

一個高效率的 VoIP 使用者位置搜尋方法

王讚彬

國立台中教育大學資訊科學學系
tpwang@ms3.ntctc.edu.tw

邱冠霖

國立中正大學資訊工程學系
Kaulin.chiu@msa.hinet.net

摘要

網際網路的成功在於低成本的服務價格及無窮止境的資源，這優勢也促成了網路電話(Voice over IP)的盛行，相較於傳統的電信服務，網路電話提供了更多樣的服務，例如允許使用者同時使用多種不同功能的通信設備。當用戶採用以 Session Initiation Protocol(SIP)為基礎的網際網路電話作為傳送語音服務的工具時，該用戶就可以享有多重註冊的服務(Multiple Registration)，而系統也會同時記錄多筆該用戶的網路位置。本篇論文探討如何有效快速搜尋用戶實際位置以降低系統負擔，傳統作法是直接採用平行搜尋(Pure Parallel Search)，但這作法非常浪費系統網路資源，因此，本論文結合循序(Sequential Search)及平行搜尋的優點，依據使用者的習慣動態搜尋使用者，這樣的作法可以快速搜尋使用者的真正位置前提之下大幅減低網路資源的耗用問題。

關鍵詞：網際網路電話，Session Initiation Protocol，位置管理。

1. 簡介

低成本的服務價格及無窮止境的資源促成了網路電話(Voice over IP)的盛行，相較於傳統的電信服務，網路電話提供了更多樣的服務，然而網路頻寬不足，使得網路電話語音品質無法確保，因此許多專家學者共同研議提出讓語音資料也能透過費用低廉的網路傳輸，但卻能擁有接近電信級的語音品質保證，基於這目標，H.323 正式由 ITU-T 組織提出[1][2]。

H.323 具有嚴謹的通訊標準及幾近於電信標準的可靠度，純粹就商業標準與規格書裡定義的完整性和複雜度來看它是最具優勢的 VoIP 通訊協定。然而正因為如此，所以 H.323 也產生缺乏彈性、可攜性不佳且

H.323 的使用呈現一個陡峭的學習曲線，加上有著很高的實作成本，所以容易造成各家廠商作出的東西會有無法互通的情形。這也造成以 H.323 為基礎的通訊設備難以被社會大眾所接受。這些問題被後來 IETF (Internet Engineering Task Force) 成員 J. Rosenberg 等人所提出討論，經長時間討論後提出一個有別於 H.323 的全新通訊協定 SIP (Session Initiation Protocol) [3]。

SIP 是由 IETF 組織於 1995 年針對網際網路電話所制訂出來的標準，現在最新規格是 RFC3261。SIP 採用類似於 HTTP(Hyper-Text Transmission Protocol 協定的文字式 (Text-based) 通訊協定以方便使用者任意連結被授權的 SIP 伺服器取得服務，SIP 屬於應用階層 (Application Layer) 控制協定[4][5]，適用於建立即時性資料連線及提供多媒體的通訊服務並且支援 Terminal, Session, Personal 及 Service mobility[6]，透過這些行動服務，SIP 可以完整支援使用者對於移動所產生的需求。另外，SIP 採用開放式架構，只要終端裝置可以連結上網，並能正確執行 SIP UA (User Agent) 即可以進行通訊，相對於 H.323，對於時下講求行動的人們來說，不啻是一個相當方便且低廉的通訊工具，也正因為如此，SIP 所帶來的 VoIP 服務深受許多企業及個人喜愛，因此，SIP 被公認為是最具潛力的通訊協定之一。

隨著網際網路電話的興起，其低廉的建置成本及通訊費用，深受企業組織及個人的喜愛，有鑑於此，許多手機製造商看準商機，準備下年度推出結合無線網路的行動手機，屆時人們將可以透過無線網路取得 VoWLAN 服務[7]，隨著使用者人數的增加，各家網路電話服務提供者之間的相互間接甚至和 3G 網路中的 S-CSCF(Serving-Call Session Control Function)連結後所產生的多重註冊機制(Multiple Registration)將會是一大課題。

隨著使用的人數增加，單一網域內的用戶將不再僅僅侷限於區域或號碼的限制，而是可以使用單一號碼任意註冊到最接近的服務提供者，真正做到號碼可攜 (Number Portability) 及多重註冊 (Multiple Registration)，SIP 為滿足上述要求提出三個行動管理机制，如下所示：

