

SIP 網路電話計費機制之研究與實作

朱朝吟¹ 翁永昌²
靜宜大學資訊管理學系¹
靜宜大學資訊工程學系²
ycwong@pu.edu.tw

摘要

SIP是目前網路電話控制信令的主要協定之一，可建立、修改、或結束多媒體的通話。本論文針對SIP環境實作一個網路電話計費系統，系統的主要元件包括SER伺服器、RADIUS伺服器、以及應用伺服器。SER伺服器是iptel開發的SIP proxy伺服器，當用戶撥打網路電話時，SER會先詢問RADIUS，以決定是否要建立通話連線。當RADIUS確認了用戶的身分之後，用戶端可以開始通話，同時RADIUS會把用戶使用的資源記錄在本身的MYSQL資料庫中。應用伺服器是以PHP模組開發而成，它將根據RADIUS資料庫中的紀錄向客戶寄送帳單。其他的功能尚有使用者帳號管理、帳務處理、核銷費用、繳費記錄查詢、費率調整、新增SIP server據點等。本文也提供了壓力測試的數據，作為系統建置的參考。

關鍵詞：網路電話、SIP 協定、計費系統、RADIUS

1. 簡介

隨著網際網路的蓬勃發展，企業運用網路無所不在的特性來傳遞語音、視訊及資料是節省電信費用有效的方法之一。SIP (Session Initiation Protocol) 由 IETF 工作小組於 1999 年所制定，是一種利用網際網路來傳送語音及多媒體資料的技術[1]。SIP 負責建立兩個或多個用戶之間的通話(session)，這些通話包括多媒體會議、多媒體發佈、以及網路電話等。SIP 被公認為 IP 網路電話與傳統 PSTN 網路整合的關鍵技術，並且也被第三代行動通訊系統(3G)採用為 IMS 多媒體子系統的通訊標準。

SIP 以純文字方式來傳送網路電話的控制信令，可建立、修改、或結束多媒體的通話。當通話建立起來之後，語音的傳輸則由 RTP [2]協定負責。SIP 可以結合 RADIUS 伺服器[3, 4]以提供用戶端認證、授權、及計費等功能。

本論文針對 SIP 網路電話實作一個計費系統，系統的主要元件包括 SER 伺服器[5]、RADIUS 認證伺服器、以及應用伺服器，系統架構如圖 1 所示。SER 伺服器是 iptel 開發的 SIP proxy 伺服器，當用戶要撥打網路電話時，SER 會先詢問 RADIUS，以決定是否要允許其建立通話連線。當 RADIUS 確認

了用戶的身分之後，用戶端可以開始通話，同時 RADIUS 會把用戶使用的資源記錄在本身的 MYSQL 資料庫中。應用伺服器是以 PHP 模組開發而成，系統管理者可以透過 Web 介面登入應用伺服器來管理使用者，應用伺服器將根據 RADIUS 資料庫中的紀錄向客戶收取費用。

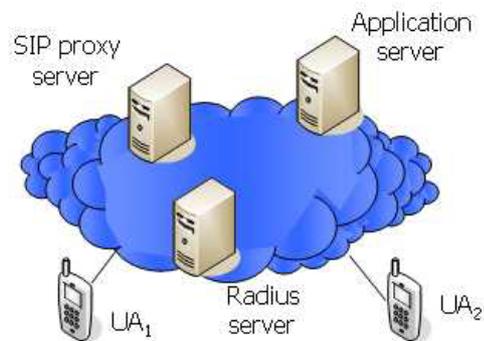


圖1計費系統架構

本文章節安排如下：第二節介紹 SIP 計費流程等相關背景知識；第三節和第四節分別敘述 RADIUS 認證伺服器和應用伺服器的功能與實作；第五節說明系統壓力測試的結果；最後是簡短的結論。

