

SIP 網路電話雲端服務系統之設計與實作

陳德安 陳泰安 李尚軒 蘇雍太 蘇暉凱*

國立虎尾科技大學

hksu@nfu.edu.tw*

摘要

由於網路日漸普及與多媒體應用迅速的發展，網路電話 VoIP (Voice over IP) 服務已廣泛應用於日常生活中。面對未來一個 All IP 的網路架構，SIP 已成為未來通訊信令共同基礎，然而目前網路電話伺服器 (SIP Server) 大部分採集中式架構，當系統話務量增加，將嚴重造成伺服器超載，影響使用者通話的品質；此外網路電話用戶端大部分為半封閉式系統，網路電話的兩端之服務提供業者緊密結合，限制了使用者的發展空間及自由度，網路電話加值服務受限於提供服務的業者。因此，本論文設計與實作 SIP 網路電話雲端服務，系統可分為兩大部分，第一部份為 SIP 網路電話雲端子系統，第二部份是 VoIP 終端機子系統，在 SIP 網路電話雲端子系統，我們能依伺服器資源使用情況做動態控制，不但可減輕伺服器負擔，還能做伺服器故障轉移，使網路電話服務更穩定、服務品質更優質化。在用戶端的部分，則打破現有的限制，利用 Google 推出的 Android 開放系統，同時結合了雲端運算，研發出開放式程式碼且以 SIP 技術為主的網路電話，其優點於方便、可攜性高、符合人性介面，也結合了更多的服務，如：Gmail、Google Map、Google Talk 等...，面對於未來的通訊世界，達到更多元化服務的影響力。

關鍵詞：網路電話、SIP 會談初始協定、Android、雲端運算。

1. 簡介

網際網路已經漸漸的成為我們生活中不可或缺的一部份，從早期的數據機，上網速度僅 56 kbps，到現在人人都是寬頻 XDSL 與 EPON/GPON，且連線速度已可達到數十 Mbps 的水準，這代表網路電話已經突破了過去連線速度的環境限制，使目前網路電話服務提供穩定的連線環境，並進一步對於現今各種不同互連限制加以改善，以達到一個完整的網路電話系統。

當前網路電話伺服器處理資料的方式以集中式架構為主，使用者必須透過集中式的伺服器來進行通話，當用戶大量增加會嚴重增加伺服器的負荷，且用戶在伺服器處於負荷很高的狀態下使用服務，所得到的服務品質則大幅下降，這也是目前 SIP

網路電話服務所面臨的瓶頸。網路電話 (VoIP; Voice over IP) 終端機大部分以半封閉式的系統為主，在操作應用軟體的過程中，只能夠使用系統所提供的應用程式進行系統的操作，而且，在開發者部分也無法開發附加元件或與其他系統做整合。

本論文提出開放式 SIP 網路電話雲端服務系統，以改善網路電話半封閉式與伺服器集中式架構的問題，藉由開放 API 的方式，使其他開發者在開發附加元件與系統整合的過程，沒有系統相容方面的問題，在伺服器架構則改用雲端運算結構，不但降低了伺服器的負荷，也大幅提升了服務品質，另外在網路電話服務 (VoIP) 系統方面我們採用 SIP 標準規格，在與其他 VoIP 系統進行協定溝通時，可免除相容性的問題。

2. 背景

本論文相關背景技術可分為三大部分：第一部分將介紹 SIP Server，第二部分為雲端服務，最後部分為 Android 系統架構與相關網路電話軟體介紹。

2.1 SIP Server

SIP 訊息的傳遞除了最簡單的點對點通訊外，尚需仰賴 SIP Server 提供身分認證、位置管理、訊息傳送、通話管理等等，都需要透過伺服器。因此，SIP Server 可說是 SIP 通訊中的樞紐[4][6][7][8]。

