

Session mobility over multiple devices using J2ME

莊詠程

中華大學資訊工程學系研
究生

Hm09602010@chu.edu.tw

陳旻秀

國立東華大學資訊工程學
系

mxchenH@mail.ndhu.edu.t

翁文彥

中華大學資訊工程學系

wenwen.weng@chu.edu.tw

w

摘要

Session Initiation Protocol(SIP)靠著簡單且可擴充的特性，已被 VoIP 與 3GPP 所採用為通訊協定，並且也被廣泛的使用在網際網路與電信領域中。為了讓使用者能夠透過各式各樣的裝置在通話過程中自由的轉移通訊設備以及將連線服務進行分流或是取回，本篇論文即以 SIP 當中的 session mobility 議題為主軸，將連線服務進行分流或是取回的機制在 J2ME 上進行實作。

關鍵字 : SIP , session mobility , J2ME

ABSTRACT

According to the capabilities of ease use and extensibility mechanisms, Session Initiation Protocol(SIP) has been adopted by VoIP and 3GPP for signaling protocol, and also be widely used in the signaling protocol of Internet and telecommunications for multimedia services. Session mobility is the key issue of mobility management of the SIP and it is used to transfer an ongoing media session from one device to another. In the paper, we propose the mechanism of session mobility to easily transfer, split and

retrieve the ongoing service over multiple devices. We also implement our prototype system using J2ME.

Keywords : SIP , session mobility , J2ME

1. Introduction

由於資料串流服務在日常生活中越來越普及，像是 VoIP 或是線上影音服務等等，但是要如何達到可讓使用者簡單且方便的控制資料串流服務成了相當重要的議題。因此，[1]就被研究人員提出來。SIP 是用來管理連線服務的通訊協定，透過 SIP 使用者可以建立、修改或是終止連線服務。此外，使用者也可以利用 SIP 選擇所感興趣的連線對象，透過特定的控制訊息要求與對方建立連線服務、修改目前正在進行的連線服務或是中斷目前與對方的連線服務。根據 SIP 的特性可應用在 terminal mobility、personal mobility、service mobility 與 session mobility 等四大範疇。

Terminal mobility 相似於 Mobile IP 的功能。透過 SIP 的架構，終端機可在不同的 IP subnet 之間移動，且維持目前正在進行的連線服務。Personal mobility 具有讓 SIP 使用者在任意終端機環境下利用 SIP URI 對不同對象或族群進行存取的能力。在存取過程中，不同的裝置可共用一個

SIP URI 或是一個裝置可同時使用多個 SIP URI。Service mobility 允許使用者切換到不同裝置或是存取到不同的網路後，只要根據個人的偏好設定，即可持續使用連線的服務，免除了繁瑣的設定程序。Session mobility 則討論即使使用者任意轉移連線裝置後，無須中斷連線仍然可以維持目前的連線服務。

因為單純建立與終止連線服務已經無法滿足使用者的需求，使用者想要的是能夠自由的將連線服務轉移、分流並整合到其他裝置上。所以本篇論文將以 session mobility 為發展重點。一般來說行動裝置雖然可以方便的與其他使用者建立連線服務，但是受限於較小的體積，只能透過較差的螢幕解析度與音質建立連線服務。假設目前環境中有較高階的視訊以及音訊設備，使用者以往都必須中斷目前的連線服務來使用較高階的影音設備，此過程相當的麻煩且不方便。鑒於重新建立連線服務的瑣碎程序，有人提出了將目前正在進行的連線服務自由轉移到其他裝置的機制。

然而這樣的轉移機制是有些問題存在的，因為連線服務轉移的機制會將連線服務全部轉移到另外的裝置上。舉例來說當使用者正透過音訊與視訊裝置與外界進行通訊，則目前的連線服務就是建立在音訊與視訊兩個連線上。假設目前環境中只有音訊的裝置，此連線轉移的機制便會在視訊方面出現連線的問題。為了解決連線轉移機制可以滿足連線設備的問題，後來我們提出將連線服務進行分流[2][3]。透過此種機制，使用者即可轉移全部的連線服務或是根據不同的設備進行部分連線服務的分流。

本篇論文是以[2][3]為基礎進而將研究成果移植至手持行動裝置上，並實作出

可根據不同的情境需求進行連線服務轉移的系統架構。在接下來的論文中，第二節將討論相關研究背景，第三節將描述關於連線轉移機制的類別，第四節將介紹系統架構、第五節是實作的成果與分析並且於第六節作結論。

2. Related work

為了要能夠讓連線服務能夠在任意兩個user agent中進行轉移，RFC 3515[4]與 RFC 3725[5]分別定義了REFER與Third-party call control這兩種機制，其轉移方式如圖 1 與圖 2。

利用 REFER 連線服務轉移機制，藉由 MN1 對 MN2 發送 REFER message，要求 MN2 對 CN 發送一個 INVITE message，在收到 MN2 所發送的 INVITE message 後 CN 即對 MN1 發送 BYE message 來終止原本建立在 MN1 與 CN 之間的連線，並與 MN2 建立新的連線服務。

