

# 以行動隨意網路為基礎架構的 VoIP 服務平台之推播機制

## A Push Mechanism for an Ad-Hoc Network-Based Mobile VoIP Service Platform

Jen-Jee Chen, Yu-Li Cheng,

Yu-Chee Tseng

國立交通大學資訊工程學系

{yctseng,chencz}@csie.nctu.edu.tw,

yulicheng.eic94g@nctu.edu.tw

Quincy Wu

國立暨南大學通訊工程研究所

solomon@ncnu.edu.tw

### 摘要

我們建立一個整合行動隨意網路 (Mobile Ad-Hoc Network) 及行動網路的架構，實作行動 VoIP (Voice over IP) 服務，因為使用行動隨意網路做為架構的基礎，這個系統具有簡單架設、結構彈性的優點，並提供使用者具移動性、彈性、以及服務無間隙功能的語音服務。但是，當系統一旦啓始，因為閘道器對外利用行動介面撥接、連結網際網路，而且必須一直保持對外的通訊，所以對系統而言無論在電力和價格上都是個負擔，開得越久，網路存取費用收得越多，電力也消耗越快，考慮到語音服務的需求並非一直存在，我們提出一個推播機制 (push mechanism)，使得語音服務不存在的時候，閘道器可以關掉對外的介面，只有當語音服務的需求發生，才透過播播機制 “wake up” 閘道器上的對外介面，保證系統不會遺失任何語音通話，同時達到省電與節費的要求。文中並且包含我們的系統實作成果。

### Abstract

In this paper, we propose a new mobile VoIP (Voice over IP) service based on ad hoc networking technologies, which integrates MANET (Mobile Ad-Hoc Network) and cellular networks. Using MANET as the base of the architecture, the system can be conveniently deployed and has an elastic structure. It also provides users mobile, flexible, and ubiquitous voice services. In this system, its gateway uses cellular interfaces to access cellular networks. Since connecting cellular interfaces normally

incurs charges and consumes power, we also propose a push mechanism to allow the gateway to stay off-line from the Internet when there are no calling activities, and “wake up” it as necessary. The proposed push mechanism can guarantee that our system will not loss any voice calls. It also achieves power-saving and cost-saving. Our prototyping experience is also reported.

**關鍵詞：**行動隨意網路，網際網路電話，會談啓始協議，推播機制，無線通訊

**Keywords:** MANET (Mobile Ad-Hoc Network), VoIP (Voice over IP), SIP (Session Initiation Protocol), push mechanism, wireless communication

### 一、Introduction

在以往 MANET 的發展中，大部份的研究多針對 MANET 本身為主，至於如何利用 MANET 所提供的彈性架構擴展與網際網路有關的應用，這方面的研究則著墨甚少；另一方面，行動通訊日益普及的今日，語音服務被視為殺手級的應用，現在的 VoIP 已經可以提供相當於傳統電話品質的語音服務，而且大大降低通話費用，SIP 則是現在最常被用來做為 VoIP 信息協定的協定，透過 SIP，除了傳統電話會議 (conference call)、指定轉接 (call forwarding)、語音信箱、以及影像通話 (media call) 服務均可以被實作外，還可以進一步提供整合網頁、電子郵件、即時訊息 (instant message)、和現狀資訊服務 (presence service) 的應用。