SIP Server 主要是控制所有 SIP 訊息，全部的 SIP 訊息都會進入 SIP Server 進行處理，由 SIP Server 負責轉送至其他用戶或是其他伺服器，SIP Server 儲存所有註冊用戶的資料包含帳號、密碼、登入位置等，也可用來設定一些特殊功能，例如：轉接、語音信箱、回撥等...[4][6][7][8]。

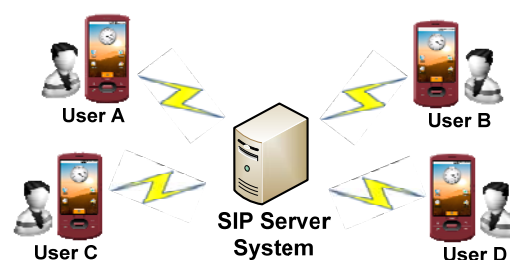


圖 1 伺服器集中式架構

如圖 1，目前的 SIP Server 大多採用集中式的架構，使用者必須透過集中式的伺服器來進行通話，當瞬間有數千筆甚至是數萬筆通話請求大量進入時，會快速增加伺服器的負載，使用者的通話品質也會因此受到影響；此外，集中伺服器風險很高，若伺服器突然發生故障，將會導致所有使用者網路電話服務異常。

2.2 雲端運算

雲端運算不是新技術，而是一種概念，主要是透過網路，由多部電腦所組成的龐大系統，可以在數秒之內，處理數千筆資料，達到和「超級電腦」同樣強大的效能，其核心技術為分散式運算 (Distributed Computing) 與網格運算 (Grid Computing) [9]。

分散式運算，是將多部獨立的電腦系統以網路匯集起來，一起做資料的處理，並將資料分割成小型資料，再交由眾多電腦各自進行運算再彙整結果。其優點為成本低、資料處理速度快、系統可靠度高並且系統擴充容易。

現階段雲端分為三大服務，分別為 IAAS (Infrastructure as a Service) 基礎架構服務，PAAS (Platform as a Service) 平台服務及 SAAS (Software as a Service) 軟體服務。其中 IAAS 是將作業系統及資料庫等透過網路方式提供服務，PAAS 是經程式平台 (如 JAVA、.Net 等) 透過網路方式提供服務，SAAS 則是將軟體應用在網路上提供服務，本研究著重於 IAAS 設計。

2.3 Android

2007 年 11 月 5 日，Google 與其他製造商、硬體供應商組成的開放手持裝置聯盟 (Open Handset Alliance)，發佈了名為「Android」的開放手機軟體硬體平台，在平台公佈的一週之後，即發佈可免費自由下載的多平台的 SDK (Software Development Kit) 與相關文件，隔數天又發佈作業系統核心 (Kernel)，與部分驅動程式的原始碼。而 Android 的正式版本是從 2008 年 9 月 23 日的 1.0 版本慢慢演進 1.1、1.5、1.6、2.0，到 2010 年 05 月最新發佈的 2.2 版本，Android 開發環境中所使用的應用程式都是由 JAVA 編寫，所以需要有 JAVA 的工具包以及支援 Android 的擴充套件[1]。

目前 Android 上有許多免費網路電話應用程式，像是 Fring、iSkoot 或 Nimbuzz 等...，這些應用程式都是半封閉性，所以必須緊密與軟體提供業者結合，而無法隨著使用者的需求發展延伸加值服務，因此對於使用者而言限制了許多彈性。

對於目前網路電話軟體可分成三大類：封閉式、半封閉式以及開放式這三種類型的系統模式。而封閉式系統就如一般市面常見的網路電話軟體，只提供一個固定的系統環境，並且不開放 API 等資源給使用者做介面加值的動作；半封閉式系統

則有提供 API 等資源給使用者，在系統介面上能夠做加值服務的動作，但並未公開程式碼與限制性授權；最後開放式系統則是公開程式碼提供給使用者自由修改，有些開放軟體則會透過 svn 等平台，管理不同使用者所修改的程式碼，以利軟體版本的更新。