2. 背景知識

我們的系統是採用 FreeRADIUS 搭配 MySQL 資料庫做為認證計費的核心。使用者的帳號密碼儲存在 RADIUS 的 radcheck 資料表中，計費相關資訊則儲存在 radacct 資料表。UA 啟動時必需先至 SIP server 註冊目前所在的位置 (IP 位址) 並核對使用者身份，之後才能撥打網路電話。

圖 2 為 SIP 網路電話認證計費的過程，包含有註冊、邀請、認證請求、認證接受、計費開始、以及計費結束等過程。詳細的作業流程說明如下：

F1：UA發出REGISTER註冊訊息。SIP server在更新位置資訊前會要求RADIUS server確認使用者的身份。

F2：SIP server會啟動RADIUS client，將UA所送來的使用者帳號、密碼等相關資訊封裝在UDP封包中，然後送出access-request給RADIUS server的UDP port 1812。

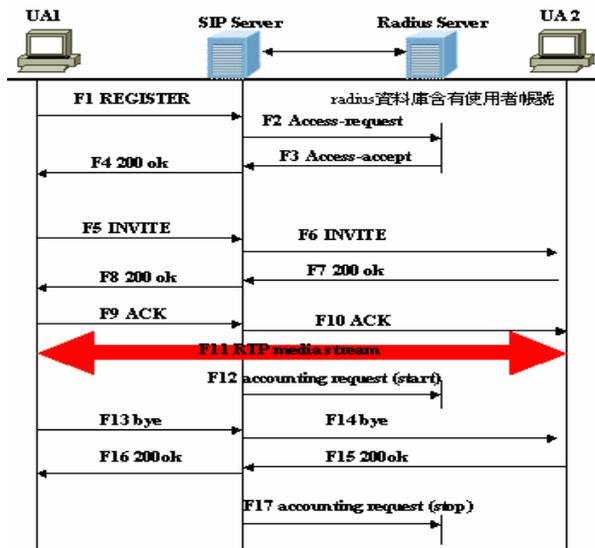


圖 2 SIP 網路電話認證計費流程

F3：RADIUS server收到這個access-request後，會查尋radcheck資料表。如果帳號密碼正確，則回覆access-accept給SIP server。

F4：然後SIP server會送出200 ok給UA，告知註冊成功。

通常UA發出REGISTER並不會攜帶密文摘要認證訊息，此時RADIUS server會找不到密文摘要而認證失敗。於是RADIUS server會向SIP server發出一個含有nonce key的challenge，其狀態碼為401(Unauthorized)，告知UA這個註冊是需要認證的[6]。當UA收到這個訊息時，會使用這個nonce key來產生密文摘要認證訊息，然後再次送出REGISTER給RADIUS Server。RADIUS Server會將收到的密文摘要與自己的密文摘要比較，如果密文摘要相同，則RADIUS server會授予用戶使用網路電話的權限。

F5~F6：當UA1向SIP server發出INVITE訊息來邀請UA2通話。INVITE訊息會攜帶著message body，當中的SDP描述了呼叫端想使用的多媒體的格式、媒體種類、IP位址、Port Number等訊息。SIP server將INVITE訊息轉送給UA2。

F7~F10：當UA2收到這個邀請之後，UA2會依據呼發話端所提出的訊息，做出接受或拒絕的回應。

F11：雙方達成協議後將建立一條RTP session，雙方通話的語音便可透過該session來傳送。

F12：SIP server收到200 OK時(F10)會送出開始計費訊息給RADIUS server，訊息中包含有AcctSessionId、用戶端帳號、SIP server IP位址、計費開始時間、撥打電話的使用者、受話者等資料。這些參數會被封裝在UDP datagram中，傳送給RADIUS server的UDP 1813port。

RADIUS server將資料儲存在radacct資料表的對應欄位中。

F13-F16：UA1送出BYE要求結束通話。UA2收到後回應200 OK。

F17：SIP server聽到200 OK時會送出停止計費訊息給RADIUS server。RADIUS server將通話結束時間存入radacct資料表中。

