

Deployment of SIP-based Voice Services over IPv6 Networks

陳如慧
靜宜大學資管所
rhoda@pu.edu.tw

黃楷珉
靜宜大學資管所
g9471013@pu.edu.tw

翁永昌
靜宜大學資工系
ycwong@pu.edu.tw

摘要

SIP (Session Initiation Protocol) 是傳送網路電話控制訊息的標準協定之一，可建立、修改、或結束網路通話。本文介紹作者在 IPv6 環境下提供 SIP 語音服務 (稱為 SIPv6 服務) 的經驗、建置過程所遭遇的問題、以及解決問題的方法。此外，針對沒有 NAT-PT (Network Address Translation - Protocol Translation) 的環境，本文提出三種讓新的 SIPv6 服務與既有的 SIPv4 服務互通的方式 (scenario)，並比較它們的優劣。本文最後提供系統壓力測試的數據，可作為相關建置的參考。

關鍵詞：網路電話、會議起始協定、IPv6 網路、IPv4/IPv6 轉換。

Abstract

SIP, the session initiation protocol, is a new signaling protocol developed to set up, modify, and tear down voice session over the Internet. This paper presents the first results of our study and deployment with the new IPv6-based SIP service. The paper also describes the possible scenarios in which the user agents residing at IPv6 network can communicate with that residing at IPv4 network.

Keywords: voice over IP (VoIP), session initiation protocol (SIP), IPv6 network, IPv4/IPv6 transition.

1. 簡介

SIP (Session Initiation Protocol) 是由 IETF 工作小組於 1999 年所制定，用來傳送網路電話的控制訊息，可建立、修改、或結束多媒體的通話 (session) [1]。SIP 採用主從式架構，將控制訊息與語音封包分開處理。SIP 控制訊息夾帶了建立通話所需的參數，如發話端的 IP 位址和 UDP port 等，這些參數可以用 Session Description Protocol (SDP) 來描述 [2]。有了這些參數，發話端和受話端稍後才能夠互相傳送語音封包。

SIP 的主要元件有 User Agent (UA)、Proxy server、以及 Registrar。UA 是網路電話機或撥話軟體，可細分為 User Agent Client (UAC) 發話端與 User Agent Server (UAS) 受話端兩種。Proxy server 負責將 UA 發出的 SIP 訊息傳送到下一個 SIP 元件。Registrar 負責接收 UA 的發出註冊請求，並更

新 UA 的位置資訊。SIP URI (Uniform Resource Indicators) 通常被用來識別使用者。URI 的格式為 sip:user@host，其中 user 為使用者姓名或電話號碼；host 為網域 (domain) 名稱或 IP 位址。例如 sip:alice@176.7.6.1。

IPv6 網路 [3] 是次世代 IP 網路的標準，將逐步取代目前的 IPv4 網路。IPv6 網路除了提供較大的定址空間之外，也提供了較佳的品質保證、安全性、IP 位址自動配置 (auto-configuration) 機制、以及較有效率的封包路由方式。在 IPv6 環境下提供 SIP 語音服務 (稱為 SIPv6 服務) 是目前熱門的研究主題之一 [4-5]。本文將介紹我們建置 SIPv6 服務的經驗、系統架設過程所遭遇的問題、以及解決問題的方法。

從 IPv4 網路過渡到 IPv6 網路需要一段漫長的時程。在兩種網路並存的這段期間，需要有適當的移轉機制 (transition mechanism)，讓 IPv6 節點 (node) 能夠和 IPv4 節點溝通。NAT-PT (Network Address Translation - Protocol Translation) [6] 便是一種符合上述需求的移轉機制。NAT-PT 負責將封包裡面 IPv6 的表頭 (header) 轉換成 IPv4 的格式，反之亦然。由於 NAT-PT 不對 IP payload 的部分作任何處理，所以當 UAv4 (支援 IPv4 的 UA) 要與 UAv6 (支援 IPv6 的 UA) 溝通時，封裝在 SIP/SDP 訊息裡面的 IP 位址便無法被轉換。