這篇論文當中，我們結合 MANET 與 VoIP，提出一個架構在 MANET 之上的行動 VoIP 服務平台——行動隨意網路為基礎架構的 VoIP 服務平台 (Ad-Hoc Network-Based Mobile VoIP Service Platform)：每一個行動主機 (Mobile Node) 均配備有一張 IEEE 802.11 無線網路卡，並且使用 ad hoc 模式組成 MANET，我們在 MANET 當中架設一台具有閘道器 (gateway) 功能的主機，這台主機對內使用 IEEE 802.11 無線網路卡，利用 ad hoc 方式與其它主機連結，對外則配備有行動電話介面 (cellular interface)，連結網際網路，這個介面可以是 GSM/GPRS/PHS/3G 手機。這個以 MANET 為基礎的架構，具有簡單架設和結構彈性的優點，每一個具備 IEEE 802.11 無線網路介面的使用者都可以加入這個網路，使用 VoIP 服務，這樣的架構還同時具備高度的移動性，可以運用在公車、火車等大眾交通工具以及行動辦公室等場景上，如果在家中，還可以構成一個無線的家庭 VoIP 系統。系統中的使用者仍然使用他們原本的 SIP URI (Uniform Resource Identifier) 與外界建立通話，毋需改變原來的使用習慣或增加新的號碼。

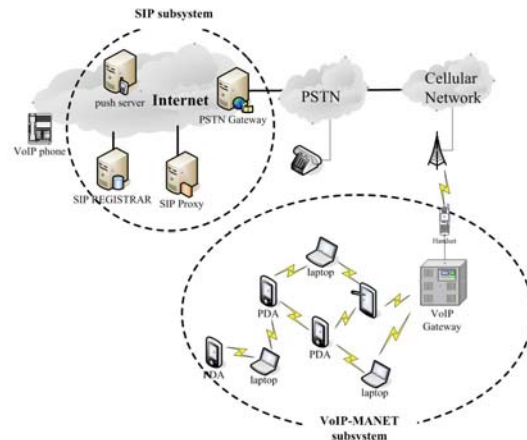
架構中的閘道器對外使用行動電話介面，如 GSM/GPRS/PHS/3G 手機連結網際網路，當系統一旦啓始，此介面就一直保持對外的通訊，使用數據服務，所以對系統而言，無論在電力消耗和計費上都是持續增加，開得越久，網路存取費用收得越多，電力也消耗越快。因為系統存在上述的缺點，我們考慮到 MANET 與網際網路之間並非隨時都有語音服務的需求，所以提出一個 push 機制 (push mechanism)，搭配這個機制，當語音服務的需求並不存在的時候，閘道器可以關閉連結網際網路的介面，只有當語音服務的需求發生，才透過 push 機制觸發閘道器撥接對外的行動電話介面，建立語音通話，這個方法可以保證系統不會遺失任何語音通話，同時達到省電與節省的需求。

在接下來的章節裡，第二節會說明本系統的架構，第三節則介紹 push 機制的流程，第四節說明我們的實作，包括 VoIP 閘道器的通訊協定層疊 (protocol stack) 和系統實作成果，第五節則是結論。

## 二、System Architecture

圖一 是我們的系統架構，包括一個 VoIP-MANET 子系統 (VoIP-MANET subsystem) 和一個 SIP 子系統 (SIP subsystem)，前者是一個可以透過行動介面存取網際網路的

MANET 系統，後者則包含執行行動 VoIP 服務所需的伺服器以及執行 push 機制所需的 push 伺服器 (push server)。



圖一、系統架構

在 VoIP-MANET 子系統中間的是一個 VoIP 閘道器 (VoIP Gateway)，閘道器配備有一個對外連接網際網路的介面，它可以是 GSM/GPRS/PHS/3G 手機介面，對內則使用一個 IEEE 802.11 無線網路卡連結內部的 MANET，MANET 由一群行動主機所組成，每一台行動主機上都配備有一張 802.11 無線網路卡，並且設定成 ad hoc 模式，因為語音服務對於服務的即時性有嚴格的需求，所以我們的 MANET 使用前置式路徑選擇演算法 (proactive routing algorithms)，每一台行動主機上面都會保持一份即時而準確的路由資訊。