在上述過程中，所有的SIP message都會被送至SIP server，然後轉送給RADIUS server。由於RTP媒體串流是由兩端的User agent直接處理，不經由SIP server來傳送，所以在radacct資料表中AcctInputOctets及AcctOutputOctets這兩個欄位的值都會是0，表示沒有RTP流量。所以我們的計費系統只以通話時間的長度來計費。

3. RADIUS 認證伺服器

FreeRADIUS執行身份認證的方式有三種：使用登錄本機時輸入的帳號密碼、使用LDAP上存放的帳號密碼、或是使用儲存於本機MySQL資料庫內的帳號密碼。本文採用第三種做法。首先我們一個名為radius的database。

```
# mysql mysql -uroot -p 密碼
> create radius;
```

接著我們在資料庫中產生認證資料表radcheck及計費資料表radacct，前者用來存放使用者的帳號密碼，後者則是存放撥打電話的相關紀錄。產生資料表的步驟如下：(1)從解壓縮後的freeradius.tar.gz檔案中找到「db_mysql.sql」檔；(2)將db_mysql.sql的tables匯入資料庫。

```
# mysql -uroot -p 密碼 radius < db_mysql.sql
```

3.1 radcheck 資料表

當用戶端想要使用網路電話時，UA會藉由SIP server送出認證請求給RADIUS Server。如果radcheck資料表含有這個使用者，則RADIUS server會送“Access-Accept”給SIP server，表示該使用者認證成功；反之，則送出“Access-Reject”，表示認證失敗。radcheck資料表內容如表1所示，欄位說明如下：Username為UA的URI，Value為網路電話帳號密碼。

表 1 radcheck 資料表

id	UserName	Attribute	op	Value
1	gary@sip.ltc.edu.tw	User-Password	=	22793167
3	hsin@sip.ltc.edu.tw	User-Password	=	hsin
54	ltc@sip.ltc.edu.tw	User-Password	=	ltc123
51	lin@sip.ltc.edu.tw	User-Password	=	lin
29	jucc@sip.ltc.edu.tw	User-Password	=	1234
26	lee@sip.ltc.edu.tw	User-Password	=	lee

3.2 radacct 資料表

當 RTP session 建立完成時，SIP server 會送出 Accounting-request 封包給 RADIUS Server，封包中含有使用者帳號以及通話開始的時間。當 RADIUS Server 收到 accounting-request 時會回覆 Accounting Response，告訴 SIP server 已收到該訊息。當任何一端要結束通話時會送出 account-request 給 RADIUS server 請求中斷計費，其中會有通話結束的時間，這些資訊都會被儲存在 radacct 資料表中，做為日後收費的依據。radacct 資料表結構如表 2 所示。

表2 Radacct資料表

Field	Type	Default
AcctSessionId	varchar(32)	
AcctUniqueId	varchar(32)	
UserName	varchar(64)	
Realm	varchar(64)	
NASIPAddress	varchar(15)	
AcctStartTime	datetime	0000-00-00 00:00:00
AcctStopTime	datetime	0000-00-00 00:00:00
AcctSessionTime	int(12)	NULL
AcctInputOctets	bigint(12)	NULL
AcctOutputOctets	bigint(12)	NULL
CalledStationId	varchar(50)	
CallingStationId	varchar(50)	
ServiceType	varchar(32)	NULL

本系統會使用到的欄位說明如下：

AccSessionId：每一通電話會有兩筆紀錄，分別存放通話開始的時間以及結束的時間。這兩筆紀錄擁有相同的 AccSessionId。

UserName：使用者的帳號。

NASIPAddress：SIP server 的 IP 位址。

AcctStartTime：開始通話的時間。

AcctStopTime：掛斷網路電話的時間。

CallingStationId：發話端的 URI。

CalledStationId：受話端的 URI。

3.3 RADIUS 客戶端

安裝 SIP Express Router 0.8.14 版後，尚須完成下面兩個步驟，才能順利透過 RADIUS server 執行認證及計費功能：(1) 在 SER 原始碼中加入 auth_radius 模組及 accounting 模組，然後重新編譯。(2) 將 radiusclient-0.4.8 軟體與 SIP ExpressRouter 整合在同一台伺服器上。radiusclient-0.4.8 軟體會將 RADIUS 協定的認證請求及計費請求封裝成 UDP 封包，然後傳送給 RADIUS Server。