為了解決這個問題，文獻 [7-8] 提出 NAT-PT 搭配 SIP-ALG (Application-Level Gateway) 的機制。SIP-ALG 認得 SIP/SDP 訊息的格式，同時負責轉換其中所攜帶的 IP 位址。NAT-PT 雖然有許多優點，但也帶來了一些限制：(1) 無法提供 UAC 到 UAS 間 end-to-end 的安全性；(2) NAT-PT 本身可能是 single point of failure 或是網路的瓶頸。本文針對沒有 NAT-PT 的環境，提出讓新的 SIPv6 服務能夠和既有的 SIPv4 服務互通的方法 (scenario)，並分析其優劣。

本文章節安排如下：第二節描述 SIP 的基本通話流程；第三節介紹我們提供 SIPv6 語音服務的經驗；第四節討論讓 SIPv6 服務與 SIPv4 服務互通的可能方式；第五節提供系統效能評估；最後是簡短的結論。

2. SIP 通話流程

本節以 Alice 撥打電話給 Bob 為例，介紹 SIP

通話建立的基本流程(圖 1)。

首先 Bob 透過話機 UAS 以 bob@pusip.edu.tw 的身份向 Proxy Server 提出註冊 (step 1)。Proxy Server 收到註冊訊息會回應 200 OK，同時把 UAS 所在的 IP 位址紀錄在 registrar (step 2)。稍後 Alice 使用話機 UAC 撥打電話給 Bob。UAC 送出 INVITE bob@pusip.edu.tw 訊息，希望與 Bob 通話(step 3)。Proxy server 收到 INVITE 訊息後，首先從 registrar 查到 UAS 所在的 IP 位址，接著把 INVITE 訊息轉送給 UAS (step 4)。UAS 收到 INVITE 訊息後發出響鈴。Bob 接聽電話後話機 UAS 回應 200 OK 訊息 (step 5)。Proxy Server 將收到的 200 OK 訊息轉送給 UAC (step 6)。最後 UAC 回應 ACK 訊息並建立 RTP 語音通道 (steps 7-8)。

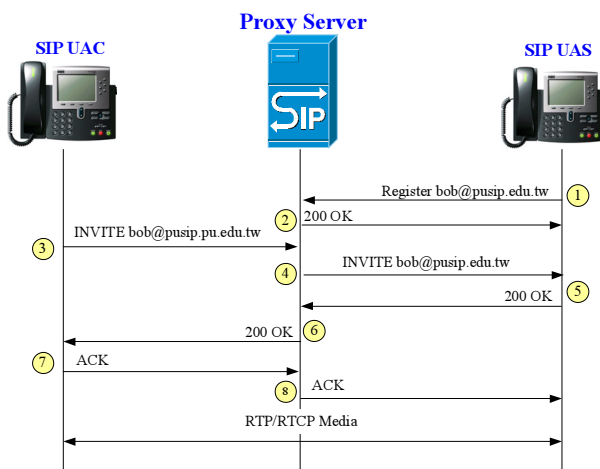


圖 1. SIP 通話建立

3. SIPv6 語音服務

本節將介紹如何在 IPv6 網路上提供 SIP 語音服務。我們以 SIP Express Router (SER) 0.9.6 版 [9] 為例，說明 SIP proxy 伺服器的安裝設定；UA 的部分則選用 kphone 3.11 版[10]。

3.1 SER 伺服器架設

架設 SER 伺服器的步驟說明如下：

步驟 1. 從官網下載 ser-0.9.6_src.tar.gz 原始碼。

步驟 2. 修改/etc/profile 檔案，新增下列三行文字，以便定義 SIP_DOMAIN 環境變數以及指定 MySQL 的路徑：

```
SIP_DOMAIN="your SER domain"  
LD_LIBRARY_PATH=/usr/local/lib/mysql  
export SIP_DOMAIN LD_LIBRARY_PATH
```