1. Personal Mobility：強調個人移動的同時，可以很簡單地被身邊的通話裝置找尋到，為提供這機制，系統必須提供多重註冊服務。
2. Session Mobility：當使用者正在通訊時發生換手 (Handoff) 的情況也可以容許不中斷連線。
3. Terminal Mobility：強調通話裝置移動時若遇上不同的環境時可以很快的恢復連線狀態。事實上，現今通話裝置遭遇不同環境時大多會發生在換手的情況，所以 terminal mobility 有部分成分是因為發生 session mobility。

通訊技術逐漸發達的現今，個人擁有多個通訊裝置是非常普遍的情況，而這些通訊裝置都會有一個共同的特色，那就是使用相同的註冊位置 (Register Name)，當這些位置都同時註冊上線時，人們不再受到通訊裝置的天生極限而被限制活動範圍，舉例來說，若通訊裝置為有線裝置，則使用者必須在附近才能順利接聽，而無線裝置則可隨身攜帶，這項新的網路通訊方式被稱為 Personal Mobility。

為了能夠有效且快速搜尋所有使用者的通訊裝置，RFC-3261 定義兩個基本且實用的演算法供系統業者採用，一為循序搜尋法 (Sequential Search)，二為平行搜尋法 (Pure Parallel Search)¹，第一個演算法可以獲得最低的網路資源浪費但卻可能需耗時較長的時間才能建立連線；而第二種方式連線建立時間較短，卻需要耗費大量網路資源。

假設系統採用平行搜尋演算法來為使用者提供 personal mobility 的服務，則通訊系統會平行、全面搜尋全部已註冊成功的通話裝置，在搜尋之後還需要對回應的訊息加以判斷及轉送給適當位置，可以想像的

若使用者的註冊位置增加，則系統及網路的負擔也將大大地成長，對於系統業者來說這是相當大的負擔，若可以減少系統及網路負擔，系統就可以以相同的規模服務更多的用戶。

本論文主要提出在 SIP 環境之下對個人移動性管理中的網路資源浪費的有效解決方法，我們提出一個折衷的方案來解決網路資源浪費的問題，在平行搜尋的概念之下系統要對一個使用者發出為數不少的訊息，假設系統可以提前瞭解使用者可能的位置，則系統將節省一些不必要的浪費也可以達到快速搜尋的目的。

其他章節組織如下：第 2 節說明 SIP 基本概念。第 3 節說明 Pipelined Search 主要定義及優缺點。第 4 節說明實驗部分，我們將理論實作在 SER 系統中，並測量相關參數及分析。第 5 節提供結論及未來後續研究方向。

2. Session Initiation Protocol (SIP)

2.1 SIP

SIP 採用開放式架構，只要終端裝置可以連結上網，並能正確執行 SIP UA (User Agent) 即可以進行通訊，相對於 H.323，對於時下講求行動的人們來說，不啻是一個相當方便且低廉的通訊工具，因此，SIP 被公認為是最具潛力的通訊協定之一 [8][9]。

其主要工作可分為下列四種：

1. 啟始通話請求
 - 找尋被呼叫端 (Callee) 的正確位置
 - 定義通話資源要求
2. 加入通話
3. 修改通話請求內容
4. 終止通話請求

SIP 網路主要由下列四個元件組成：

1. Proxy Server：是 SIP 網路的核心，提供邏輯上的服務，例如：來電轉接、語音信箱...等。
2. Redirect Server：負責儲存、回應 UAS 的現在位置。

¹ 本文中提及的 Pure Parallel Search 即是 RFC-3261 中的 Parallel Search，為有效區隔差異，故增加 Pure 字眼。

3. Location Server：提供使用者認證及移動管理。
4. User Agent：可分為 UAS（User Agent Server）和 UAC(User Agent Cleant)，負責起始通話請求，語音轉換及終止通話。其中 UAS 為被呼叫端，UAC 為呼叫端，UAS 並不一定是使用者的終端裝置，也可能是語音信箱伺服器。