4 應用伺服器之功能與實作

應用伺服器透過系統子程式提供諸如帳號申請、帳號管理、核銷費用、繳費記錄查詢、費率調整、新增SIP server據點管理、帳單郵寄等功能。表 3 描述了系統功能、子程式、與使用的資料表之間的關係。

系統子程式使用 ADODB(Active Data Objects Data Base)來存取資料庫[7]，ADODB是一種 PHP 存取資料庫的函式元件，它最大的優點是存取後端任何種類的資料庫的方式都是一致的。

表 3 系統子程式所使用的資料表

系統功能	程式名稱	API	相關之資料表
計費系統登入	index.php	A D O B	管理者 manager
			用戶端 radacct
帳號申請	useradd.php		usertable
帳務管理系統	userlist.php		列出所有用戶 usertable
	user.php		繳費記錄查詢 payed
			通話明細 radacct
修改帳號	modifyfilter.php		usertable
啟用帳號	enablefilter.php		usertable->radcheck
凍結帳號	disablefilter.php		radcheck
帳單核銷	payfilter.php		payed
郵寄帳單	emallist.php	radacct	
調高費率	charge.php	billinhtype	
更改資費方案	modifyfilter.php	usertable	
新增SIP據點	newclient.php	Linux目錄/tmp/newclient /etc/raddb/clients.conf	
資料庫管理	Phpmyadmin.php	All database	

4.1 系統功能

系統登入畫面如圖 3 所示。重要功能分述如下：



圖 3 系統登入畫面

● 帳號申請

圖 4 為申請網路電話帳號的操作畫面，用戶資料會被儲存在資料表 usertable 中，供相關程式使用。



圖 4 帳號申請

● 帳務查詢

當系統管理員想查詢特定用戶的個人資料、繳費紀錄、通話明細等，可以點選帳務管理系統，然後點選要查尋的對象。圖 5 為帳務查尋的操作畫面。

▶ 帳務管理系統 帳務查詢

帳戶ID: ltc@sip.ltc.edu.tw	姓名: 張俊傑	電子郵件: alex@mail.ltc.edu.tw
帳單地址:	台中市文心路123號	
登錄日期	2005-03-08 16:44:38	狀態: 正常
資費方案: 以分計費		

繳費紀錄

帳號	繳費日期	月份	金額
ltc@sip.ltc.edu.tw	Mon 16, May 2005, 12:00:00	3	120
ltc@sip.ltc.edu.tw	Thu 02, Jun 2005, 07:36:15	4	10
ltc@sip.ltc.edu.tw	Thu 02, Jun 2005, 07:36:16	5	40

Page 1/1

本月通話明細 總通話時數 52 秒 費用 (5 元)

帳號	開始時間	結束時間	受話號碼	通話時間
ltc@sip.ltc.edu.tw	2005-06-04 10:25:15	2005-06-04 10:26:07	sip:gary@sip.ltc.edu.tw	52

Page 1/1

圖 5 帳務查尋

● 凍結帳號

當用戶在每個月的 15 日尚未繳清前一個月的費用時，計費系統會自動在 16 號的凌晨執行 disablefilter.php 程式將其帳號凍結。帳號凍結會將 radcheck 資料表中的使用者帳號刪除，並且會將 usertable 資料表中的 state 欄位註記為「凍結」。

● 啟用帳號

當點選啟動帳號時會執行 enablefilter.php，然後會帶出請輸入帳戶 Id 視窗。當我們輸入被凍結帳號 Id 或新申請帳號 ID 後，會執行 user1.php 程式，這支程式會使用 dbconnect.php 來連接資料庫，然後將 usertable 資料表中的帳號 ID 及密碼寫入 radacct 資料表中的 username 及 value 這兩個欄位。被凍結帳號的使用者，馬上就可以使用 SIP 網路電話。