若是使用 3PCC 的連線服務轉移機制，首先 MN1 發送不包含 SDP 的 INVITE message 給 MN2，MN2 則回傳本身的 SDP 給 MN1。利用 MN2 所回傳的 SDP，MN1 將其包入 INVITE message 並發送至 CN，收到來自 MN1 的 INVITE message 後，CN 也回傳本身的 SDP 給 MN1。接著 MN1 對 MN2 發送包含 CN 的 SDP 的 ACK message 給 MN2，並對 CN 發送 ACE message。如此一來 CN 與 MN2 接擁有對方的 SDP 資訊，透過此機制可將原本建立於 MN1 與 CN 之間的連線服務轉移至 MN2 與 CN。

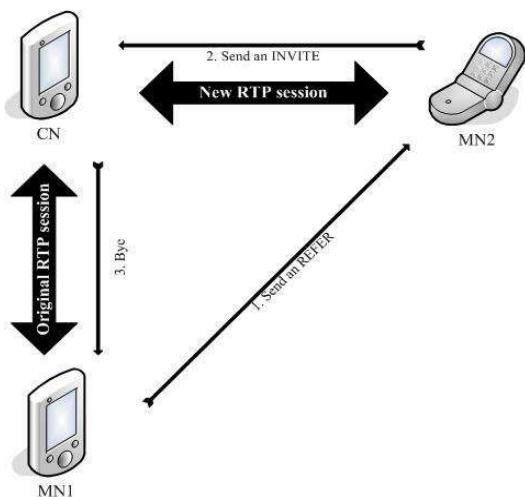


圖 1 REFER 連線服務轉移

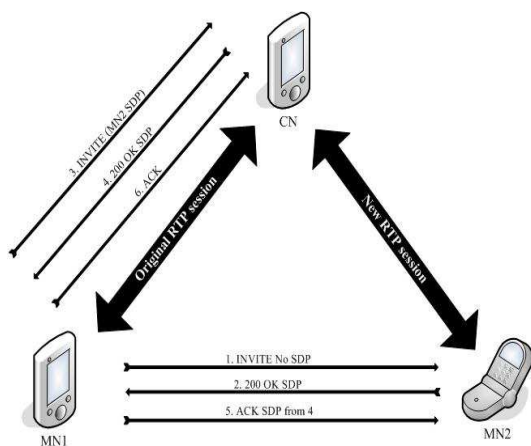


圖 2 3PCC 連線服務轉移

另外當兩台裝置無法直接通訊時，可利用由 RFC 3892[6] 所定義的 Nested REFER 請求其他的裝置轉送 REFER message 來進行連線服務轉移。原始的 REFER message 把要 REFER 裝置的 SIP URI 填入 Refer-To 欄位內，而 Nested REFER 則對 Refer-To 欄位做了修改，以下為 Refer-To 的內容：**Refer-To : < sip : C ; method=REFER ? Refer-To = " < sip : D > "** 如圖 3 所示：裝置 A 對裝置 B 發送 Nested REFER，裝置 B 可從此 Nested REFER 的 Refer-To 欄位中得知要對裝置 C 發送 REFER message，此 message 的 Refer-To 欄

位為裝置 D 的 SIP URI。裝置 C 在接收到裝置 B 所發送的 REFER message 之後便使用一般的 REFER 流程，將正在進行的連線服務進行轉移。

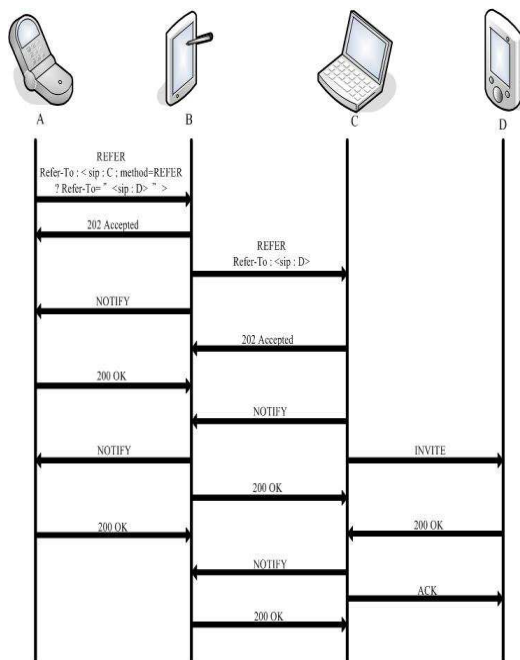


圖 3 Nested Refer 流程

3. Category of Split session

隨著家用網路技術的快速成長，像是 jini，upnp 以及 OSGi[7] 能讓家中的一些電器產品具備網路能力，因此我們能夠輕易的將正在進行的連線服務轉移到其他的裝置上。

連線服務自由移轉與整合包含服務轉移與連線取回兩種機制。簡單來說服務轉移就是將正在進行的連線服務由目前所使用的裝置轉移到其他的裝置上；而連線取回則是從其他裝置轉移控制權回到原來所使用的裝置。在進行連線服務轉移的時候分為全部連線服務轉移以及部分連線服務轉移兩種。使用者能夠透過全部連線服務轉移機制將連線服務直接轉移到另一個裝置上，但是此種機制只提供將