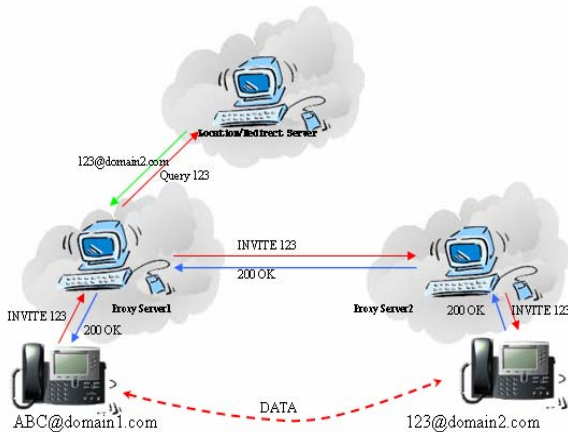


圖 2.1 SIP Call Setup

圖 2.1 說明使用 SIP 協定去啟始通話的流程，本圖簡化通話建立流程將 Location Server 跟 Redirect Server 放置在一起並且在 Proxy 中採用 Stateless 機制，根據 Iptel 組織所建議，這三種伺服器（Proxy、Location、Redirect）可以有效地整合在一起，本圖為有效說明，所以還是將其分開。

當 ABC 裝置想要撥電話給 123 裝置時，ABC 會先發送 INVITE 訊息給他的 Proxy Server，如下所示：

```
INVITE sip:123@domain2.com SIP/2.0
From: <sip:abc@domain1.com>;tag=d95c305
To: <sip:123@domain2.com>
Call-ID:7504b7df@192.168.1.15
CSeq:1 INVITE
```

當 Proxy Server 接收到訊息時，會先去詢問 Redirect 目的端真正位置，Proxy Server 再依據 Redirect 的回應內容傳送給目的端的 Proxy，若相關訊息資訊都正確且資源都可以順利保留，裝置 123 會回應 200 OK 的訊息給裝置 ABC，此時就會建立起連線。

3. Pipelined Search

眾所皆知，在網際網路上進行相關服務的取得都

需要進行雙方確認的動作，以建立收、送雙方都認可的連線方式，再以 SIP 為基礎的 VoIP 架構，更是重要，因為雙方都需要確認採用的語音編碼、對方實際位置及所能提供的服務為何…等，因此會產生許多信令。當系統採用 Stateless Proxy 時，UA 僅僅會將 INVITE、Register、BYE…等信令送至系統進行簡易轉送，其餘信令皆不要求經過。但若採用 Stateful Forking Proxy 時，則系統會要求 UA 將所有信令都需送至 Forking Proxy 作轉送處理，相較於 Stateless Proxy，Stateful Forking Proxy 將會處理相當大量的轉送及查詢動作。

在 RFC-3261 SIP 規範書中提及，若系統採用 Stateful Forking Proxy 時，有兩個 Forking 的策略可以運用，一為 Sequential Search，二為 Pure Parallel Search。Sequential Search(SS)會依據使用者 UA 的註冊時間先後順序（FIFO）發送會議起始(INVITE)信令，若該位置沒有任何回應則會在發送下一個位置，直到搜尋到最後一個註冊位置，這個演算法雖然可以降低系統負擔，但搜尋時間冗長，對於呼叫方來說會造成相當的不便。而 Pure Parallel Search 則是不論先後註冊順序，將全部的位置同時發送出去，搜尋時間短，卻也犧牲了網路資源。

根據上述兩個演算法的討論而提出一個折衷的演算法，這演算法主要目標是為降低網路負擔及提高搜尋效率，這演算法就是 Pipelined Search。相關說明將在 3.2 節討論。

3.1 Pure Parallel Search

在網際網路日益盛行的風氣之下，低價的通訊費用及高速傳輸資料使得未來人們同時擁有多多的通話裝置將是可以預期的，假設某個使用者日常在辦公室使用桌上型話機，外出攜帶智慧型手機而下班後希望將通話轉至家裡，那麼該使用者就有可能同時註冊許多位置，每當該使用者被呼叫時，根據之前所述系統就必須同時轉送信令給該使用者所有的位置，這就是 Pure Parallel Search，對於用戶少的系統而言或許輕而易舉，但對於用戶數量龐大的系統而言，將會是一大負擔。

3.2 Pipelined Search

有鑑於此，我們可以改變些許作法將這效能上的瓶頸加以改善。假設一個使用者已經使用該系統所提供的 VoIP 服務已經些許日子，而且使用者的生活作息固定，則系統可以觀察該使用者的相關習性定義出具有優先順序的通話裝置，高優先裝置意味著使用者在這裝置旁邊的機率是非常高的，若可以將預計要發出的啟始信令依據通話裝置的優先順序作優先權排序且高優先權的群體(Group)和次一優先權的群體之間延遲一段時間發送。

