 去處理，例如 SIP message 的資料。

1. 環境變數：

$rU：Caller 的 username

$ru：Caller 的 SIP URI

$fU：Callee 的 username

$fu：Callee 的 SIP URI

$si：Caller 的 IP address

$Ri：Callee 的 IP address

2. Exported Functions

2.1. exec_msg(command)

執行一個外部指令，執行過後的結果 OpenSER 並不會去處理，只會交由系統去另作處理。

語法格式：

command-外部指令。

範例：

```
exec_dset("echo $rU > /tmp/username.txt");
```

說明：

執行 echo 指令將 Caller 的 username 輸入到/tmp 目錄底下的 username.txt 檔案中。

2.2. exec_avp(command [, avplist])

執行一個外部指令，執行過後的結果將會存放於 avplist 中。

語法格式：

command-外部指令。

avplist-指定的 AVP 標籤。

範例：

```
exec_avp("echo $rU", "$avp(s:test)");
```

說明：

執行 echo 指令將 Caller 的 username 輸入到$avp(s:test)標籤中，以供 OpenSER script 所用。