連線服務全部轉移，若原來使用者正在進行視訊與音訊的通訊，則此連線服務包含了影像與聲音兩種部分連線服務。當使用者進入一個只有音訊裝置的環境時，全部連線服務轉移機制則會出問題。因此[3]提出了部分連線服務轉移機制來解決此問題。透過分流將連線服務轉移到各個不同的裝置上。因此，我們可以根據不同的情境對目前環境中的裝置進行連線服務的轉移或是進行分流。接下來我們將介紹在連線服務轉移機制的四種情境。

a. Split all sessions

全部連線服務轉移機制指的是將目前正在進行的連線服務完全地轉移到其他的裝置上，經由轉移之後在原本的裝置就不會再有連線服務的進行。我們可從圖 4 中得知一開始使用者 Joe 透過行動裝置與遠端用戶 Manny 建立視訊對話。當 Joe 回到辦公室後，即將未結束的連線服務轉移到 Joe 辦公室內的影音設備：螢幕、攝影機、麥克風以及喇叭。Joe 藉由將目前正在進行的視訊對話轉移到其他的影音設備後即可以較佳的影像和音質繼續與遠端 Manny 繼續進行視訊對話。

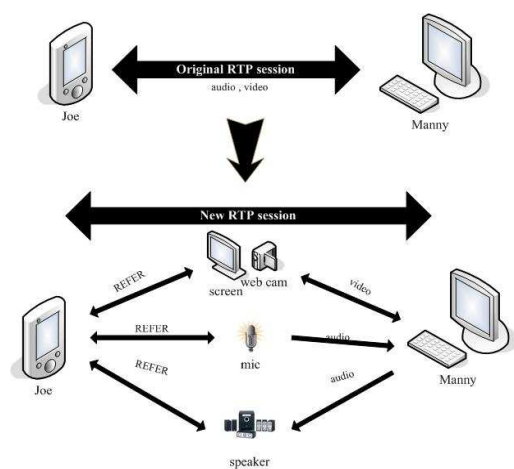


圖 4 全部連線服務轉移

b. Split partial sessions

部分連線服務轉移機制指的是在使用者的裝置保留某個或是某些多媒體連線服務並將剩下的多媒體連線轉移到其他的裝置上。我們可從圖 5 中得知使用者 Joe 利用行動裝置和遠端用戶 Manny 進行視訊會議。當 Joe 回到家後想將尚未結束的連線服務轉移到品質較佳的影音裝置上，因此 Joe 將影像的部分轉移到家中的高畫質電視與攝影機。透過部分連線服務轉移機制 Joe 即可以電視、攝影機和原本的行動裝置與遠端用戶 Manny 繼續通話。

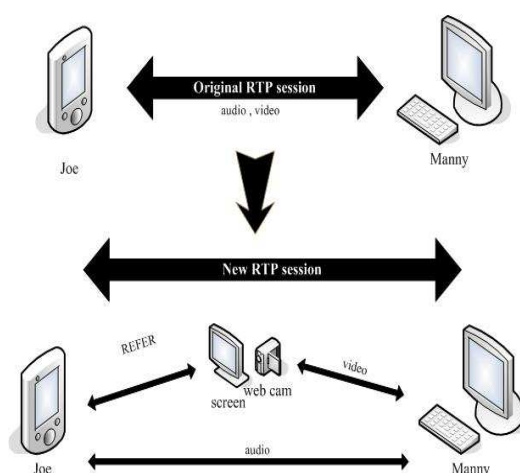


圖 5 部分連線服務轉移

c. Add new sessions

新增連線服務是在原本的連線服務中加入原本不存在的多媒體服務。假設對方用戶不具備其他類型的多媒體裝置，那就不會改變原本的連線服務。否則可以將新的多媒體裝置加入原始的連線服務中。從圖 6 中可觀察到一開始 Joe 利用行動裝置與 Manny 建立語音連線服務，當 Joe 從室外回到辦公室後，即將螢幕與攝影機加入目前的語音通話中，若是 Manny 也有對

等的裝置，Joe 就可以透過新加入的多媒體裝置與 Manny 建立視訊對話。

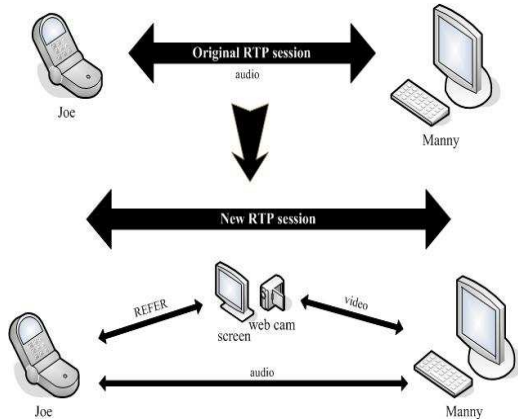


圖 6 新增連線服務

d. Retrieve sessions

透過全部連線服務轉移或是部分連線服務轉移到其他裝置上的連線服務，因為環境的改變，導致使用者需要將正在進行的連線服務轉回至原本的裝置上稱為連線服務取回。由圖 7 可得知原來在室外，進入辦公室後利用高品質的影音裝置與遠端用戶 Manny 延續視訊對話的使用者 Joe，可能因為某些原因要離開辦公室到其他地方。此時 Joe 利用連線服務取回機制即可將視訊對話由個別的影音裝置轉移回其行動裝置，如此一來當 Joe 離開辦公室後還能與 Manny 進行視訊對話。