在實作中，優先順序可以根據 RFC-3261 中所預先定義的 qvalue 求得，另外我們也定義 dvalue 來定義應該要延遲時間的長度，dvalue 可以是固定的時間值，也可以依據上一群體和次一群體之間的 qvalue 之差的加權值求得，這演算法可以被稱為『Pipelined Search』，本程式實作時因願慮簡單化，故採用固定值，在實際情況，應為變動值會有最佳效能表現。群組化示意圖如圖 3.1 所示。

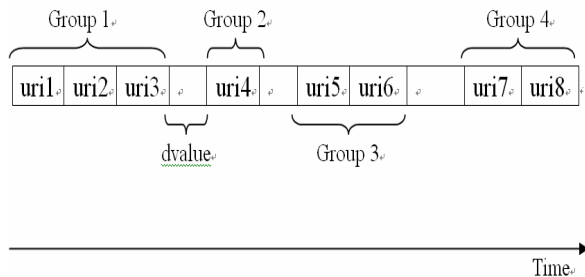


圖 3.1: The group in the pipelined

圖 3.1 中可以瞭解，假設當 Registrar 伺服器回傳排序後 Callee 的所有位置及 dvalue 給 forking proxy，forking proxy 會根據不同位置之間 dvalue 的差瞭解哪些相鄰的位置是同一群組，可以平行處理，而哪些為次優先群組可以延遲處理，這就是群組化的概念，透過群組化及優先權的概念，系統可以有效率地搜尋，以減少不必要的搜尋成本支出。

我們將以下列數個步驟來表示 pipelined search 演算法的運作流程，相關圖示請參照圖 3.2：

- I. forking proxy 收到 caller 的 INVITE 信令要求代為轉送給 callee。
- II. forking proxy 向 Registrar 伺服器查詢 callee 的正確位置。

- III. Registrar 伺服器搜尋出 callee 擁有多個位置及每個位置的 qvalue。
- IV. Registrar 依據 qvalue 的大小依序將所有 callee 位置排序。
- V. Registrar 伺服器透過簡單的數學公式將 qvalue 轉換成 dvalue。
- VI. Registrar 伺服器將排序後的位置及 dvalue 傳送給 forking proxy。
- VII. forking proxy 依據 dvalue 值產生群組，並分群、延遲送出給適當的位置。

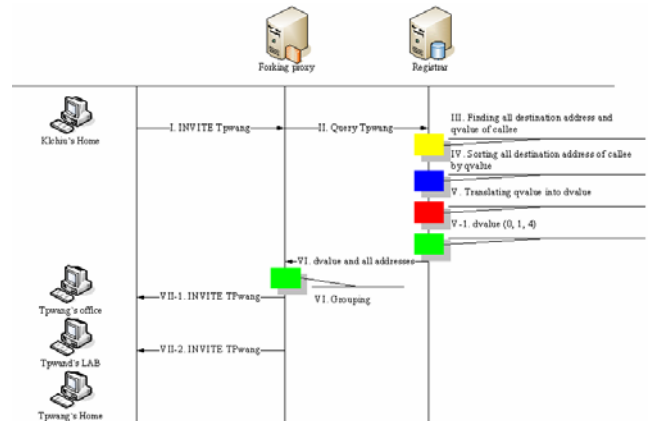


圖 3.2: Pipelined Search 演算法示意圖

根據以上描述，pipeline search 在最佳的情況中，系統呼叫第一個群體時就完成任務，則系統只需發送少數幾個信令就可以完成，而在最糟的情況中，系統則需呼叫到最後一個群體，這也僅僅花費和平行搜尋相當的成本。

這樣的作法使得系統可以服務的用戶大大增加，假設系統每秒可以服務 150 通通話請求，每通通話請求又需平行處理 10 通通話轉送服務，而每通轉送的通話請求又需後續處理暫時的(Provisional)及最後的(Final)確認，則系統約略需每秒處理 6600 個信令，這是相當大的負擔，若採用新作法，在一般的狀況下，則只需處理約略一半的信令。進一步說明。如圖 3.3，若 KLchiu 欲撥電話給 TPwang 並且發出起始通話請求 INVITE 給 Forking Proxy，這時 ForkingProxy 就會在資料庫中找尋 TPwang 所註冊的資訊，並依據這些註冊訊息發出 INVITE 信令給所有註冊的位置，這些位置收到信令後會主動回應 180 Ringing 信令回來告知已經完成相關設定且已經開始震鈴通知使用者接電話，但這當中僅僅只有一個通訊裝置可以被順利接通，假若 TPwang 當時正在實驗室，則 LAB 中的通訊裝置則會