相較於封閉式系統，半封閉式提供系統功能的可選擇性，但無法提供完全開放的系統環境，導致使用者在修改系統環境的能力受到限制，進而影響程式在不同系統環境相容性的問題及系統功能在升級效能的發展空間，對於使用者方面也降低使用半封閉系統的意願，以及抑制了系統自由發展的空間。

3. 系統架構

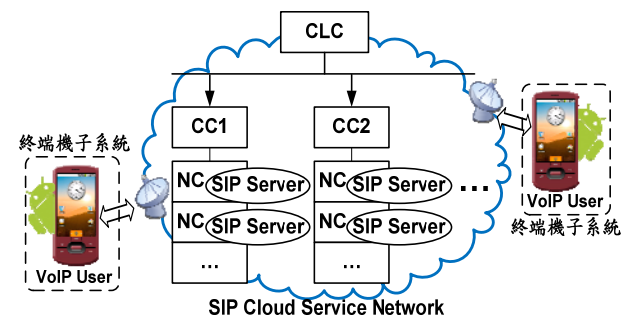


圖 2 系統架構圖

圖 2 為本系統架構圖，終端機子系統部分使用了 Android 嵌入式系統，並且在此系統環境下開發網路電話終端機系統；另一部分則是網路電話雲端子系統，使用 EUCALYPTUS System 的架構，建置出雲端伺服器，透過 EUCALYPTUS 內部的查詢機制，使網路電話雲端子系統有別於傳統 SIP 伺服器架構，以改善傳統網路電話伺服器集中式負載與伺服器單點錯誤所造成服務停止的缺點，提供穩定的網路電話服務。

終端機子系統內主要包含 SIP Call 與地圖定位兩個功能，SIP Call 部分與一般傳統網路電話的半封閉式系統不同，而是有著自由開放的開放源碼特性，透過此特性提升系統的擴充性，並將地圖定位與網路電話做整合，以達到一般網路電話所沒有的地圖互動，而終端機子系統可連接至網路電話雲端子系統，進行 SIP Call 的通話動作。

網路電話雲端子系統，採用 EUCALYPTUS System 架構[2]，EUCALYPTUS 提供完整雲端架構，以 CLC (Cloud Controller) 為最上層，是控制整個雲端系統的中心，並可透過 CC 掌握整個系統資源，再來是 CC (Cluster Controller)，是 CLC 與 NC (Node Controller) 聯繫的媒介，經由 CC 可掌握下層的 NC (Node Controller) 負載情況，並做系統資源動態調整，但對於 SIP Server 而言，EUCALYPTUS System 架構並沒有提供網路電話服

務的控制與管理，所以我們在 EUCALYPTUS System 增加 SIP Server 管理與控制元件，把雲端和 SIP Server 整合，以改善 SIP Server 負載集中與解決單點錯誤造成的服務中止。

4. 系統元件設計

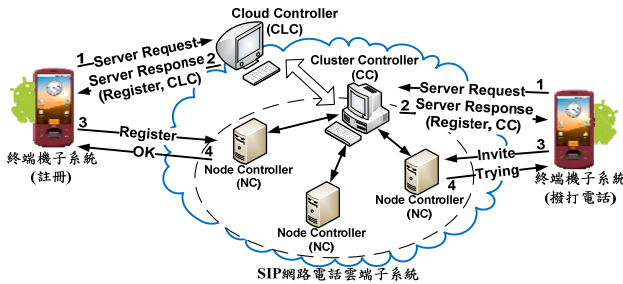


圖 3 系統元件架構圖

如圖 3，雲端網路電話系統分為兩個子系統，分別為 SIP 網路電話雲端子系統與終端機子系統的設計。

SIP 網路電話雲端子系統是建立在 EUCALYPTUS System 架構上，利用最上層的 CLC (CLOUD Controller) 作為雲的主控制，並透過 CC (Cluster Controller) 控制 NC (Node Controller) 進行資源管理與分配，因此我們利用此架構，將 SIP Server 整合在 NC 上，經由 CC 監控管理依附的 NC，並即時回報訊息給 CLC，以達到動態控制的目的，當 NC 發生錯誤，使得通話服務無法進行時，即立刻透過 CC 回報至 CLC，並將系統服務轉移至其他，以維持網路電話服務的進行。