● 帳單核銷

用戶因逾期繳款而被系統自動凍結帳號，當用戶想要重新啟用帳戶時，必須先繳完費用，再由系統管理者執行核銷帳單，才可以重新啟用帳號。

● 調整費率

用戶計費方式有兩種方式，以分計費及以秒計費。如果要調整費率，系統管理者可以執行 charge.php，該程式會連至 billingtype 資料表，然後更改 type 欄位的內容。欄位 type 表示計費方式，值為 0 是以秒計費，值為 1 是以分計費。另外欄位 unit 表示費率。帳務管理子程式 userdetail.php 在計費時先檢查欄位 type，然後將通話時間乘上費率，即為使用者必須繳交之費用。

● 新增 SIP server 據點

RADIUS server 可以同時服務多台 SIP server，前提是 RADIUS server 的 /etc/raddb/clients.conf 檔案內必須含有這些 SIP server 的 IP 位址以及 shared key。要增加 SIP server 據點時，系統管理者可以點選新增 SIP server 據點，此時會執行 newclient.php 程式，然後會帶出視窗讓我們輸入 SIP server 據點 IP 位址及 shared key。newclient.php 會將此輸入資料寫入 /tmp/newclient/aaa 檔案，然後在 crontab 中建立每天 09:00AM 將此檔案附加在 /etc/raddb/clients.conf 檔案最後面。當附加檔案完成後，再將 /tmp/newclient/aaa 檔案刪除，就可以將新增的據點納入系統的營運範圍內。

4.2 程式碼範例

本小節以計算當月份總通話時間為例（參考圖 5），說明相關程式碼的細節。程式如下：首先每個用戶打電話給受話者，從受話者接受邀請到結束電話，在 radacct 資料表中會儲存通話開始計費及通話結束計費兩筆記錄，並且這兩筆資料中的 AccUniqueID 是一樣的，我們的帳務系統會將最大的 AcctStopTime 減去最小 AcctStartTime 而得出通話時間。

```
01 include('dbconnect.php');
02 include('adodb/adodb-pager.inc.php');
03 include('checkuser.php');

04 $userid = $HTTP_COOKIE_VARS["userid"];
05 $passwd = $HTTP_COOKIE_VARS["userpw"];

06 $month = $_GET["month"];
07 $range_start = mktime(0,0,0,date("m"),1,date("Y"));
08 $range_end = mktime(0,0,0,date("m")+1,1,date("Y"));
09 if($month) {
10     $range_start = mktime(0,0,0,$month,1,date("Y"));
11     $range_end = mktime(0,0,0,$month+1,1,date("Y"));
12 } else {
13     $month = date("m");
14 }

15 if($userid) {
16     $sql = "select UserName as '帳號', MIN(AcctStartTime)
17           as '開始時間', MAX(AcctStopTime) as '結束時間',
18           CalledStationId as '受話號碼',
19           UNIX_TIMESTAMP(MAX(AcctStopTime)) -
20           UNIX_TIMESTAMP(MIN(AcctStartTime)) as '通話時間',
21           from radacct where UserName='$userid' AND
22           AcctStartTime > FROM_UNIXTIME($range_start) AND
23           AcctStartTime < FROM_UNIXTIME($range_end)
24     ";

25 $sum = 0;
26 $rs = $conn->Execute($sql);
27 while (!$rs->END) {
28     $sum = $sum + $rs->fields[4];
29     $rs->MoveNext();
30 }
}
```

Line 01：用來連接 Mysql 資料庫。這支程式會啟動 adodb，並且使用 ADONewConnection() 來建立一個連結物件。要連接資料庫時呼叫 dsn() 函數，dsn() 函數含有連接 Mysql 資料庫的帳號、密碼。