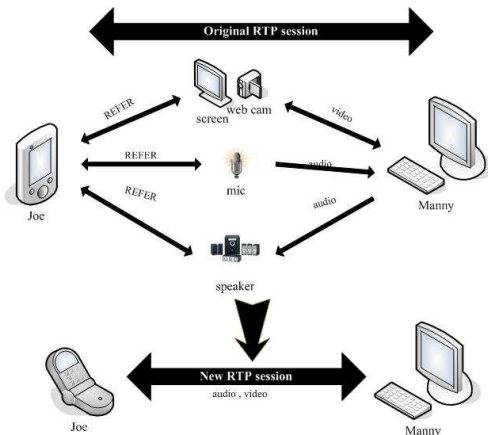


圖 7 連線服務取回

4. System architecture

4.1. Session manager, session user and free node

為了要能夠在不同的裝置上進行連線服務轉移，我們首先要詮釋三種新的 user agent，分別是 session user，session manager 以及 free node。在我們定義與描述這三種邏輯成員的功能之前，要先介紹部分連線服務的概念。也就是說其中一個 user agent 擁有一個或多個多媒體連線服務，其他的連線服務則在不同的 user agent 上進行。在圖 8 中有三個 user agent，分別是 A，B 與 C。假設在 A 與 C 之間建立了聲音與影像的連線服務，接著 A 將聲音連線服務轉移至 B。如此一來 A 與 B 都擁有一部分連線服務。

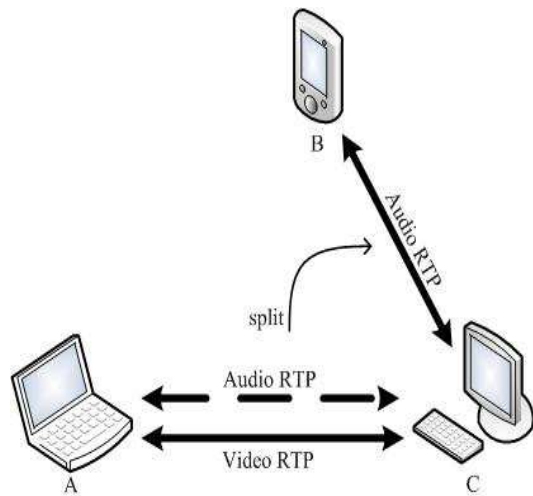


圖 8 部分連線服務示意圖

不具有部分連線服務的 user agent 我們稱為 free node，至少具備一個部分連線服務的 user agent 則稱為 session manager 或是 session user。session manager 與 session user 的差別在於 session manager 擁有所有的部分連線服務的資訊，像是設定、

認證或是授權等等，而 session user 只具備本身的部分連線服務。藉由所有的部分連線服務資訊，session manager 就可以對其他的 user agent 進行轉移、分流、取回或是終止部分連線服務。session user 因為只具備本身的部分連線服務，所以只能對 session manager 轉移部分連線服務或是被動的接受 session manager 的連線服務轉移或取回。透過這三種 user agent，我們就能夠在不同的裝置上進行連線服務轉移。

4.2. Methods and call flow

為了要將[3]順利移植至J2ME的環境中，我們使用了J2ME Wireless Toolkit 2.5.2 所提供的範例程式碼進行改寫。原始的程式碼只能夠對某個SIP URI發送文字訊息，並不具備與其他user agent進行連線的能力。經過修改後，程式能夠由使用者輸入想要進行連線的SIP URI並且建立連線服務。另外，為了要能夠讓連線服務自由的在不同的裝置間進行轉移，我們新增了inviteReferTo、notifyREFER、refer、receiveRTPAudio 與 receiveRTPVideo 等方法。

系統操作流程如圖 9 所示，當使用者對 SIP server 完成登入動作後，狀態即改為 Registered，此時使用者可接受其他 user agent 的連線服務要求、或是接受 session manager 的連線服務轉移，以及主動對其他 user agent 要求建立連線服務。

a. Call In

當接收到其他 user agent 的連線服務邀請時，使用者可以決定是否建立連線服務。當連線服務建立之後，使用者可將此連線轉移至其他的裝置上繼續進行。

b. REFER

當接收到來自其他 user agent 的 REFER request 後，完成登入動作的使用者可以決定是否要與其他 user agent 建立分流後的部分連線服務。部分連線服務建立之後，使用者可根據原 user agent 的 Notify REFER request 將部分連線服務轉回原 user agent。

c. Call Out

完成登入動作的使用者也可以主動對其他的 user agent 發出建立連線服務的要求。並且可在環境改變時將連線服務轉移至其他的 user agent 上繼續進行。

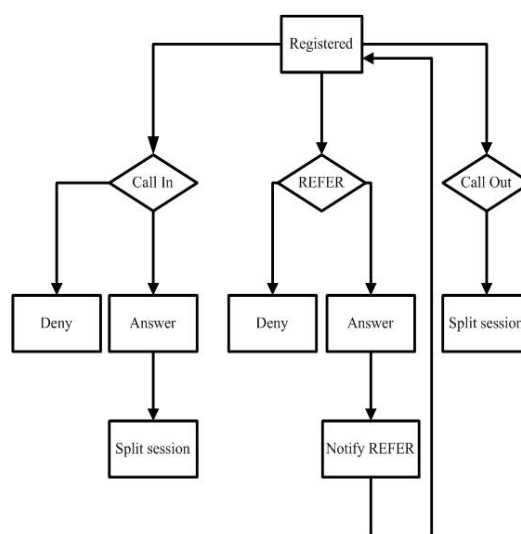


圖 9 System call flow

5. Implementation and evaluation

5.1. Environment overview

透過SIP，使用者能夠輕易地選擇所感興趣的連線對象，並可使用一些特定的控制訊息來與其他使用者建立、修改或是

結束連線服務，也因為SIP的相關熱門技術，許多的SIP project [15]與user agent [2]慢慢的被開發出來。

在 SIP 的開發過程中需要用到許多的工具與函式庫，為了能夠提供更方便且迅速的開發環境，由 National Institute of Standards and Technology(NIST)提出一個專案，最新版本為 1.2 版。NIST-SIP 不僅僅提供包含通訊協定堆疊與開發工具，並且可以幫助使用者開發與測試 SIP 相關的應用程式與 SIP server。