SIP 子系統包括四個主要的伺服器：SIP registrar、SIP proxy、PSTN 閘道器、和 push 伺服器。SIP registrar 是一個資料庫，紀錄使用者的相關資料 (包括位置資訊) 以及現在的狀態，每一個 SIP 服務的使用者都必須定期傳送 SIP REGISTER 要求訊息到這裡更新它最新的位置，SIP proxy 則負責路由 SIP 信令，SIP registrar 與 proxy 可以視為邏輯元件，通常把兩個元件實作在同一台主機裡，我們稱它為 SIP 伺服器 (SIP server)，PSTN 閘道器連結網際網路和 PSTN 網路，可以把 VoIP 通話轉接到 PSTN 網路內，當 VoIP 閘道器暫時切斷與網際網路的連結時，push 伺服器負責在語音需求發生的情況下能夠 “waked up” VoIP 閘道器。

下一個章節將仔細介紹 push 機制，我們將可以看到，當 VoIP 閘道器切斷與網際網路的連線時，push 機制如何使得 VoIP-MANET 子系統不會遺失任何語音通話。

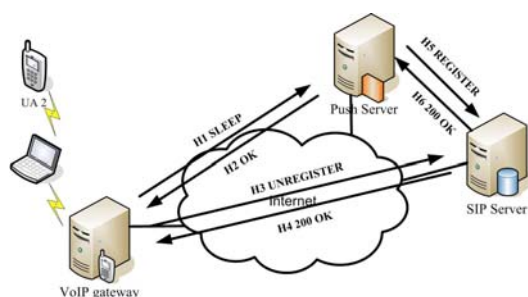
### 三、Push Mechanism

當語音服務的需求不存在的時候，VoIP-MANET 子系統與網際網路之間沒有服務的交流，VoIP 閘道器應該要可以切掉與網際網路之間的連結，這樣可以達到節費與省電的目的，但是在 VoIP 閘道器切斷所有對外連線的期間，任何來自網際網路的通話需求將無法到達 VoIP-MANET 子系統內。為了解決上述的問題，我們提出一個 push 機制，該機制利用短訊服務 (Short Message Service)，使得 VoIP 閘道器可以保持斷線但是卻不至於遺失欲到達 VoIP-MANET 子系統的通話。

下面我們將分兩個子章節分別講解 VoIP 閘道器關閉對外連線的流程以及關閉之後，push 機制如何使得下一次的語音通話需求到達 VoIP-MANET 子系統內。

#### (一) Sleep Procedure

我們的 push 機制沒有改變任何標準的 SIP 元件，但是引進一個新的節點--push 伺服器，當 VoIP 閘道器切斷與網際網路連結的期間，push 伺服器會作為 VoIP-MANET 子系統的代理者，接收所有欲傳往 VoIP-MANET 子系統的語音服務需求，而這個角色轉換的流程稱為 sleep procedure，當流程結束，VoIP 閘道器便會切斷與網際網路的連結。



圖二、sleep procedure

當 VoIP-MANET 子系統與網際網路之間沒有服務需求存在的時候，VoIP 閘道器會自動執行 sleep procedure，並在該流程結束之後，切斷與網際網路的連結，圖二顯示 sleep procedure 的流程：首先，VoIP 閘道器會傳送一個 sleep 要求訊息給 push server (H1)，這個訊息包含 VoIP-MANET 子系統內所有 SIP 用戶端 (SIP clients) 的 SIP URI，以及一個 MSISDN (Mobile Station International ISDN Number)，MSISDN 即手機電話號碼，在 push 伺服器以及 VoIP 閘道器上都配備 GSM/GPRS 手機介面，push 伺服器 and VoIP 閘道器與該介面之間的通訊協定則利用 AT 指令