終端機子系統建置在 Android 的作業系統環境下，將網路電話與 Google 的應用服務 (例如：Google Map、Gmail、Google Contacts 等...) 做嵌入式的整合[3]。當終端機首次與網路電話雲端子系統連接時，必須先向 CLC 發出請求訊息，以查詢何處有可用 NC 之 SIP Server 進行註冊；當 CLC 回覆後，終端機即立刻連結至可使用的 SIP Server 進行註冊動作，以完成註冊。若非初次通話，則可免除註冊動作，並且終端機直接向 CC 傳送請求訊息，查詢何處有可用 NC 之 SIP Server 代理訊息傳送，當 CC 回覆訊息時，終端機即可直接連接至指定 SIP Server 進行通話動作。

4.1 VoIP 終端機子系統

VoIP 終端機子系統主要建構於開放式的 Android 系統平台，使用開放源碼的網路電話軟體 Sipdroid，透過開放 API 的 Google Map，進行網路電話與地圖定位的系統服務整合，再經由符合人性操作界面的 GUI，將這兩個獨立的服務元件整合成完整的網路電話終端機系統，以便在 Android 嵌入式開發平台操作。

而系統將保留 Sipdroid 的開放源碼特性[5]，以利日後對於終端機系統改進的空間與自由度，並且降低終端機系統擴充相容性問題，提升了系統改進的空間。

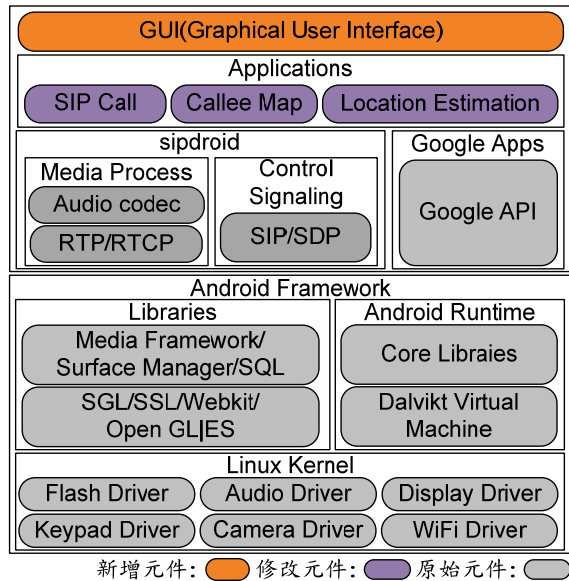


圖 4 VoIP 終端機子系統架構圖

如圖 4 所示，系統底層為 Android 結構，採用 Linux 核心，接著函式庫建立於核心層的架構之上，在同層架構上又包含 Android Runtime，而應用程式層須透過函式庫取得系統檔進行系統動作執行，最後應用服務層則是透過呼叫應用程式層的方式執行所下達的動作。在架構圖中，橘色方塊代表開發時所新增元件，而紫色方塊為修改後再加入的元件，灰色方塊則是無更動的原始元件：

(1) GUI (圖性用戶介面)

GUI 實作部分與系統以及使用者之間有相當密切的關係，透過完善的 GUI 設計，以減低使用者的認知負擔，以及提高系統服務的相容性。

(2) SIP Call

SIP Call 部分是以開放源碼特性的 Sipdroid 修改，所以通話功能保留部份元件功能，並且透過連接雲端伺服器的方式，以達到支援雲端服務之 VOIP 終端機。

(3) Callee Map

透過 Google 提供的開放 API 的方式，將 Google Map 與網路電話整合，藉由地圖的定位服務，提供使用者之間的定位互動，以達到進階的通話互動，並在通話中，可以於 Google Map 顯示通話對方之位置。