為了能讓[3]的系統更具有移動性與方便性，我們利用由Sun Microsystems所發展的J2ME Wireless Toolkit 2.5.2 與 NetBeans IDE 6.5 beta進行開發和改寫到行動裝置上。在J2ME的開發環境我們需要透過JSR-180 SIP API與JSR-135 MMAPI來支援在J2ME環境中所要使用到的SIP函式庫以及多媒體影音功能。接下來我們將簡單介紹此兩種開發工具與圖 10 系統架構圖：

5.1.1. JSR-180 SIP API

JSR-180 SIP API 最新版本為 JSR-000180 SIP API for J2ME Final Release 3。J2ME SIP API 在 J2ME clients 部分定義了一個多用途的 SIP API。其主要目的是讓 SIP applications 能在記憶體有限的裝置上執行，特別是針對行動電話裝置。J2ME SIP API 可讓 MIDP device 執行 SIP 的 MIDlet，並具下列特性：(a)允許行動裝置回應與啟動 SIP Request；(b)提供 Session management support (c)MIDlet 透過 High level API 來 Access SIP object；(d)可支援各類 SIP 應用，如：Instant Messaging, Presence, Chatting and Gaming。

5.1.2. JSR-135 MMAPI

JSR-135 MMAPI 最新版本為 JSR 135 Mobile Media API for J2ME Final Release 3。JSR-135 MMAPI 提供一個高階的介面使得支援 J2ME 的行動裝置都能夠具備撥放聲音以及多媒體的能力。為了確保在不同的行動裝置上都能享受多媒體服務，在有限記憶體或是資源有限的行動裝置上此 MMAPI 都能夠提供最基本的聲音功能，並且讓 J2ME 應用程式能夠具備多樣性的多媒體功能。

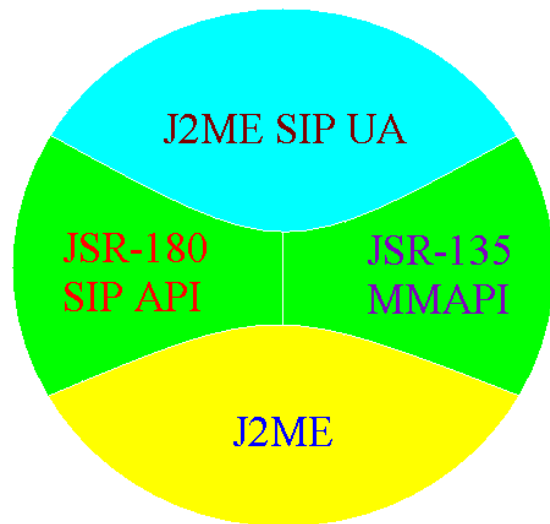


圖 10 J2ME SIP UA

我們將[3]所提出的概念移植到J2ME的J2ME SIP UA上，如圖 11 所示。為了要驗證我們所實作軟體的可行性，接下來將針對連線服務自由轉移的能力進行驗證。

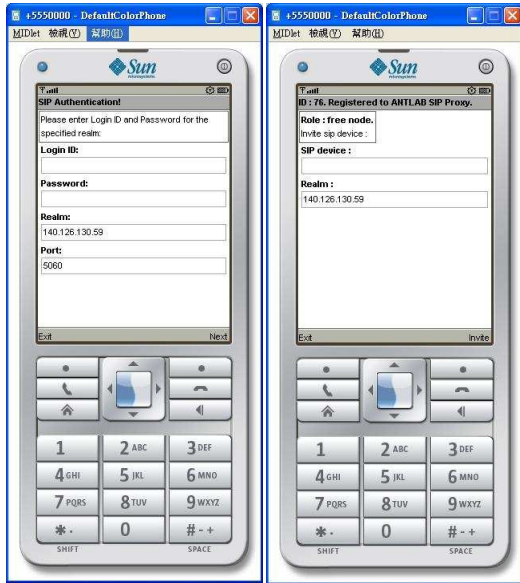


圖 11 系統介面

5.2. REFER scenario

為了要證明透過 J2ME 可將連線服務自由的在不同通訊設備中進行轉移，接下來我們將提出三種連線服務轉移的情境。首先是情境一：將目前正在進行的連線服務進行分流，透過分流可將部分的連線服務轉移至其他的通訊設備。再來是情境二：將轉移至其他通訊設備的連線服務取回。最後是情境三：將全部的連線服務轉移至其他的通訊設備上。