步驟 3. 將檔案 ser-0.9.6_src.tar.gz 解壓縮後進行編譯與安裝。在這裡，我們會將所有的模組都一併安裝，以利往後的功能擴充。相關指令如下：

```
#tar zxvf ser-0.9.6_src.tar.gz  
#cd ser-0.9.6
```

```
#make all include_modules="mysql jabber cpl-c  
auth_radius group_radius uri_radius avp_radius  
postgres pa"  
# make install include_modules="mysql jabber  
cpl-c auth_radius group_radius uri_radius  
avp_radius postgres pa"
```

步驟 4. SER 搭配 MySQL 可以進行身分驗證的功能，故我們在此要啟動 MySQL，並且建立 SER 資料庫與資料表。相關指令如下：

```
#/etc/init.d/mysqld start  
# /usr/local/sbin/ser_mysql.sh create
```

步驟 5. 修改設定檔/usr/local/etc/ser/ser.cfg，反註解以下這幾行，以便 MySQL 進行身分驗證。

```
loadmodule "/usr/lib/ser/modules/mysql.so"  
loadmodule "/usr/lib/ser/modules/auth.so"  
loadmodule "/usr/lib/ser/modules/auth_db.so"  
modparam("usrloc", "db_mode", 2)  
modparam("auth", "calculate_ha1", yes)  
modparam("auth_db", "password_column",  
"password")  
if (!www_authorize("your SER Domain",  
"subscriber")) {  
    www_challenge("your SER Domain", "0");  
    break;  
};
```

步驟 6. 為能順利執行 SER 的 IPv6 模組，我們必需在 ser.cfg 加入 SER 伺服器的 IPv4 及 IPv6 位址，並且指定 port 為 5060。

```
listen= 140.128.1.233  
listen= 2001:E10:6440:1:214:85FF:FE4E:B1E2  
port=5060
```

步驟 7. 啟動 SER 伺服器。

```
#/usr/local/sbin/serctl start
```

步驟 8. 利用 ser -c 確認 SER 能正常運作：

```
[root@sip233 src]# ser -c  
Listening on  
udp: 140.128.1.233 [140.128.1.233]:5060  
udp: 2001:E10:6440:1:214:85FF:FE4E:B1E2  
[2001:E10:6440:1:214:85FF:FE4E:B1E2]:5060  
tcp: 140.128.1.233 [140.128.1.233]:5060  
tcp: 2001:E10:6440:1:214:85FF:FE4E:B1E2  
[2001:E10:6440:1:214:85FF:FE4E:B1E2]:5060  
Aliases:  
tcp: sip233.pu.edu.tw:5060  
udp: sip233.pu.edu.tw:5060
```

根據我們的經驗，如果出現下列錯誤訊息，請於檔案/etc/hosts 中加入 SER 伺服器 IPv4 位址以及 IPv6 位址對應的 domain name。

```
0(2921) WARNING: fix_socket_list: could not rev.  
resolve 140.128.1.233  
0(2921) WARNING: fix_socket_list: could not rev.  
resolve 2001:E10:6440:1:214:85FF:FE4E:B1E2
```

3.2 Kphone UA 安裝設定

安裝 kphone UA 的過程茲說明如下：

步驟 1. 從官網下載 kphone 3.11 版套件。

步驟 2. 我們發現在 kphone 的安裝過程中需要用到 kdb2html 程式檔。該檔案並沒有包含在 kphone 套件中，請自行下載 kdelibs-3.1.4-4.i386.rpm [11]，解開後取出 kdb2html 並複製到 /usr/bin 目錄底下。