(4) Location Estimation

地圖定位的精準度處理部分，則是先蒐集 Google Map 預測之座標點，再透過機率統計的方

式，將計算後權重值最高的座標點定為最佳作標點，經由 Location Estimation，以提高地圖定位的精準度。

4.2 SIP 網路電話雲端子系統

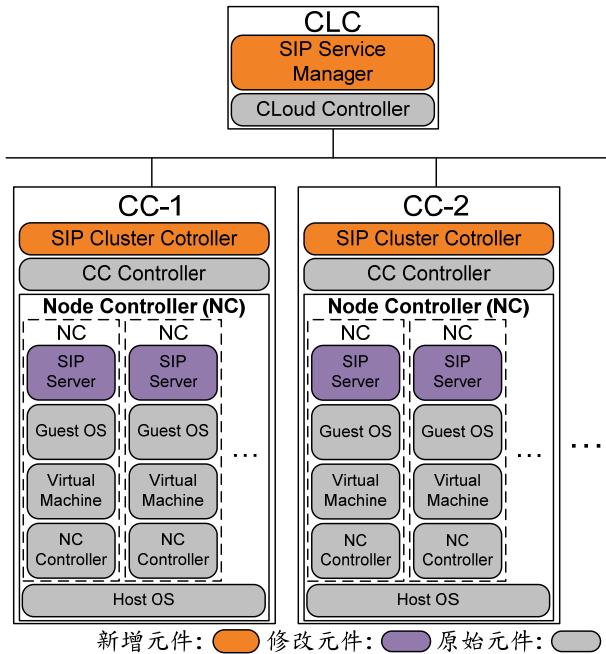


圖 5 SIP 網路電話雲端子系統功能方塊圖

由圖 5 可知，此系統最上層為 CLC (Cloud Controller) 也是系統主控制元件，主要透過網路與各個 CC (Cluster Controller) 連結，並控制虛擬出來的 NC (Node Controller) 做系統資源控制，使架設在 NC 上的網路電話伺服器 (SIP Server) 能夠有效的完成系統資源動態控制，並可隨時改變伺服器資源，以達到最高效能。假若其中 NC 當機或損壞，則經由 CC 回報給 CLC 並做系統調節，讓使用者依然能夠繼續使用網路電話服務。

當終端機進行網路電話註冊時，CLC 上的 SIP Service Manager 會提供使用者目前負載最低的 CC，並回傳使用者後，讓使用者直接與 CC 下層的 NC 之 SIP Server 做網路電話註冊的服務，而進行網路電話通話服務時，使用者會向 CLC 提出請求，CLC 會回傳系統負載量較低的 CC 給使用者，使用者再透過 CC 下的 SIP Cluster Controller 與 NC 上的 SIP Server 進行通話服務。

因此，本研究在 EUCALYPTUS System 下設計 SIP Service Manager 和 SIP Cluster Controller 為目標，透過 EUCALYPTUS System 建立的雲端結構，經由 SIP Service Manager 與 SIP Cluster Controller，可以使 SIP Server 做有效的系統資源控制，並改善傳統網路電話伺服器負載集中與單點錯誤所造成的系統服務終止，因此本論文的设计具有使網路電話效率提高和單點錯誤服務不中斷的優點。

4.2.1 CLC (CLoud Controller) 元件

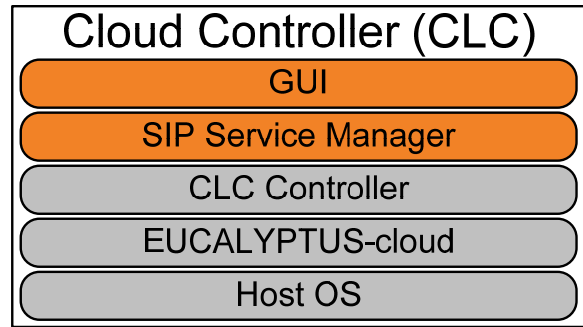


圖 6 CLC 功能方塊圖

如圖 6 為 CLC 功能方塊圖，在 Linux 作業系統上建立 EUCALYPTUS-cloud，並設計 SIP Service Manager 和 GUI，以達到方便管理與系統資源狀態控制。