5.2.1. Split partial session

使用者可在連線服務環境改變或是當通訊設備不足時利用分流機制將部分的連線服務轉移至其他的裝置上繼續進行。

由圖 12 可得知原始的連線服務是由 CN1 與 CN2 透過影音設備所建立，假設 CN1 發覺其音訊裝置故障時，可對 MN 發送 REFER request，其內容是利用 MN 與 CN2 建立音訊連線服務。MN 收到 REFER

request 後會回傳 202 accepted 給 CN1，再對 CN2 發送 INVITE request 要求建立音訊連線，接著 MN 再對 CN1 發送 NOTIFY request，表示已經對 CN2 送出 INVITE request。在收到來自行動裝置的 INVITE request 後，CN2 即回傳 200 OK 給 MN，表示願意建立音訊連線。這時 CN1 對 CN2 發送 INVITE request 要求重新建立影像連線服務，CN2 也針對這個請求回傳 200 OK。經過一連串的連線服務請求後，CN1 就能利用原本的影像設備與新加入的 MN 繼續與 CN2 進行通訊。在圖 13 中可發現與 CN2 建立音訊連線服務後，原本是 free node 的 MN 已成為 session user。

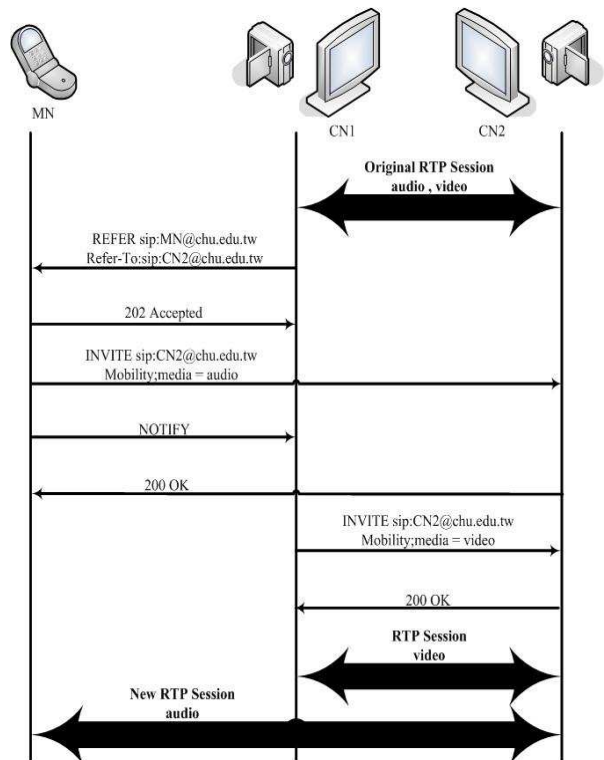


圖 12 情境一：部分連線轉移



圖 13 session user

們可以看到與 CN2 終止音訊連線服務的 MN 由 session user 的角色轉回 free node。

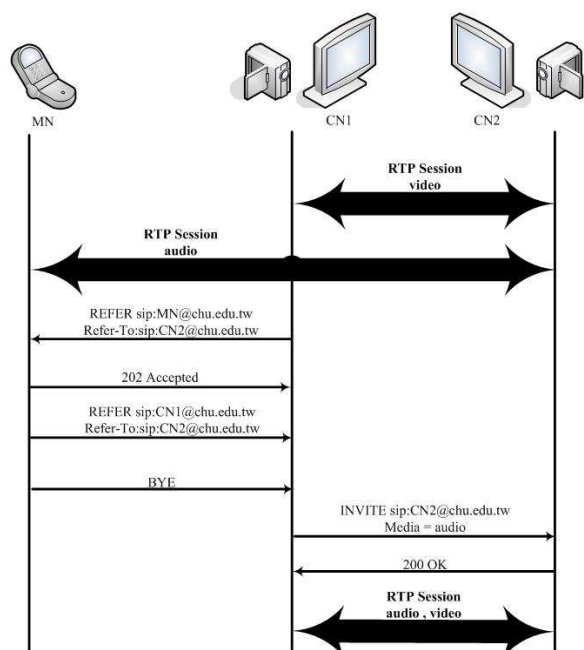


圖 13 情境二：連線服務取回

5.2.2. Retrieve session

在上面的例子裡，因為音訊裝置的故障，使得 CN1 將音訊連線藉由 REFER request 轉移至 MN 上。假設原本連線服務所使用的音訊裝置在 CN1 與 CN2 的連線過程中修復，那麼轉移至 MN 的音訊連線就可以轉移回原始的音訊設備上。

如圖 13 所示，當 CN1 想將音訊連線服務轉移回原本的裝置時，會對 MN 發送 REFER request，收到 REFER request 後，MN 即回傳 202 accepted 並利用 Nested Refer 機制對 CN1 發送 REFER request，將要連線的對象包在 Refer-To 欄位內，並發送 BYE request 終止與 CN2 的音訊連線服務。接著 CN1 再重新對 CN2 發送 INVITE request 要求建立音訊連線，CN2 收到此 request 後即回傳 200 OK。透過 Nested Refer 機制，CN1 即可將轉移出去的連線服務轉移回原始的設備上。在圖 14 中我