Line 02：引入分頁功能。

Line 03：檢查使用者是否有認證，若認證有問題則跳出網頁。

Line 04-05：userid及passwd的值是經由HTTP cookies的方式所取得的，如果browser cookies存在就不需要再認證。

Line 06-12：變數 \$range_start表示從1970-01-01 00:00:00開始至本月1日的0時0分0秒為止之Unix時間戳記。變數 \$range_end表示從1970-01-01 00:00:00開始至次月1日的0時0分0秒為止之Unix時間戳記。

Line 13-14：查詢使用者每通電話的明細列表，包含帳號、開始時間、結束時間、受話號碼、通話時間。

Note 1. UNIX_TIMESTAMP(MAX(AcctStopTime))

假設 MAX(AcctStopTime): 為 2005-06-04 10:26:07，它是一個字串，必需轉換為Unix時間戳記，也就是從1970-01-01 00:00:00'開始至2005-06-04 10:26:07的秒數。

Note 2. FROM_UNIXTIME(unix_timestamp)

將 Unix 時間戳記轉成以 'YYYY-MM-DD HH:MM:SS'的字串。

Line 16：SQL指令執行後，一個 ADORecordSet物件會被回傳回來，其中fields[0]是帳號，fields [1]是開始時間，fields[2]是結束時間，fields[3]是受話號碼，fields[4]是通話時間。

Line 17-19：計算總通話時間。當EOF的值被設成ture時，就是最後一筆。

5 系統效能評估

本節將評估 SIP proxy server 以及 RADIUS server 在單位時間內所能處理的最大連線數。圖 6 是我們的測試環境。詳細的系統規格如表 4 所示。另外，表 5 是 Mysql 資料庫微調後的參數設定。

圖 6 中 SIP server 包含了 SIP proxy server 以及 RADIUS server。我們另外將 SIPp 測試軟體[8]安裝在 UAC 及 UAS 上，用以發送大量的 SIP message。當我們想要建立或終止一個連線測試流程時，我們可以將要連線測試的流程預先定義在 XML scenario 中。SIPp 會將測試的結果匯成 csv 檔，供我們做系統壓力分析。

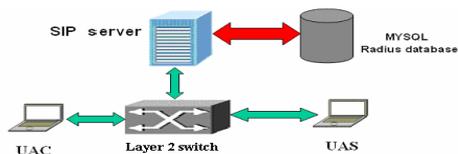


圖 6 效能測試架構圖

表 4 系統規格

硬體規格	Intel Pentium 4 CPU 2.8 GHZ，並具有 1GRAM	安裝 SIP Server、Radius server
SIP Server	VoIP軟體是iptel的SIP Express Router(SER0.8.14)	SIP message request/response 處理
	radiusclient的版本為0.4.8	負責送出 authentication request 及 accounting request 給Radius server
Radius server	Freeradius-1.0.1-1版本	負責認證、授權、計費
	資料庫mysql-3.23.58-13版本	
User agent	測試軟體採用SIPp	UAC為發話端 UAS為接收端

表 5 Mysql 資料庫參數設定

Variable_name	Value
max_connections	2048
key_buffer_size	16M
sort_buffer	4M
record_buffer	1M
table_cache	128

5.1 SER 壓力測試

圖7為SER的壓力測試結果。首先我們使用SIPp軟體發送每秒10個SIP INVITE messages至UAS，歷時300秒。此時SIP Server每秒會送出10個Accounting request 封包儲存在Radacct資料表中。然後SIPp以10為一個單位逐漸增加call Rate至130。

圖中說明當call rate在30以下時，連線成功的比率幾乎是100%。當call rate在40至60之間時，連線成功的比率將從80%下降到50%。當call rate超過100以後，連線成功的比率將掉到30%以下，此時SER的效能變的非常差。