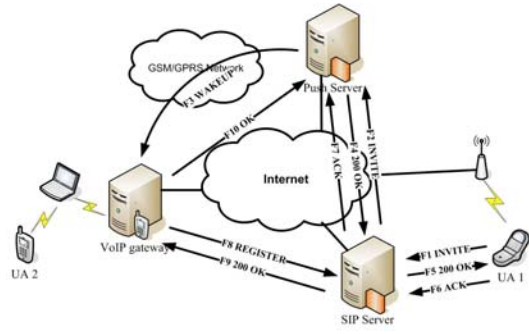
集 (AT Command Set)，透過 AT 指令集，push 伺服器以及 VoIP 閘道器可以利用 GSM/GPRS 手機介面傳送簡訊至通訊網路上，或者接收手機介面上收到的簡訊，所以，push 伺服器取得 VoIP 閘道器的 MSISDN 後，就可以透過簡訊的方式與 VoIP 閘道器交換訊息，而 VoIP 閘道器用來連結網際網路的手機介面可以和這個收送簡訊的介面是同一個，如此一來還可以提升介面的使用率；push 伺服器內有二個表：閘道器表 (gateway table) 和 SIP 用戶表 (SIP client table)，閘道器表包含 VoIP 閘道器的代號、狀態、MSISDN、以及 IP 位址，SIP 用戶表則包含 SIP URI、閘道器代號、以及 SIP 註冊的終止時間 (expire time)，一旦 push 伺服器收到來自 VoIP 伺服器的 sleep 要求訊息，它會把內容紀錄在閘道器表和 SIP 用戶表內，將 VoIP 閘道器的狀態標註為離線，並且回覆一個 ok 訊息給 VoIP 閘道器 (H2)；VoIP 伺服器收到 ok 訊息後，便替 VoIP-MANET 子系統內的所有 SIP 用戶端發送 UNREGISTER 訊息給 SIP 伺服器 (H3)，這是一個 SIP REGISTER 要求訊息，Expires 欄位為 0，因為 VoIP-MANET 子系統內可能有一個以上的 SIP 用戶端，所以可能會發送多個 UNREGISTER 訊息，SIP 伺服器收到之後會回覆 SIP 200 OK 訊息給 VoIP 閘道器 (H4)；另一方面，當 push 伺服器回覆 VoIP 閘道器 ok 訊息之後 (H2)，push 伺服器會開始發送 SIP REGISTER 要求訊息給 SIP 伺服器 (H5)，替 VoIP-MANET 子系統內的 SIP 用戶端作註冊，這個 SIP REGISTER 要求訊息內的 Contact 欄位會填 push 伺服器，表示將來和 VoIP-MANET 子系統有關的 SIP 信令均送到 push 伺服器來，SIP 伺服器收到 SIP REGISTER 要求訊息後，更新資料庫內的資料，並且回覆 SIP 200 OK 訊息給 push 伺服器 (H6)。一旦 sleep procedure 完成，VoIP 伺服器便切斷與網際網路的連結，而 push 伺服器則成為 VoIP-MANET 子系統的代理者，push 伺服器除了定期替 SIP 用戶端作 SIP 註冊外，以後所有 VoIP-MANET 子系統有關的 SIP 信令都會經由 SIP 伺服器送到 push 伺服器。

#### (二) Wake Up Procedure

Sleep procedure 結束後，push 伺服器成為網路上 VoIP-MANET 子系統的代理者，一旦網際網路上發生與 VoIP-MANET 子系統有關的通話要求訊息，都會被 push 伺服器收到，這個時候 push 伺服器便會啓始 wake up procedure，通知 VoIP 閘道器重新打開與網際網路的連結，並且轉接此通話至 VoIP-MANET 子系統內的 SIP 用戶端，wake up procedure 可以分為兩個部份，wake up 程序以及通話轉接程