圖 14 free node

在情境一中，我們先讓 CN1 與 CN2 建立連線服務。接著由 CN1 對 MN 發出部分連線服務轉移的 REFER request，使 MN 暫時取代 CN1 與 CN2 的部分連線服務。在情境二中，CN1 對 MN 發送部分連線服務取回的命令，表示 MN 暫時取代 CN1 與 CN2 的部分連線將被轉移回 CN1。在圖 14 我們可以看到紅線與黑線，紅線代表的是 SIP 的封包變化量，黑線則是 RTP 的封包變化量。另外縱軸代表的是封包變化量，橫軸表示時間變化。

我們可以看到大約在第 9 秒時 MN 接收到來自 CN1 的部分連線服務轉移的命令，並且在第 12 秒時開始接收來自 CN2 所傳送的 RTP 封包。從圖中我們可以發現大約在第 75 秒時由 CN1 對 MN 發送連線服務取回的命令，此時 MN 即中止對 CN2 的連線服務，CN2 也停止對 MN 傳送 RTP 封包。

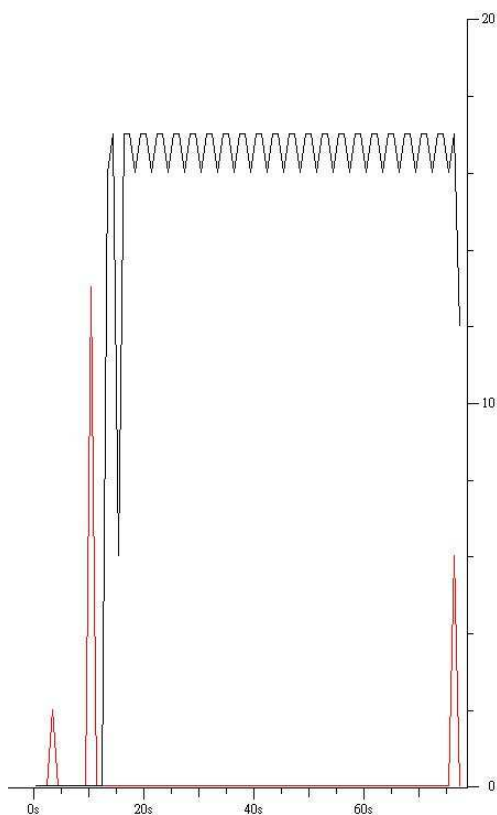


圖 14 連線服務轉移與取回的封包流量

5.2.3. Split session

當所處環境改變，或是裝置的電力不足時，使用者可透過全部連線服務轉移機制，快速的將連線服務進行轉移。

從圖 15 我們可得知連線服務是建立在 MN 與 CN2 之間，MN 的角色也從 free node 成為 session manager，如圖 16 所示。在連線服務的過程中，MN 發現自身的電力即將耗盡，即對 CN1 發送 REFER request，要求連線服務轉移至 CN1 並對 CN2 進行連線服務。CN1 收到此 request 後即對 CN2 發送 INVITE request 要求建立連線服務，接著也對 MN 發送 202 accepted。CN2 收到 CN1 的 INVITE request 後即回傳 200 OK，表示願意建立連線。CN1 在收到 CN2 的 200 OK 後會對 MN 發送 NOTIFY，並回傳 ACK 給 CN2。收到 CN1 發送的 NOTIFY 後，MN 即回傳 200 OK。根據一連串的 REFER request 與 INVITE request 後，連線服務即由 MN 轉移至 CN1 繼續與 CN2 進行連線。當連線在 CN1 與 CN2 建立之後，MN 隨即對 CN2 發送 BYE request 終止連線服務，並且從 session manager 的角色轉變回 free node。

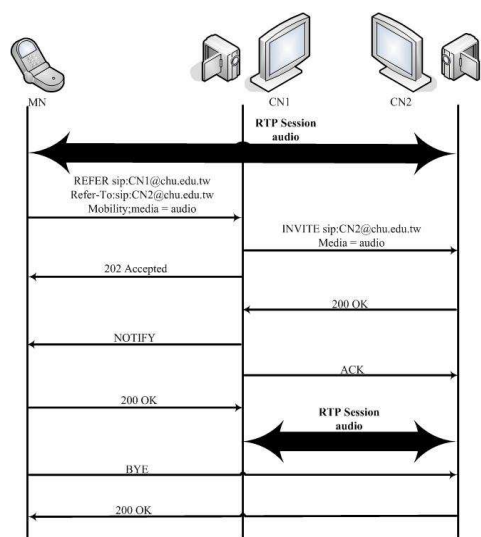


圖 15 情境三：全部連線服務轉移



圖 16 session manager

在情境三中，我們設定先讓 MN 與 CN2 建立連線服務。接著再由 MN 對 CN1 發送全部連線服務轉移的命令，將 MN 與 CN2 的連線服務由 MN 全部轉移至 CN1 後繼續與 CN2 進行連線服務。

如圖 17 我們可以得知大約在第 6 秒時 MN 與 CN2 建立連線，隨即在第 8 秒時開始接收來自 CN2 的 RTP 封包。接收封包的動作一直持續到第 74 秒由 MN 將連線服務全部轉移至 CN1，接著 MN 即中止與 CN2 連線，並且停止接收 RTP 封包。