步驟 3. 根據我們的經驗，kphone 安裝時需手動告知 QT 的位置。建議先使用 updatedb 指令將搜尋資料庫更新，接著再使用 locate qt 來找出 qt 所在目錄。相關指令如下：

```
# tar -zxvf kphone-3.11-ipv6.tar
# rm -f config.cache config.status config.h
# updatedb
# locate qt
# ./configure --with-qt-dir=/usr/lib/qt-3.3
--with-qt-includes=/usr/lib/qt-3.3
--with-qt-libraries=/usr/lib/qt-3.3/lib
--enable-mt
# make
# make install
```

步驟 4. 到此 kphone 的安裝已經完成。如果您的桌面環境程式是以 KDE 開啟，那麼在開啟 kphone 時有可能會發生錯誤。請找出桌面環境轉換程式，並將桌面環境程式更改為 GNOME (圖 2)。



圖 2. 更改桌面環境

步驟 5. 執行 kphone，請點選 x-win 視窗的應用程式->執行程式，然後輸入 kphone。

步驟 6. kphone 一開始執行時會詢問是否要啟動 IPv4 及 IPv6 介面。請回答 Yes。

步驟 7. 接著會進入 kphone 的啟始畫面(圖 3)。畫面右邊的 Identify Editor 視窗可輸入註冊的相關訊息。舉例來說，如果使用者的 SIP URI 是 rhoda@sip233.pu.edu.tw，那麼 Full Name 欄位、User Part of SIP URL 欄位、Host Part of SIP URL 欄位、以及 Authentication Username 欄位請分別輸入 rhoda@sip233.pu.edu.tw、rhoda、

sip233.pu.edu.tw、以及 rhoda。

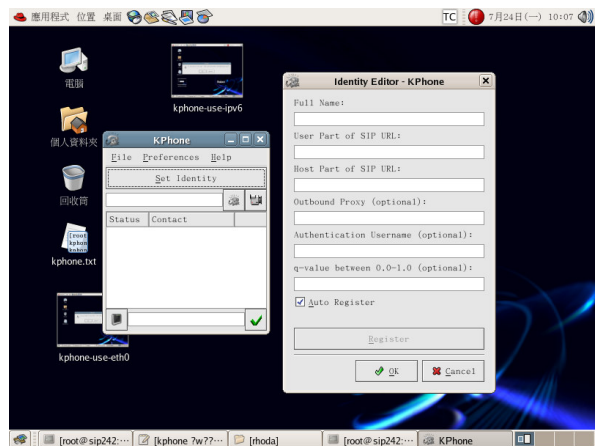


圖 3. kphone 啟始畫面

在步驟 6 如果同時啟動 IPv4 介面以及 IPv6 介面，kphone 內定由 IPv6 介面來發送註冊訊息 REGISTER sip: sip233.pu.edu.tw。DNS 伺服器必須有相關的設定，以便解析 sip233.pu.edu.tw 對應的 IP 位址。

4. Scenarios for SIPv4-SIPv6 Cooperation

在本文中，所謂 dual-mode UA 是指一個 UA 同時能支援 IPv4 與 IPv6；反之則稱為 single-mode UA。本節將討論在沒有 NAT-PT 的環境下，一個在 IPv6 網路上的節點如何與另一個在 IPv4 網路上的節點通話。我們提出三種可能的方案：single-mode (SM) UA、dual-mode single-registration (DMSR) UA、以及 dual-mode dual-registration (DMDR) UA。

4.1 Single-mode UA

如圖 4 所示，UA1 與 SIP proxy server 1 只支援 IPv4，SIP proxy server 2 則同時支援 IPv4 與 IPv6。UA2 執行一個 UA_{v6} 軟體。為了與 IPv4 網路上的 UA1 通話，UA2 又另外安裝了一套 UA_{v4} 軟體(比如 x-lite)。UA2 開機後從 DHCP 伺服器取得一個 IPv4 與一個 IPv6 位址。UA_{v4} 與 UA_{v6} 分別用剛才取得的位址向 proxy server 1 與 proxy server 2 註冊。

