

國立交通大學

資訊科學與工程研究所

碩士論文

以 JXTA 實作的同儕網路會話發起協議平台

JXTA Implementation of a Peer-to-Peer SIP Platform



研究生：郭芳伯

指導教授：邵家健 教授

中華民國九十七年八月

以 JXTA 實作的同儕網路會話發起協議系統

JXTA Implementation of a Peer-to-Peer SIP Platform

研究生：郭芳伯

Student : Fang-Po Kuo

指導教授：邵家健 教授

Advisor : Dr. John Kar-Kin Zao

國立交通大學

資訊科學與工程研究所



A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master in Computer Science

August 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年八月

以 JXTA 實作的同儕網路會話發起協議平台

學生：郭芳伯

指導教授：邵家健 教授

國立交通大學資訊科學與工程研究所 碩士班

摘要

隨著越來越多的同儕網路應用開始提供多媒體服務，將建立媒體連線的 SIP (Session Initiation Protocol，會話發起協議)與同儕網路架構結合，也因此成為迫切的需要。我們將探索以多用途(general-purpose)同儕網路平台支援 P2P SIP (Peer-to-Peer Session Initiation Protocol，同儕網路會話發起協議)的可能性與實現方法。

我們選用 JXTA 為底層的多用途同儕網路平台，在它之上設計了 P2P 中間層以支撐 SIP 服務。P2P 中間層的設計包括，利用 P2P 架構提供的分散式目錄(distributed directory)來實現 SIP 的定位服務(location service)，以及讓同儕節點(peer node)能夠提供 SIP 伺服器服務的功能，並且能透過同儕網路傳遞 SIP 訊息。

我們不但實現了核心的架構，經過實際的實驗測試後，也確實能夠達到我們的目標，透過多用途同儕網路架構提供 SIP 伺服器服務。我們也將實作成果申請加入 JXTA 社群計畫，並在 GNU GPL 的授權下分享我們的軟體。

JXTA Implementation of a Peer-to-Peer SIP Platform


Student: Fang-Po Kuo

Advisor: Dr. John Kar-Kin Zao

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

The logo of National Chiao Tung University is a circular emblem. It features a gear-like outer border. Inside the circle, there is a stylized building or structure with the letters 'ES' and 'A' on it. Below the building, the year '1896' is inscribed. The entire logo is rendered in a light blue color.

There are more and more Peer-to-Peer (P2P) applications providing multimedia services. Therefore, integrating Session Initiation Protocol (SIP) with P2P infrastructure has become an urgent need. We explore the possibility and realize approach of using a general-purpose P2P platform to support P2P SIP service.

We choose JXTA as the general-purpose P2P platform in the lower layer. Above it, we design a P2P middleware to support SIP service. The functions of the P2P middleware include, using the Distributed Directory provided by P2P infrastructure to realize SIP location service, and making SIP server functionalities on the peers.

We have developed JXTA SIP to implement P2P SIP service upon a general-purpose P2P infrastructure and it really works on a real environment. We also have applied to become a JXTA Community Project and release our software under GNU General Public License (GPL) allowing everyone to download our implementation for future development.

誌謝

首先，感謝邵家健老師一直以來的指導，每次與老師一起研究、討論問題都讓我收穫甚多，能夠有能力做出這篇論文，要歸功於邵老師的諄諄教誨。也非常感謝黃寶儀教授、曹孝櫟教授以及彭文孝教授擔任畢業口試委員，不但指出了我研究的盲點，更給予我許多寶貴的意見，謝謝幾位老師。

接著要謝謝國晉學長，在我每次碰到挫折時，給我的鼓勵與建議，讓我能夠一再地突破挫折，以及輔國與哲民更是一起做研究、寫論文的好友，你們三位對我的論文研究功不可沒。還有在實驗室裡朝夕相處的每個成員們，嘉錡、博政、星閔、梅瑛、勝焜與彥霖，謝謝你(妳)們。

還要謝謝從高中就認識的兩位好友明儒與維逸，與你們一塊相處的日子，是非常值得回憶與珍惜的，感謝你們這麼多年來的相伴。特別感謝良泰、孟均、晏寧、豫琦與姿媚，在我就讀研究所的期間，帶給我無數的歡笑與鼓勵。

最後，要感謝永遠支持著我的家人，再多的篇幅也寫不盡我心中的感謝，謝謝！

目錄

摘要	i
ABSTRACT	ii
誌謝	iii
目錄	iv
圖目錄	vii
Chapter 1 Introduction	1
1.1 Problem Statement.....	1
1.2 Research Approach.....	1
1.3 Thesis Outline.....	2
Chapter 2 Background.....	3
2.1 SIP	3
2.2 Peer-to-Peer SIP	4
2.2.1 dSIP	5
2.2.2 RELOAD.....	6
2.3 JXTA.....	6
2.3.1 Advertisement.....	7
2.3.2 Peer Group.....	7
2.3.3 Rendezvous Super-Peers	8
2.3.4 Pipe	8
2.4 MjSip	9
Chapter 3 Design and Realization.....	10
3.1 Rationale.....	10