序，下面分別描述這兩個程序。

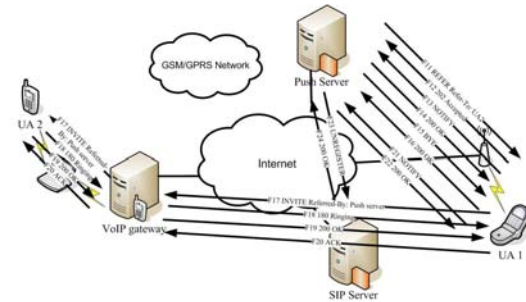


圖三、wake up process

圖三顯示 wake up 程序的流程，假設 sleep procedure 已經完成，VoIP-MANET 子系統已切斷與網際網路的連結，這個時候，網際網路上 SIP 用戶端 UA1 (User Agent 1) 欲與 VoIP-MANET 子系統內的 SIP 用戶端 UA2 建立通話。

因為 push 伺服器已經是 VoIP-MANET 子系統在網際網路上的代理人，所以 SIP 伺服器會把 UA1 的 SIP INVITE 要求訊息轉送給 push 伺服器 (F1,F2)；一旦 push 伺服器收到來自 UA1 的 SIP INVITE 要求訊息，會根據請求的 SIP URI 查詢 SIP 用戶表和開道器表，得到 VoIP 開道器的狀態以及 VoIP 開道器註冊的 MSISDN，因為 VoIP 開道器已經離線，push 伺服器必須通知 VoIP 開道器要重新建立與網際網路的連結，令 VoIP-MANET 子系統可以接收 UA1 的通話要求，所以，push 伺服器會透過 GSM/GPRS 手機介面傳送包含 wakeup 訊息的簡訊給 VoIP 開道器，觸發 VoIP 開道器重新建立與網際網路的連結 (F3)；VoIP 開道器重新啟動對外連結的期間，push 伺服器會先與 UA1 建立通話 (F4~F7)，如此一來可以避免因 VoIP 開道器重新開啓對外連線時間超過使用者 (UA1) 所允許，或通話信令過期 (timeout) 緣故使得 UA1 掛斷電話，而 push 伺服器可以先準備一段預先錄好的錄音，讓 UA1 稍待片刻，也就是說，當 UA1 與 push 伺服器建立連線之後，撥放如：“通話建立中，請稍後”、“通話轉接中，請稍後”...等錄音，讓 UA1 等待系統替他與 UA2 的建立通話；另一方面，一旦 VoIP 開道器收到 push 伺服器所送的 wakeup 簡訊，VoIP 開道器重新撥接手機介面連結網際網路，並且替 VoIP-MANET 子系統內的 SIP 用戶重新註冊 (F8, F9)，這時候 SIP REGISTER 要求訊息內的 Contact 欄位會更新為 VoIP 開道器對外手機介面新得到的 IP 位址，這個位址可能已經不同於之前使用的 IP 位址，因為該手機介面切斷網際網路連結的期間，原本的 IP 位址可能已經分配給其它人使

用；一旦手機介面重新連結網際網路，VoIP 開道器會透過網路連線回覆 ok 訊息給 push 伺服器 (F10)，並且更新在 push 伺服器上的資料與狀態。到這裡為止，wake up 程序就算完成，VoIP-MANET 子系統也重新回復與網際網路的連結，接下來將說明本系統如何將通話轉換給 UA2。