(1) GUI 設計

設計圖形化介面，使管理人員容易掌控雲端網路電話服務系統的狀況。

(2) SIP Service Manager

此元件設計是透過網路與 CC 聯繫，並遠端控制 CC 下層的 NC 狀況。根據網路電話服務系統的負載，去做系統資源的調整，並動態分配負載低的 SIP Server 進行網路電話註冊服務。

4.2.2 CC (Cluster Controller) 元件

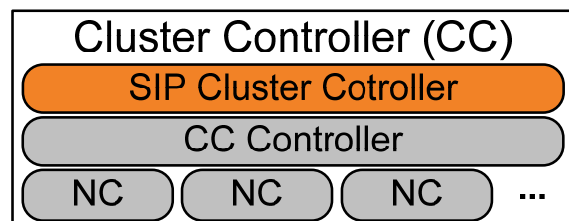


圖 7 CC 功能方塊圖

圖 7 為 CC 功能方塊圖，CC 為 CLC 與 NC 中間的連接媒介，在 CC 的下層由許多虛擬 NC 組成，以 CC Controller 進行資源控管，並將訊息傳導至 CLC，以達到系統資源控制，並透過 SIP Cluster Controller，可動態指派 NC 上的 SIP Server 進行網路電話代理服務。

在 CC 上我們額外設計 SIP Cluster Controller 以管理下層 NC 所有 SIP Server 資訊，此設計是在 CC 上查詢虛擬 NC 群，隨著資源變動，CC 會將系統資訊透過 CC Controller 傳至 CLC Controller，並透過 SIP Cluster Controller 提供負載低的 SIP Server 給使用者進行網路電話註冊服務或網路電話通話服務。

4.2.3 NC (Node Controller) 元件

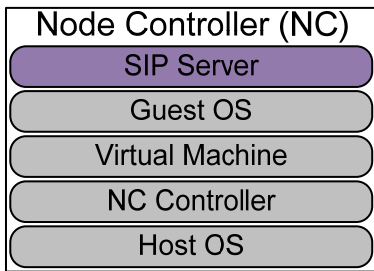


圖 8 NC 功能方塊圖

圖 8 為 NC 功能方塊圖，在 Host OS 上利用 NC 的 Virtual Machine，並在虛擬環境建置 SIP Server，並透過 NC Controller 將 SIP Server 的資源狀態回傳至 CC Controller，為提供資源動態調整，我們修改 SIP Server 資料同步機制，以提供 SIP Service Manager 與 SIP Cluster Controller 進行管理與控制網路電話服務。

5. 使用案例說明

本系統情境主要分為註冊以及通話兩種，說明如下。

5.1 註冊

如圖 9 為描述進行註冊的情境，Client 端會先發出服務請求至 Cloud Controller (CLC)，此時 CLC 會找到目前可用的 Cluster Controller (CC)，並且將訊息傳回給 Client 端，確認後使用者會再透過 CC 指派可用的 NC 進行註冊，當伺服器處理完成註冊訊息時，會回傳 OK 的訊息至 Client 端。

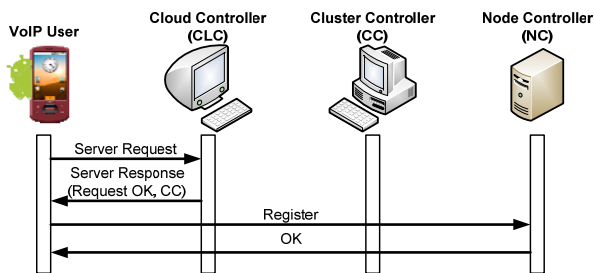


圖 9 系統註冊情境圖

5.2 建立通話

圖 10 描述的情境為 User A 與 User B 進行通話連接，首先 User A 端發出服務需求至 Cluster Controller (CC)，CC 會回傳需求成功並提供目前有哪些負載輕的 Node Controller (NC) 進行通話邀請，確認之後 NC 會與使用者兩端通道連結，當通道連結完成時，受話端則會回傳完成訊息通知。