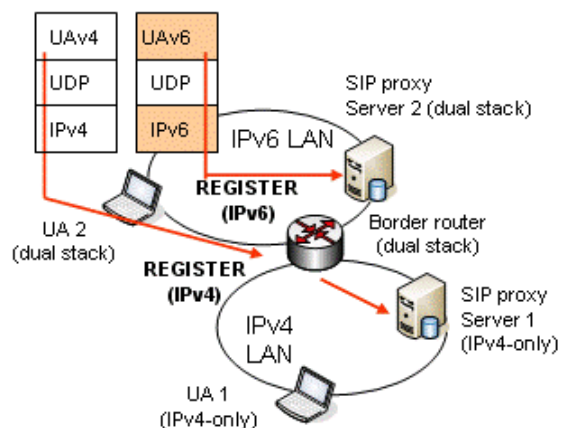


圖 4. SM UA 註冊流程

UA1 撥話給 UA2 的流程如圖 5 所示。首先 UA1 送出 INVITE 訊息給 proxy server 1 (step 1)。Proxy server 1 收到 INVITE 訊息後，從 registrar 查到 UA2 的 IPv4 位址，接著把 INVITE 訊息轉送給 UA2 上的 UA4 軟體 (step 2)。UA2 收到 INVITE 訊息後回應 200 OK (step 3)。Proxy Server 1 將收到的 200 OK 轉送給 UA1 (step 4)。最後 UA1 回應 ACK 訊息以便建立 RTP 語音通道 (step 5)。

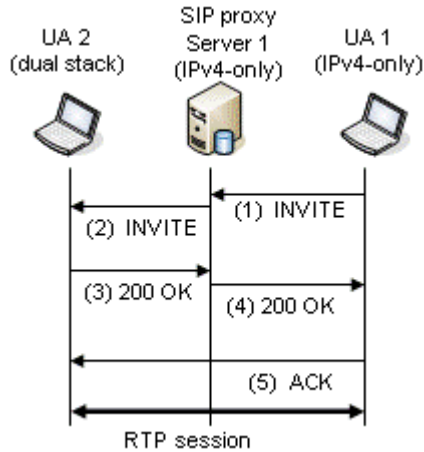


圖 5. SM UA 通話建立

4.2 Dual-mode single-registration UA

在 DMSR 模式底下，UA2 配置了 dual-mode UA，以便同時支援 IPv4 及 IPv6 (圖 6)。UA2 內定使用 UA v6 模組向 proxy server 2 註冊，註冊訊息中必須額外攜帶 UA2 的 IPv4 位址。如此一來，proxy server 2 將會知道 UA2 的 IPv4 以及 IPv6 位址。在這個模式下 UA v4 模組不會發送註冊訊息，所以被稱為 single-registration 模式。

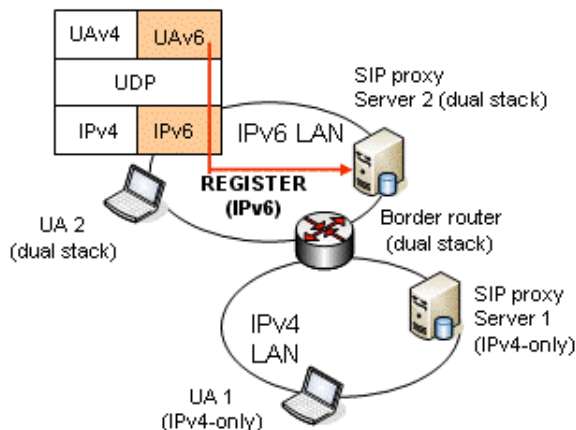


圖 6. DMSR UA 註冊流程

UA1 撥話給 UA2 的流程如圖 7 所示。首先 UA1 送出 INVITE 訊息給 proxy server 1 (step 1)。Proxy server 1 無法從 registrar 查到 UA2 所在的位址，所以把 INVITE 訊息轉送給 proxy server 2 (step 2)。此時 proxy server 2 扮演 SIP redirector 的角色，負責回應 302 move temporary 訊息給 UA1 (step 3)。UA1