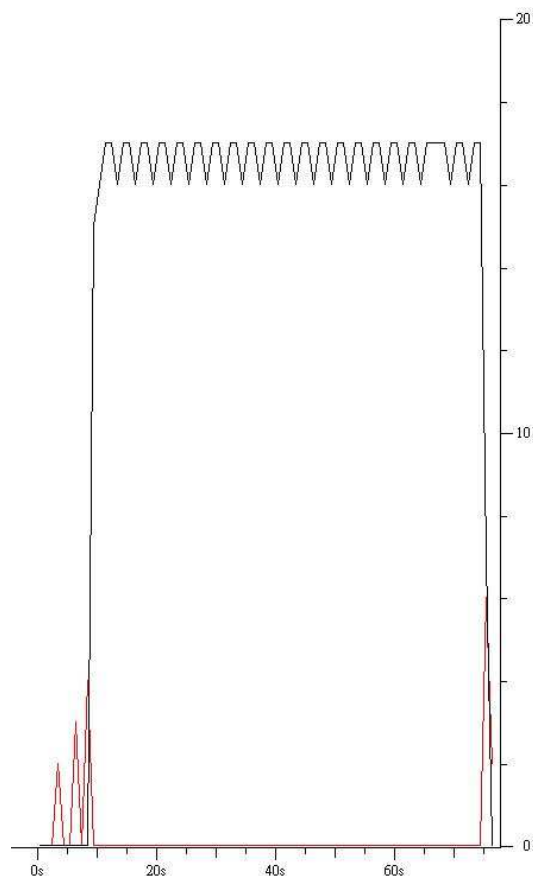


圖 17 連線服務轉移的封包流量

6. Conclusion

在本篇論文中我們提出在 J2ME 的環境下可將目前正在進行的連線服務自由的轉移其他的通訊設備的機制。在實驗的部分驗證了可將連線服務從固定式通訊設備轉移至行動裝置上、或是將轉移至行動裝置上的連線服務取回到固定式通訊設備，以及將行動裝置與固定式通訊設備之間所建立的連線服務由行動裝置轉移至固定式通訊設備上。並且對轉移服務之後的封包變化進行分析。

雖然目前 J2ME 不支援 RTP 的服務，我們只能透過擷取到的封包變化來確認連線服務轉移的正確性，但只要未來 J2ME 支援 RTP 的服務，就能夠讓使用者更方便的根據不同的連線服務需求來使

用各式各樣的通訊設備。

References

- [1]. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, “**SIP: Session Initiation Protocol**”, RFC 3261, IETF, June 2002
- [2]. Min-Xiou Chen, Chen-Jui Peng and Ren-Hung Hwang, “**SSIP: Split a SIP Session over Multiple Devices**” Computer Standards and Interfaces. Vol. 29, No. 5 pp. 531-545, July, 2007.
- [3]. Min-Xiou Chen, and Fu-Ju Wang, “**Session Mobility of SIP over Multiple Devices**”, Tridentcom 2008, Innsbruck, Austria, March 18-20, 2008.
- [4]. R. Sparks, “**The Session Initiation Protocol (SIP) Refer Method**”, RFC 3515, IETF, April 2003
- [5]. J. Rosenberg, J. Peterson, H. Schulzrinne and G. Camarillo, “**Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)**”, RFC 3725, IETF, April 2004
- [6]. R. Sparks, “**The Session Initiation Protocol (SIP) Referred-By Mechanism**”, RFC 3892, IETF, Sep 2004
- [7]. Open Service Gateway Initiative, <http://www.osgi.org>
- [8]. Bushmitch, D.; Wanrong Lin; Bieszczad, A.; Kaplan, A.; Papageorgiou, V.; Pakstas, A., “**A SIP-based Device Communication Service for OSGi Framework**”, Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE, 5-8 Jan. 2004.
- [9]. R. Shacham, H. Schulzrinne, S. Thakolsri, W. Kellerer, “**Session Initiation Protocol (SIP) Session Mobility draft shacham sipping session mobility 04**”, IETF, July 2007
- [10]. Razvan Matei, “**JAIN SIP Approach in Ad-Hoc Networks Mobility Management**”, Research Seminar on Telecommunications Software, Autumn 2002
- [11]. Henning Schulzrinne, ElinWedlundb, “**Application-Layer Mobility Using SIP**”, ACM SIGMOBILE Mobile Computing and Communications Review Volume 4, Issue 3, July 2000
- [12]. Mehdi Mani, Noel Crespi, “**Session Mobility Between Heterogeneous Accesses with the Existence of IMS as the Service Control Overlay**”, 10th IEEE Singapore International Conference on Communication systems, Oct. 2006
- [13]. Oberle, K.; Wahl, S.; Sitek, A., “**Enhanced Methods for SIP based Session Mobility in a Converged Network**”, Mobile and Wireless Communications Summit, 16th IST, July 2007
- [14]. Guiran Chang, Chuan Zhu, Matthew Y. Ma, Weiguo Zhu, Jingbo Zhu, “**Implementing a SIP-based Device Communication Middleware for OSGi Framework with Extension to Wireless Networks**”, 2006
- [15]. Network Research Team, “**SIP-Communicator A Java**

**Softphone based on JAIN SIP with
audio/video and instant ”**

<https://sip-communicator.dev.java.net/>

- [16]. M. Handley and V. Jacobson, “**SDP:
Session Description Protocol**”, RFC
2327,IETF, April 1998
- [17]. H. Schulzrinne, X. Wu, and S.
Sidiroglou, “**Ubiquitous Computing
in Home Networks**” Communications
Magazine, IEEE , Volume: 41 , Issue:
11 , Nov.2003