再回應 200 OK 信令通知 Forking proxy 及 KLchui 完成通話起始流程,其他 TPwang 的通訊裝置則會被自動切斷。

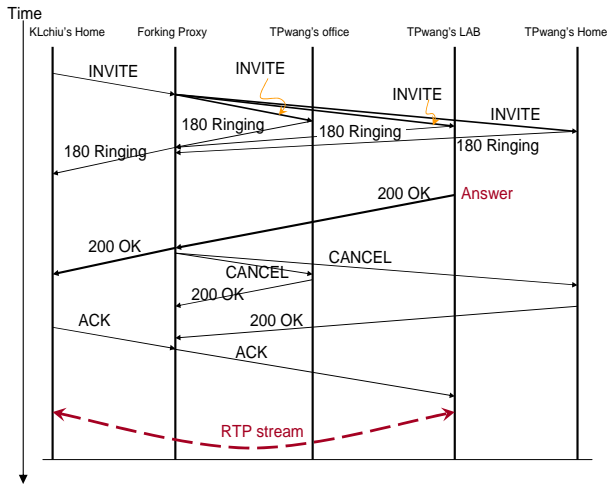


圖 3.3: Timing diagram for parallel search

不論如何,系統採用 Pipelined Search 時,由於系統會延遲 d 時間長度去等待該群組中的通話裝置被接起話筒,當使用者正在優先權較高的通話裝置旁邊時,系統僅僅需要發送少許的信令即可。如圖 3.4 所示,TPwang 正在實驗室辦公,若有 KLchui 恰巧撥網路電話給 TPwang,則系統則會在第二順位址找尋到 TPwang,並且不需要再發送第三個通話啟始信令給 TPwang 位於住家通話裝置。基於上面所描述,採用 Pipelined Search 的系統比起採用 Pure Parallel Search 的系統可以節省四個信令處理成本,若使用者註冊通話裝置越多,則可以節省信令處理成本將會更加成長。

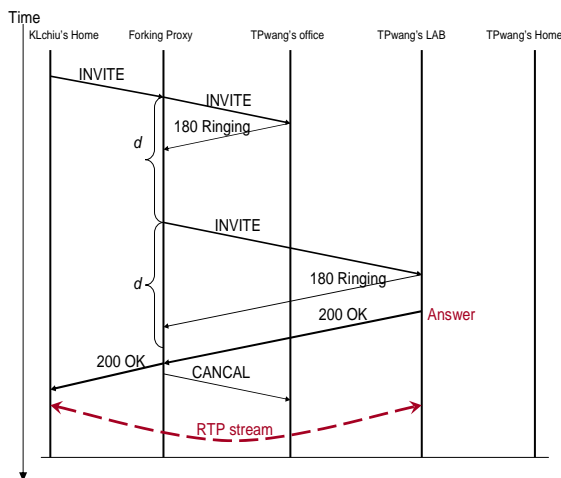


圖 3.4: Timing diagram for pipelined search

4. 實驗結果

我們將相關概念及想法撰寫在 iptel 組織所發表的 SER 0.8.14 版本中[10-19],並測試、比較在 Pipelined Search 中兩種常見的分配及原先的 Pure Parallel Search 之間的傳輸成本和通話建立延遲時間上的差異。

4.1 實驗架構

在 SER 原先版本中,已經定義最大 Fork 數目為 11,因此本實驗採用最大被呼叫數目為 9 個,如圖 4.1 所示,該架構中所有的 UAS 都僅僅在同一個或跨過一個交換機,並和 Forking Proxy 及 UAC 在同一網域下。

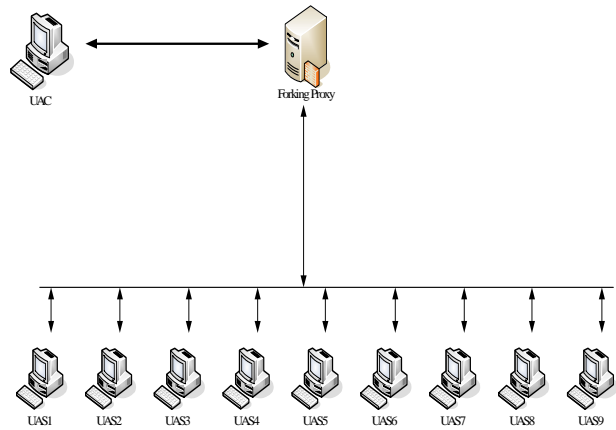


圖 4.1: 系統實驗架構圖

UAC 採用依據 SIP Stone 所開發的 SIPp 效能測試軟體,本實驗中並不使用該程式測試效能,僅藉由 SIPp 發送 SIP 相關信令。UAS 的部分採用 xten 所開發的 UA 程式。