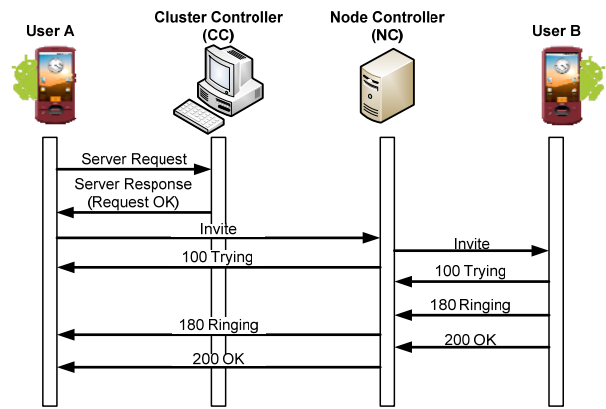


圖 10 系統通話情境圖

6. 系統測試與結果

6.1 測試流程

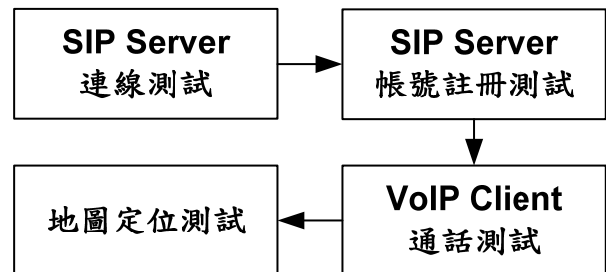


圖 11 系統測試流程

圖 11 為系統測試流程，透過此測試流程可檢測目前系統是否完善，以解決系統所發生的錯誤，首先我們先以 SIP Server 部份進行系統檢測，如功能正常，再對終端機部份進行後續的測試，當兩邊系統完成後，會進行系統整合，並且檢測系統整體是否有相容性的問題。

6.2 測試結果

(1) SIP Server 連線測試

圖 12 為終端機尚未於 SIP Server 連線，可以看到左上角的指示燈呈現紅色，而圖 13 為連線成功，左上角的燈號會變成綠色。

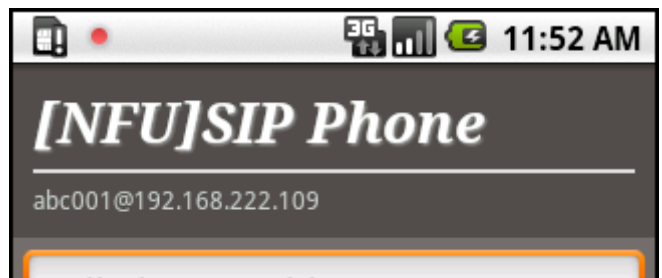


圖 12 終端機尚未連線



圖 13 終端機連線成功

(2) SIP Server 帳號註冊測試

圖 14 為終端機註冊後，黃色方框為 SIP Server 內部註冊清單。

```

終端機 - root@cloudclic: /home
檔案(F) 編輯(E) 檢視(V) 終端機(T) 前往(G) 說明(H)
root@cloudclic: /usr/local/etc/partysip
chan@cloudclic:~$ sudo su
[sudo] password for chan:
root@cloudclic: /home/chan# partysip -d 6 -i
<userinfo>
#SIP service register list
# static user list of known users
# for "user" type:
# <user> <sip:NAME@DOMAINNAME> <auth_login> <
user sip:jack@192.168.222.109 jack 1234
user sip:abcdef@192.168.222.109 abcdef 123456
user sip:abc001@192.168.222.109 abc001 123456
user sip:abc002@192.168.222.109 abc002 123456
</userinfo>