圖四、通話轉接程序

圖四顯示通話轉接程序，該程序利用 SIP REFER method (RFC3515) 將 UA1 與 push 伺服器之間的通話連線轉接給 UA2，使得網際網路上的 SIP 用戶端 UA1 能與 VoIP-MANET 子系統內的 UA2 開始進行通話。首先，push 伺服器送給 UA1 一個 SIP REFER 要求訊息，該訊息內的 Refer-To 欄位填 UA2 的 SIP URI (F11)，通知 UA1 把通話轉到 UA2，建立與 UA2 之間的通話連線；UA1 會回覆 SIP 200 Accepted 訊息給 push 伺服器 (F12) 表示接受；接著，UA1 先送給 push 伺服器一個 SIP NOTIFY 訊息 (F13) 報告通話建立的進度，該訊息 body 裡頭包含 “SIP/2.0 100 Trying” 訊息，而 body type 為 “message/sipfrag”，表示 UA1 正在嘗試與 UA2 建立通話，準備 INVITE UA2，而 push 伺服器則回覆 SIP 200 OK 給 UA1 (F14)；F15 顯示 push 伺服器會主動送 SIP BYE 要求訊息給 UA1，結束 push 伺服器與 UA1 之間的 RTP (Real-Time Transport Protocol) 通話，儘管如此，push 伺服器和 UA1 之間的 dialog 還是會存在，直到整個 REFER 流程結束；接著，UA1 傳送 SIP INVITE 要求訊息給 UA2，請求與 UA2 建立通話連線，過程同一般通話建立的流程 (F17~F20)，UA1 與 UA2 會交換彼此之間的通話資訊，如 Contact IP，使用的語音壓縮格式，收送 RTP 的 IP 以及通訊埠 (port)，成功建立連線後，UA1 與 UA2 便開始他們的通話；UA1 最後還會以 SIP NOTIFY 訊息通知 push 伺服器已完成 REFER 流程，body 裡頭包含與 UA2 最後的 SIP 200 OK 訊息，Subscription-State 欄位可以是 “terminated;reason=noresource”，並且結束與 push 伺服器之間的 dialog (F21)；VoIP 開道器重新連結網際網路後，push 伺服器可以選擇向

SIP 伺服器 UNREGISTER 所有由它代為註冊的 SIP 用戶端 (F23, F24), UNREGISTER 是一個 SIP REGISTER 要求訊息, Expire 欄位為 0。

到這裡為止, 網際網路上的 UA1 與 VoIP-MANET 子系統內的 UA2 已經成功建立通話連線, 我們由上述流程可以清楚發現, 在我們所提出的架構裡, 沒有變更到任何原來的 SIP 元件, 包括 SIP registrar 和 SIP proxy, 而 SIP 用戶端, 如 UA1 以及 UA2, 也都使用標準的 SIP 信令, 只有在 VoIP 閘道器與 push 伺服器之間增加了使用簡訊來溝通的介面, 並且設計它們之間專門使用的訊息格式, VoIP 閘道器是我們所提供, push 伺服器也是我們新增的節點, 所以這兩者都是我們所能控制的元件, 除此之外, 這樣的架構還有兩個優點, 其一是 VoIP-MANET 子系統內的每一個 SIP 用戶端仍然是跟原來自己的 SIP registrar 作註冊, 完全沒有影響到 SIP 信令原本的邏輯, 所以 SIP 用戶端不用改變自己原來的使用習慣, 二是一個 push 伺服器可以同時服務多個 VoIP-MANET 子系統, 非常具有彈性。

#### 四、Implementation

這個章節分為兩個部份, 分別描述 VoIP 閘道器的 protocol stack 以及我們所實作的原型 (prototype)。

##### (一) Protocol Stack for VoIP Gateway

圖 五顯示 VoIP 閘道器的 protocol stack, VoIP 閘道器是本系統中最重要的節點, 它對外連結網際網路, 對內則連結系統內部的 MANET, 並且路由所有通過系統的 SIP 信令和語音封包。除了原本的 TCP/IP protocol stack, 它還包括四個新的元件, 分別是 (1).Ad-Hoc Routing Protocol、(2).SMS (Short Message Service) Agent、(3).Push Communicator、和(4).IP Address Manager, 以下依序說明這四個元件。