本實驗,總共分為三階段實驗:

- 第一階段設定 UAS 數量在 3 個,Uniform 的機率為(1/3, 1/3, 1/3),Locality 機率為(0.57, 0.285, 0.145)。
- 第二階段設定數量為 6,Uniform 機率為(1/6, 1/6, 1/6, 1/6, 1/6, 1/6),Locality 機率為(0.51, 0.255, 0.125, 0.065, 0.03, 0.015)。
- 第三階段設定數量為 9,Uniform 機率為(1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9),Locality 機率為(0.5, 0.25, 0.125,

0.065, 0.03, 0.015, 0.01, 0.005, 0)。

為了減輕 UAS 的處理負擔，避免因流量過大影響測試的準確性，SIPp 中的 Call rate 參數被設定為六秒，單位週期內處理腳本數量限制為一個，且在發送 ACK 及 BYE 信令後延遲 3000 milliseconds。

在 forking proxy 部分，實驗採用 ethereal-0.9.3 負責擷取 SIP 所有的封包，並在後端使用交通大學資工系所研發的 SIPv6 封包分析軟體來進行計數及過濾的工作。每一階段實驗都進行 200 次。

4.1.1 範例 第一階段

在 Personal Mobility 的環境之下，使用者可以有許多個裝置，但在同一時間，僅僅只會有一個地方會有使用者接起電話的情形發生，為了模擬使用者的移動特性，我們分別假設使用者在第一個位置的機率、第二個位置的機率……，經由總呼叫次數的換算可以求得該位置應該被呼叫次數，並設定該位置可以自動接聽電話，以完成全自動實驗為目標。

在最佳情況 (Best Case) 下，系統僅需要搜尋第一個位置即可以找尋到使用者，因此系統可以不必在發出下個位置的 INVITE 信令，而第一個位置的相關信令交換就會如同只有一個位置註冊時一樣，最後 SIPp 會發出 BYE 信令，來完成一次實驗，在圖 4.2 中，forking proxy 總共需要十四筆信令。

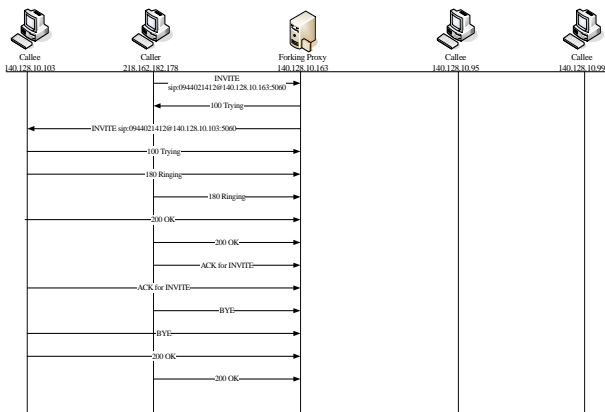


圖 4.2：範例 實驗劇本-在第一個位置接聽

較為常見的情況是系統需要發送到第二個位置時才能找到使用者所在，因此系統需要對另位置進行取

消命令的動作，這時候 UAS 會回應 487 Request Terminated 的訊息以表示正在取消命令回收資源，當相關資源回收完畢就會 200 OK 的信令給系統，完成整個動作。整個過程約略需要 22 個信令交換。而在最差情況下，forking proxy 需要搜尋到最後一個位置，而發出的總信令數將會和 pure parallel search 一樣約略二十九個。