收到後，從該訊息的 Contact 欄位中取得 UA2 的 IPv4 位址，接著回應 200 OK (step 4)。根據上一步驟所取得的位址，UA 1 重新發送一個 INVITE 訊息給 UA2 (step 5)。接下來為 SIP 建立通話的標準流程 (steps 6-8)。

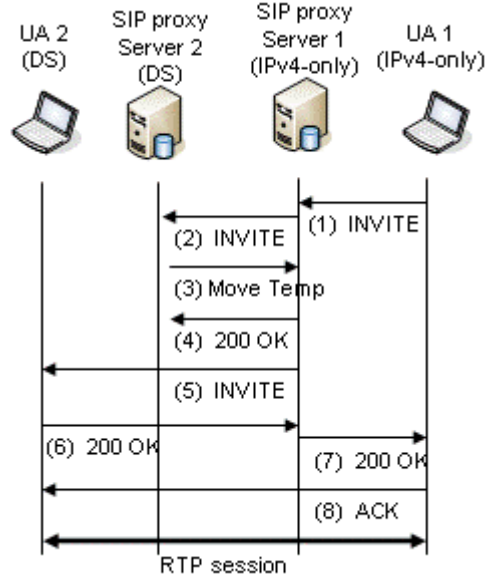


圖 7. DMSR UA 通話建立

4.3 Dual-mode dual-registration UA

在 DMDR 模式底下，UA2 也是配置 dual-mode UA。與上一個模式不同的是，UA2 的 UA v4 模組和 UA v6 模組會分別向 proxy server 1 以及 proxy server 2 註冊 (圖 8)。因為兩個模組都會發送註冊訊息，所以這個模式被稱為 dual-registration。在這個模式底下，UA1 撥話給 UA2 的流程如圖 5 所示，在此不再贅述。

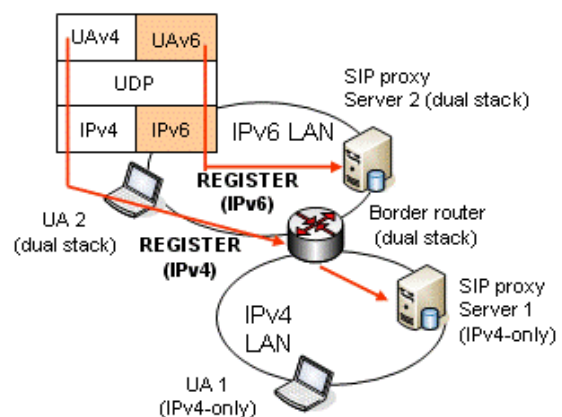


圖 8. DMDR UA 註冊流程

4.4 Discussion

SM 模式的優點是簡單，不需要修改 UA 和 proxy server；缺點是要安裝兩種 single-mode UA。DMSR 模式的優點是 dual-mode UA 只需發送一個註冊訊息；缺點是 proxy server 2 必須維護 dual-mode

UA 的 IPv4 位址和 IPv6 位址，以及這兩種位址之間的對照關係。相對於 DMSR 模式，DMDR 模式的優點是 proxy server 2 不用維護 dual-mode UA 的位址對照表；缺點是 dual-mode UA 必須支援多重註冊，然而目前的 UA 多沒有這種功能。

5. 系統效能評估

本節將介紹 SER 伺服器在 IPv6 環境下的效能。我們利用安裝在 Linux 平台上的 SIPp 軟體[12] 來扮演 UAC 和 UAS，透過不斷地向 SER 伺服器發出 INVITE 訊息，來觀察 SER 的 CPU 使用率和 INVITE 訊息流失的比率。測試平台的架構如圖 9 所示，PC 的規格為 AMD 3.0 CPU 搭配 1GB 主記憶體。