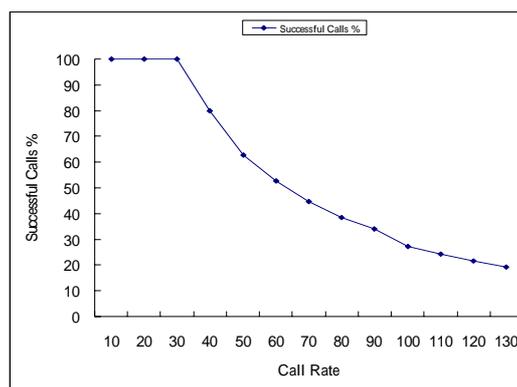


圖 7 SER 的最大連線數

5.2 RADIUS server 壓力測試

圖8為RADIUS server 的壓力測試結果。首先我們使用SIPp軟體在一秒中之內發送20個SIP Request至UAS，歷時1秒。此時會在RADIUS 資料庫建立20筆sessions，每一筆session包含有accounting-request start及accounting-request stop兩筆記錄。然後SIPp以20為一個單位逐漸增加發送的

Request數目至每秒280個。

由圖我們可以很清楚的知道，當call rate 在20至200之間時，RADIUS 資料庫可以正確地儲存每一筆記錄。當call Rate在200至260時，RADIUS 資料庫會遺失部份的計費資料。當call rate 增加至280時，RADIUS 資料庫crash掉而無法運作。結論是我們的計費資料庫最多可同時建立200個sessions。

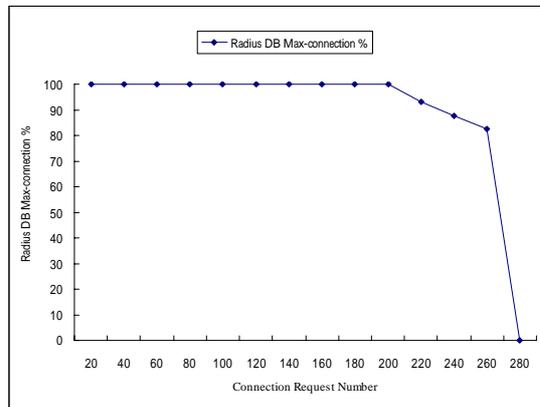


圖 8 RADIUS server 同時可建立的最大連線數

6 結論

本文實作一個SIP網路電話計費系統。我們利用 iptel 所開發的 SIP Express Router 軟體來架設 SIP proxy。我們在 SIP Express Router 中增加 radius 模組及 accounting 模組，並且整合了 Radius-client 0.4.8 版軟體。我們使用 PHP 來撰寫應用伺服器，已提供使用者帳號管理、SIP proxy 節點管理、計費系統管理、用戶帳務查詢、帳單自動郵寄等功能。

我們在維護系統時發現，當發起通話的用戶未正常離線時，SER 不會主動傳送計費中斷請求至 RADIUS，造成 radacct 資料表沒有計費結束的記錄。要解決上述問題，可以考慮使用 Diameter[9] 認證及計費協定來取代 RADIUS。SER 也支援 Diameter 模組，實作以 Diameter 為核心的計費系統是我們未來努力的方向之一。目前我們的計費系統是運行在 IPv4 網路，但是 IPv4 網路有 IP 位址不足、NAT 穿透等問題。未來我們也希望能將系統升級到 IPv6 環境。

參考文獻

- [1] J. Rosenberg, et al., "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [2] H. Schulzrinne, et al., RTP: "A Transport Protocol for Real-Time Applications", RFC 1889, Jan. 1996.
- [3] C. Rigney, S. Willens, A. Rubens and W. Simpson, "Remote Authentication Dial In User Service", RFC 2865, June 2000.
- [4] C. Rigney, "RADIUS Accounting", RFC 2866, June 2000.

- [5] SIP Express Router, <http://www.iptel.org>.
- [6] J. Franks, P. Hallam-Baker, J. Hostetler, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.
- [7] <http://phplens.com/lens/adodb/docs-adodb.htm>.
- [8] SIPP, <http://sipp.sourceforge.net>.
- [9] P. Calhoun, et al., "Diameter Base Protocol", RFC 3588, Sep. 2003.