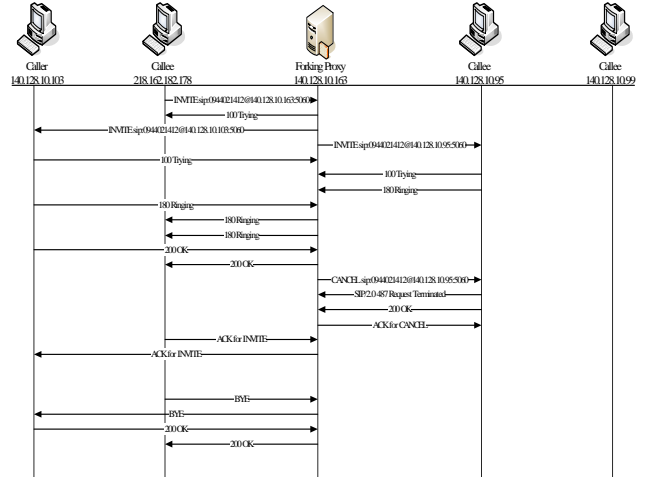


圖 4.3：範例 實驗劇本-在第二個位置接聽

4.2 實驗結果

4.2.1 搜尋成本

經由三個階段，每階段 200 次實驗後，可以計算出 forking proxy 總共送出多少信令，圖表請參照圖 4.4。

圖 4.4 中，Uniform 及 Locality 在 N=3 的時候，成本幾乎一樣，主要是因為第一順位 UAS 及第二順位 UAS 之間 qvalue 差異極小，所以被歸為同一群體，而與第三順位 UAS 的差異也不大，所以 dvalue 值也不會相差很大，且 x-lite 程式本身需要取得資源後才會接聽電話，這過程約略需要 2 秒的時間，所以在大多數的實驗中，forking proxy 會發送前三個位置後才會找到使用者，因此 N=3 時，Uniform 及 Locality 差異不大的原因在此。

N=5 時，Locality 所花費的成本逐漸趨於平緩，但仍舊有微幅成長，與 4.2 節中圖 4.6 比較可以發現理論數據顯示 N=7 的時候成本將不再成長，而實際測試

中，卻會小幅成長，推測應為網路傳輸快慢而造成 forking proxy 偶而會發出不必要的 INVITE 信令，尤以 N=3 到 N=6 之間最為嚴重。pure parallel search 則如同預期一般穩定成長。

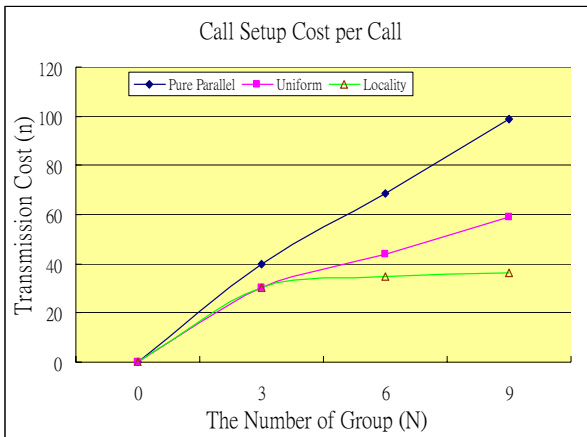


圖 4.4：通話建立成本曲線圖

在通話建立延遲時間方面，由於在實驗室裡測量出來的 RT 值都非常小，所以難以研究 RT 與 dvalue 之間有什麼關係上的變化，所以實驗僅只能就總建立延遲時間來作分析。

圖 4.5 中可以看出 Locality 不論群組數是多少，對於建立通話的延遲時間都不會有很大的變化，主要原因是有很大的機率通話會在前幾個群組時就已經被成功建立。而 Uniform 則會因為系統無法預先猜測使用者在那個位置(因為被搜尋機率相同)，而會類似於循序方式在搜尋使用者，所以時間上會呈線性成長。

Pure parallel search 則不會有任何時間上的變化，因為他是平行處理所有位置。

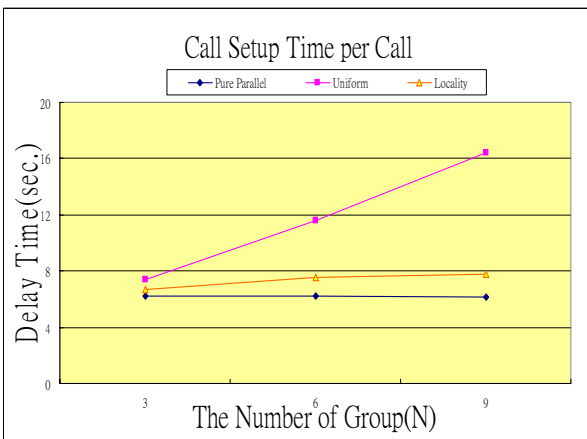


圖 4.5：通話建立延遲時間曲線圖

5. 結論與未來工作

本論文提出在 personal mobility 環境下一個解決網路資源大量被消耗的解決方案，在傳統方法之中若要有好的搜尋速度就必須犧牲網路資源，若要保留網路資源則要犧牲搜尋速度，若有折衷方案，將可以依據當時狀況選擇最佳參數。藉由優先權概念的引進及群組化搜尋目標，系統可以很快瞭解最有可能的位置是哪些，進而增加搜尋到使用者的機率。