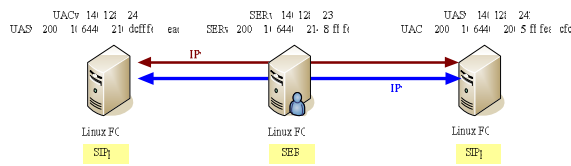


圖 9. 效能測試平台

測試時由 SIPp1 扮演 UACv4。SIPp1 發出的 INVITE 訊息經由 SER 到達 SIPp2 所扮演的 UASv4 端。同一時間，SIPp2 扮演 UACv6。SIPp2 發出的 INVITE 訊息經由 SER 到達 SIPp1 所扮演的 UASv6 端。啟動 SIPp1 及 SIPp2 的指令如下(詳細的參數說明見[13])：

```
SIPp1
#./sipp -sn uas -i
[2001:e10:6440:1:210:dcff:fe6a:eac2] -p 5062
-rsa [2001:e10:6440:1:214:85ff:fe4e:b1e2]:5060
[2001:e10:6440:1:200:58ff:fea0:cfc5]:5063
```

```
SIPp2
#./sipp -sn uas -i 140.128.1.242 -p 5065 -rsa
140.128.1.233:5060 140.128.1.240:5066
```

我們希望 SIPp1 和 SIPp2 分別以每秒 10 個封包的速率來發送 INVITE 訊息，共發送 1000 個。相關指令如下：

```
SIPp1
#./sipp -sn uac -i 140.128.1.240 -p 5066 -rsa
140.128.1.233:5060 140.128.1.242:5065-m 1000
-rp 1000 -r 10 -m 1000
```

```
SIPp2
#./sipp -sn uac -i
[2001:e10:6440:1:200:58ff:fea0:cfc5] -p 5063
-rsa [2001:e10:6440:1:214:85ff:fe4e:b1e2]:5060
[2001:e10:6440:1:210:dcff:fe6a:eac2]:5062 -m
1000 -rp 1000 -r 10 -m 1000
```

圖 10-12 是 SER 伺服器壓力測試的結果。在這些實驗中，SIPp 發送 INVITE 訊息的速率從每秒鐘 10 個依序遞增到每秒鐘 120 個。SER IPv4 介面的封包遺失率如圖 10 所示。從圖中我們觀察到 SER

可以處理的 INVITE 訊息最多為每秒 110 個。圖 11 是 SER IPv6 介面的封包遺失率。結果與 IPv4 介面的情況相同。

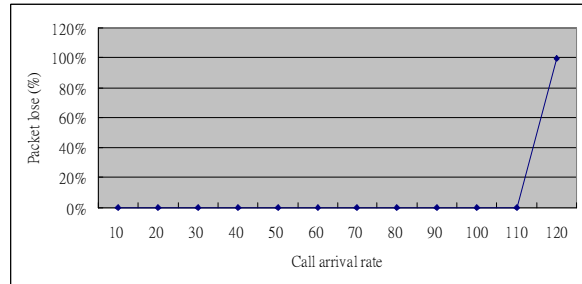


圖 10. SER 流失的 INVITE messages (IPv4 介面)

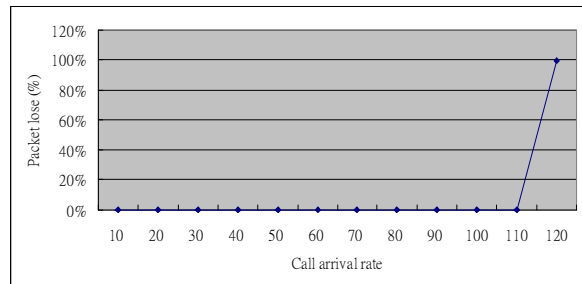


圖 11. SER 流失的 INVITE messages (IPv6 介面)