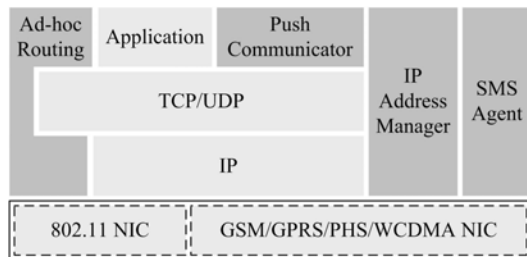


圖 五、Protocol Stack for VoIP Gateway

(1).Ad-Hoc Routing Protocol: MANET 在未加入某些專屬的路由協定之前, 封包傳送

可達的範圍只有 single-hop 而已, 所以我們實作 Ad-Hoc Routing Protocol 在包含 VoIP 閘道器在內的 MANET 上的每一台行動主機, 使其能支援 multi-hop。我們的作法是讓 MANET 中的每台主機都紀錄一個自己的路由表 (routing table), 並且不斷廣播自己的路由表, 相鄰的行動主機就可以從廣播出來的封包中獲得目前網路拓撲的資訊, 並且與自己所紀錄的路由表資料作比較, 以新增或更新自己的路由表, 然後再繼續將自己的路由表對相鄰節點廣播。

(2).SMS Agent: SMS Agent 可以利用 VoIP 閘道器上的 GSM/GPRS 手機介面傳送簡訊, 亦可以將手機介面上收到的簡訊取出, 它與手機介面之間的溝通利用 AT 指令集完成, SMS Agent 同時提供 API (Application Program Interface)讓其它的程式也可以利用簡訊服務。除了 VoIP 閘道器之外, 在 push 伺服器上面亦有 SMS Agent 元件, 兩者可以利用簡訊交換訊息。

(3).Push Communicator: Push Communicator 元件負責與 push 伺服器之間的資訊交換, 包括更新 push 伺服器上 VoIP 閘道器的資料與狀態, 以及接受由 push 伺服器端主動送來的訊息, VoIP-MANET 子系統與網際網路的連結存在的時候, Push Communicator 與 push 伺服器之間透過網路交換訊息, 一旦 VoIP 閘道器切斷與網際網路之間的連結, Push Communicator 則透過 SMS 與 push 伺服器作訊息交換。

(4).IP Address Manager: IP Address Manager 是 VoIP 閘道器的核心, 它有以下幾個主要功能: (a).解決 SIP 信令與 RTP 封包穿越 NAT (Network Address Translation)的問題; (b).判斷切斷 VoIP 閘道器與網際網路連結的時機, 啓始 sleep procedure; (c).重新啓始與網際網路之間的連結, 建立 VoIP-MANET 子系統與 SIP 伺服器的關係。傳送 SIP 信令的過程中, SIP 客戶端會把自己的 Contact、傳接語音封包的 IP 位址以及通訊埠的資訊填寫在信令內, 因為 VoIP-MANET 子系統是一個私密網路 (private network), 所以這時候 NAT 穿越 (NAT Traversal)的問題就發生了: 網際網路上的 SIP 用戶端要怎麼使用這些私密資訊與 VoIP-MANET 子系統內的 SIP 用戶端交換 SIP 信令和語音封包呢? 網際網路上的 SIP 用戶端可能會遭遇到僅有單向通話的問題 (聽得到來自 VoIP-MANET 子系統內 SIP 用戶端的聲音, 但是自己的聲音無法送給對方), 觀察到這樣的問題, 我們的 IP Address Manager 具有 ALG (Application Layer Gateway)的功能, 一方

面它轉換應用層 SIP 信令的內容以穿越 NAT, 另一方面也保證語音封包的交換沒有問題。一旦 IP Address Manager 判斷對外連結網際網路的資源閒置, 接下來也會有好一陣子沒有通話需求, IP Address Manager 便會通知 Push Communicator 啓始 sleep procedure, 當 sleep procedure 完成以後, 就關閉與網際網路的連結, 直到 Push Communicator 收到來自 push 伺服器包含 wakeup 訊息的簡訊, 由 Push Communicator 通知 IP Address Manager 再次開啓與網際網路的連結為止, 才重新連上網際網路, 這個時候, IP Address Manager 會主動替 VoIP-MANET 子網路內的行動主機恢復在 SIP 伺服器的註冊。