實驗結果顯示 pipelined search 方法在網路資源耗用方面可以比 pure parallel search 更加節省網路資源的浪費，在延遲通話建立時間方面也可以比 sequential search 縮短許多，尤其在 Locality 分配狀態之下，效能表現會更加明顯。

SIP-based VoIP 逐漸風行的現日，personal mobility 的問題將會是服務供應商的一大課題，在營運成本及便利性服務的考量下，本論文提供一個折衷的方式，在網路搜尋成本及搜尋效率上取得一個平衡點，藉由 pipelined search 機制，系統不僅可以大幅降低搜尋成本，可以提供更好的系統服務。而當系統提供服務一段時間後，使用者的移動特性將會趨於 Locality 分配，藉由實驗證明，當這分配產生時，不論使用者的通訊裝置數量為何，對於服務供應商而言，都不會增加太多成本的支出，真正做到 personal mobility 的境界。

未來我們將持續研究下列相關問題：

1. qvalue 的給定問題：qvalue 的產生和使用者的移動特性有關，能夠正確掌握使用者移動特性才能精確控制搜尋成本，但在許多情況，使用者的移動特性並不會非常穩定。
2. 本論文目的在於研究如何減少網路資源消耗的問題，因此並未對現有 refer、3PCC 等的機制作整合測試，未來後續研究者可以進一步研究如何做到整合運用的研究。

致謝

This work was supported by NSC, Taiwan, under grant numbers NSC 93-2219-E-126-001 and NSC 94-2213-E-126-006.

參考文獻

- [1] ITU-T Recommendation H.323, “Packet-based multimedia communication systems”, 1998.
- [2] B. Vlaovic, Z. Brezocnik, “Packet based telephony”, *International Conference on EUROCON'2001, Trends in Communications*, Vol.1, pp.210–213, July 2001.
- [3] K.K. Tam , and H.L. Goh, “Session Initiation Protocol,” *IEEE International Conference on Industrial Technology*, Vol. 2, pp. 1310-1314, Dec 2002.
- [4] J. Rosenberg *et al.*, “SIP: Session Initiation Protocol,” *RFC 3261, IETF*, June 2002.
- [5] SIP Tutorial at <http://www.iptel.org/sip/> (last visit 2 September 2004)
- [6] H. Schulzrinne, E. Wedlund, “Application layer mobility using SIP,” *Mobile Computing and Communications Review*, Volume 4, Number 3.
- [7] E. Wedlund, H. Schulzrinne, “Mobility support using SIP”, *2nd ACM/IEEE International Conference on Wireless and Mobile Multimedia*, Seattle, Washington, Aug. 1999.
- [8] Henry Sinnreich, Alan B. Johnston, “Internet Communications Using SIP”, John Wiley & Sons, Inc., New York, NY, USA.
- [9] Jyh-Cheng Chen and Tao Zhang, “IP-based next-generation wireless networks: systems, architectures, and protocols”, John Wiley, New York, 2004.
- [10] SER Product Sheet, at <http://www.iptel.org/ser/doc/ser.pdf>
- [11] <http://www.iptel.org/>
- [12] Ser-0.8.14, at <ftp://ftp.berlios.de/pub/ser/latest/>
- [13] Jon Peterson, etc.” Mapping of ISUP parameters to SIP headers in SIP-T,” draft-jfp-sip-isup-header-00
- [14] G. Camarillo, Ericsson, A. B. Roach “Integrated Service Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping” , RFC 3398, November 2002.
- [15] Schulzrinne, Columbia U,” Mapping between ISUP and SIP” INTERNET-DRAFT, January 1999.
- [16] Jiri Kuthan,” SIP and PSTN Connectivity” <http://www.ripe.net/ripe/meetings/ripe-46/presentations/ripe46-eof-enum-sip-pstn.pdf>, September 2003.
- [17] Dorgham Sisalem, Jiri Kuthan,”Understanding SIP”, <http://www.iptel.org/sip/>
- [18] Yuan Zhang, SIP-based VoIP network and its interworking with the PSTN, *Electronics & Communication Engineering Journal* , Volume: 14 , Issue: 6, pp.273 – 282 , Dec. 2002.
- [19] Cisco System inc.,”Cisco 2621 Modular Access Router Security Policy”, Cisco System inc., http://www.cisco.com/en/US/products/hw/routers/ps259/products_user_guide09186a00800a9604.html