圖 12 是 SER 的 CPU 使用率。假設 INVITE 訊息的發送率為每秒 n 個。當 $n \leq 60$ 時，CPU 使用率維持在 2% 左右；當 $70 \leq n \leq 120$ 時，CPU 使用率落在 4.8%-6.8% 之間。由此推估，SER 0.96 版的實作相當有效率，不會消耗大量的系統資源。

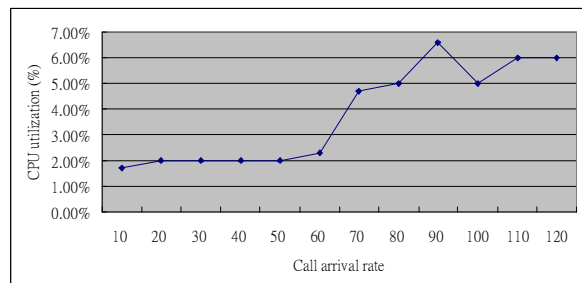


圖 12. SER CPU 使用率

6. 結論與未來工作

本文介紹如何在 IPv6 網路上提供 SIP 語音服務。我們以 SIP Express Router (SER) 0.9.6 版為例，說明 SIP proxy 伺服器的安裝設定；UA 的部分則選用 kphone 3.11 版。另外，我們也討論在沒有 NAT-PT 的環境下，位於 IPv6 網路上的 UA 要與位於 IPv4 網路上的 UA 通話的可能模式，並比較這些模式的優劣。本文最後提供 SER 的壓力測試數據，可作為系統建置的參考。

目前開放原始碼的 UA 多半只支援 Linux 平台。為了讓 IPv6 網路能加速普及，未來我們計畫開發 Windows XP 平台的 dual-mode UA。

誌謝

感謝國家高速網路中心林炳銓先生，當我們遇到 kphone 的安裝問題，或是需要完整的測試環境時，都能適時地得到他的協助，特此致謝。本計畫由國科會贊助，計畫編號 NSC 95-2745-E-126-003-URD。

參考文獻

- [1] J. Rosenberg, et al., "SIP : Session Initiation Protocol, " RFC 3261, June 2002.
- [2] M. Handley and V. Jacobson, "SDP: Session Description Protocol, " RFC 2327, April 1998.
- [3] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification, " RFC 2460, December 1998.
- [4] V. Gurbani and C. Boulton, "Recommendations on the use of IPv6 in the Session Initiation Protocol (SIP), ", draft-gurbani-sipping-ipv6-sip-01.txt, October 2005.
- [5] S. Olson, G. Camarillo, and A.B. Roach, "Support for IPv6 in Session Description Protocol (SDP), " RFC 3266, June 2002.
- [6] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT), " RFC 2766, February 2000.
- [7] Whai-En Chen, Chia-Yung Su and Jui-Hung Weng, "Development of IPv6-IPv4 Translation Mechanisms for SIP-based VoIP Applications", in Proceedings of the 19th International Conference on Advanced Information Networking and Applications, 2005.
- [8] T. Robles, et al., "Porting the Session Initiation Protocol to IPv6", IEEE Internet Computing, pp. 43-50, May 2003.
- [9] SIP Express Router (SER), web site: <http://www.iptel.org/ser>
- [10] Kphone-IPv6, web site: <http://www.iptel.org/products/kphone>
- [11] Kdb2html package, web site: <http://download.fedora.redhat.com/pub/fedora/linux/core/1/i386/os/Fedora/RPMS/kdelibs-3.1.4-4.i386.rpm>
- [12] SIPp, web site: <http://sipp.sourceforge.net>
- [13] Using SIPp to run performance tests, web site: http://sipx-wiki.calivia.com/index.php?title=Using_SIPp_to_run_performance_tests&diff=prev&oldid=4620