## (二) Implementation

基於上述的 push 機制以及架構, 我們實作了一套系統原型 (prototype), 以下分別報告幾個主要的元件及執行畫面。

圖六顯示的是 push 伺服器執行的情形, 我們在 Windows XP 系統上實作 push 伺服器, 主要使用 C 語言來撰寫, 目前 push 伺服器可以支援簡單的 SIP 信令以及 REFER method 以和 SIP 元件溝通, 並且提供一個簡單的圖形化介面, 可以顯示 SIP 信令的內容和交換的過程, 不但可以讓使用者充份瞭解系統的流程, 還可以作為 debug 之用。目前除了自動模式 (automatic mode) 之外, 還可以選擇 step-by-step 模式讓 push 伺服器一步一步進行下一個動作, 方便展示之用。

圖六、push server 程式

我們在 Fedora Linux 上實作 VoIP 閘道器, 利用 iptables 以及 libipq 函式庫開發 VoIP 閘道器, 目前的原型裡, VoIP 閘道器使用 PCMCIA 介面連接 PHS J88 手機作為對外的連

上網際網路的介面, 使用 usb 介面的 IEEE 802.11 無線網路卡連接內部的 MANET, 如圖七。



圖七、VoIP 閘道器及其使用的介面卡

SIP 用戶端方面, UA1 我們使用友訊科技公司的 IP Phone 產品作為 SIP 用戶端來測試, 該產品有支援 SIP REFER method (圖八)。



圖八、IP phone

系統實作與測試過程中, 我們使用 IPTel SIP Server (Freeware) 作為測試之用, PSTN 閘道器則是 Cisco2600 PSTN 閘道器。

## 五、Conclusions

這篇論文當中我們提出一個以行動隨意網路為基礎架構的 VoIP 服務平台, 以及它的 push 機制, 前者維持原本行動隨意網路簡單架設、結構彈性的優點外, 還提供使用者具移動性、彈性、以及服務無間隙功能的語音服務, 後者使得 VoIP-MANET 子系統與網際網路間沒有通訊需求的時候, VoIP 閘道器可以切斷對外網路的連結, 只有當需要的時候才把對外介面撥接上網, 不但替系統達到省電、節費的目的, 亦不會遺失任何通話。我們的系統沒有變更任何標準的元件, 不管 VoIP-MANET 子系統內部或外面的 SIP 用戶端也都不用改變他

們原本的使用習慣，一台 push 伺服器可以同時服務多個 VoIP-MANET 子系統，提供 push 服務。

未來，我們會進一步在行動隨意網路為基礎架構的 VoIP 服務平台上設計服務，並且在 push 伺服器上開發更多種的 push 服務。

### 參考文獻

- [1] D. Collins, Carrier Grade Voice over IP. 2nd ed., McGraw-Hill, 2003.
- [2] K. Egevang and P. Francis, The IP Network Address Translator (NAT). IETF RFC1631, May 1994.
- [3] Y. B. Lin, Y. C. Lo, and C. H. Rao, A Push Mechanism for GPRS Supporting Private IP Addresses. IEEE Comm. Letters, vol. 7, no. 1, pp. 24-26, Jan. 2003
- [4] C. E. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. Proceedings of the ACM SIGCOMM, pp. 234-244, 1994.
- [5] C. H. Rao, D. F. Chang, and Y. B. Lin, iSMS: An Integration Platform for Short Message Service and IP Networks. IEEE Networks, vol. 15, pp. 48--55, Mar/Apr 2001.
- [6] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, SIP: Session Initiation Protocol. IETF RFC3261, June 2002.
- [7] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP) . IETF RFC3725, Apr. 2004.
- [8] R. Sparks, The Session Initiation Protocol (SIP) Refer Method. IETF RFC515, Apr. 2003.
- [9] P. Srisuresh and M. Holdrege, IP Network Address Translator (NAT) Terminology and Considerations. IETF RFC2663, Aug. 1999.