```

圖 14 SIP Server 註冊帳號清單

(3) 通話測試結果

圖 15 為通話中的影像，透過 Sipedroid 進行通話動作，而圖中也顯示受話者的名稱。

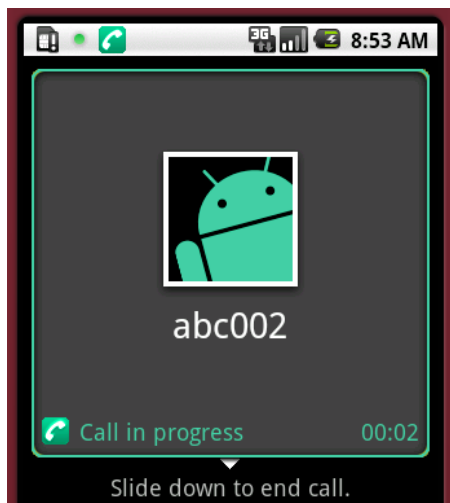


圖 15 終端機通話連線圖

(4) 地圖定位測試

圖 16 為執行終端機系統內的地圖，定位使用者目前所在地的圖像，由圖可察知地圖上有藍色圖標標示使用者目前的所在地。

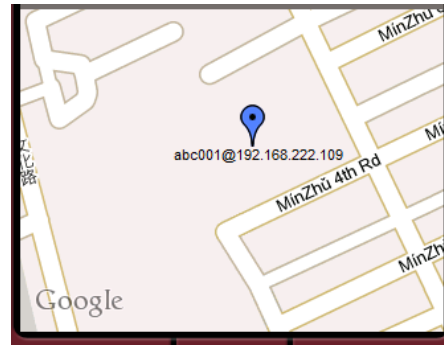


圖 16 終端機地圖定位測試

7. 結論

本論文提出開放式 SIP 網路電話雲端服務系統，以改善網路電話半封閉式與伺服器集中式架構的問題。在 SIP 伺服器部份，本論文運用現有 EUCALYPTUS Open Source 結構，成功將 SIP Server 與雲端運算整合在一起，實現 SIP 網路電話雲端服務；在 SIP 終端機部份，本論文採用 Android 開放性結構，實現支援本系統之 SIP 網路電話雲端服務與網際網路公開式雲端服務之網路電話終端機，網路電話終端機可以成功使用本系統支援端服務與 Google Map 之公開式雲端服務。因此，本系統改善 SIP 集中伺服器負荷超載與單點錯誤造成服務中斷的問題，並且提供 SIP 網路電話服務與雲端服務功能性整合之成功案例。未來在 SIP 網路電話雲端服務的資源分配與優化，相信會有更多議題值得探討。

參考文獻

- [1] Android Developers, <http://developer.android.com/index.html>
- [2] Eucalyptus systems, <http://open.eucalyptus.com/>
- [3] Google code, <http://code.google.com/intl/zh-TW/more/>
- [4] SIP: Session Initiation Protocol <http://www.apps.ietf.org/rfc/rfc3261.html>
- [5] sipdroid - Project Hosting on Google Code, <http://code.google.com/p/sipdroid/>
- [6] 周柏超,“階層式 SIP 網路環境的設計與實現”, 國立成功大學工程科學系碩博士班, 碩士論文, 2004 年 7 月。
- [7] 周暉禎,“一個雲端計算平台上針對互動式工作流程應用的最小負載分配法之動態資源供應架構”, 國立交通大學資訊科學與工程研究所, 碩士論文, 2009 年 9 月。
- [8] 秦浩倫, 林錦惠, 馮輝文,“SIP 伺服器之研究與建置”, 國立台灣科技大學資訊工程系, 2005 年。
- [9] 鄭坤益,“針對雲端計算環境建置以分散式雜湊表為基礎之同儕式檔案系統”, 國立高雄大學資訊工程學系, 碩士論文, 2009 年 8 月。