3.1.1	Peer-to-Peer (P2P) Middleware Layer for SIP	10
3.1.2	Choice of JXTA	11
3.2	System Architecture.....	11
3.3	Implementation.....	12
3.3.1	Modes of Instantiation.....	12
3.3.2	SIP Layer	13
3.3.2.1	SIP Message Dispatcher: Server Engine Implementation.....	14
3.3.2.2	Registrar	14
3.3.2.3	Proxy.....	15
3.3.2.4	Redirector	15
3.3.3	Peer-to-Peer Layer.....	15
3.3.3.1	SIP Service.....	16
3.3.3.2	User Binding.....	17
3.3.3.3	Peer-to-Peer Location Service.....	18
3.3.3.4	Peer-to-Peer Transport.....	22
3.3.3.5	Message Conversion.....	26
3.3.4	SIP Service Finder	26
Chapter 4	Operation	28
4.1	Registration.....	28
4.2	Proxy: Message Forwarding.....	29
4.3	Redirection	30
4.4	Multiple Domain	30
4.5	Interoperability	31
4.5.1	Interoperation with Conventional SIP Domain	31
4.5.2	Interoperation with other P2PSIP Domain	32

Chapter 5	Experiment	33
5.1	Environment Setup	33
5.2	Results	33
5.2.1	Registration.....	34
5.2.2	Session Setup.....	34
5.2.3	Session Termination.....	35
Chapter 6	Conclusion	36
6.1	Accomplishment.....	36
6.2	Future Work.....	36
Reference	37
Appendix A	39
Appendix B	40



圖目錄

圖 1：SIP 的基礎梯型架構	3
圖 2：P2P SIP 同儕網路參考模型	4
圖 3：P2P UA 之間建立會話連線的步驟	5
圖 4：dSIP 示意圖	5
圖 5：JXTA 架構圖	7
圖 6：JXTA 集結節點運作圖	8
圖 7：MjSip Server 相關類別圖	9
圖 8：JXTA SIP 架構圖	12
圖 9：JXTA SIP 運作模式	13
圖 10：ServerEngineImpl 類別圖	13
圖 11：ServerEngine 與 ServerEngineImpl 繼承關係類別圖	14
圖 12：Proxy 與 P2P Transport 的訊息傳遞方式	15
圖 13：SIP Service 類別圖	16
圖 14：SIP Service 啟始動作循序圖	17
圖 15：User Binding 與 JXTA User Binding 繼承關係類別圖	18
圖 16：User Binding Advertisement 類別圖	19
圖 17：JXTA Location Service 更新 contact address 的循序圖	20
圖 18：JXTA Location Service 搜尋遠端 User Binding 的循序圖	22
圖 19：P2P Transport 與 P2P Transport Listener 相關類別圖	23
圖 20：Pipe Transport 類別圖	23
圖 21：Pipe Transport 的建立與接收訊息的循序圖	24
圖 22：Pipe Transport 傳送訊息的循序圖	24

圖 23 : BiDi Pipe Transport 相關類別圖	25
圖 24 : BiDi Pipe Transport 的建立與接收訊息的循序圖	25
圖 25 : BiDi Pipe Transport 傳送訊息的循序圖	26
圖 26 : Registration 的運作方式.....	28
圖 27 : Proxy: Message Forwarding.....	29
圖 28 : Redirection 運作方式	30
圖 29 : Interoperation with Conventional SIP Domain	31
圖 30 : 實驗配置圖	33



Chapter 1 Introduction

1.1 Problem Statement

隨著越來越多的多媒體服務從以往的 client/server 架構轉移到 Peer-to-Peer(P2P)的架構上運作，將現有 SIP(Session Initiation Protocol，會話發起協議)[1,2]建構在 P2P 架構之上的 Peer-to-Peer SIP[3,4,5]也理所當然地成為迫切需要的服務。

本研究探索以 general-purpose 的 Peer-to-Peer(P2P)平台支援 Peer-to-Peer SIP 服務之可能性與實現方法。

IEFT¹的 P2PSIP Working Group 提出兩種不同的實現途徑：一種是將 SIP 擴充成同儕網路(Peer-to-Peer overlay network)的管理協定，例如 dSIP[6]；另一種則是利用 general-purpose 的 P2P 架構，定義一個 P2P 中間層(middleware)，讓 SIP 運作在 P2P 中間層上面，例如 RELOAD(RESource LOCation And Discovery)[7]就是其中一種實現方式。這個做法不需要更動到 SIP 本身，可以保證與 SIP 的相容性，甚至往後 SIP 的改變也不會影響到同儕網路使用的命名(naming)與搜尋(search)機制；同樣的，若同儕網路所使用的命名與搜尋機制改變，亦會不影響 SIP，SIP 層與 P2P 層的運作方式是不會互相影響的，因此本論文將採用類似後者的概念進行設計探討與實作。

1.2 Research Approach

本論文從兩個不同層面進行探討並提出研究方法，以利用 general-purpose P2P 來實現 P2P SIP。

首先本論文探討如何用 P2P 架構來提供整套的支援服務。SIP 伺服器的主要工作就是透過定位服務(Location Service)的資訊以決定 SIP 訊息的傳遞方向。定位服務就是要提供類似 DNS(Domain Name System)的服務，給予定位服務一個使用者的 AoR(Address of

¹ The Internet Engineering Task Force，<http://www.ietf.org/>。

Record) ，則它會傳回零到數個 URI(Uniform Resource Identifier) ，告訴 SIP 伺服器更接近該使用者的位置，或者直接為該使用者目前的位置。因此定位服務也可以視為一種目錄服務(directory service) ，本研究就提出了使用 P2P 架構提供的分散式目錄(distributed directory)來支援定位服務的實現方法。

另一個層面就是研究如何用同儕節點(peer node)來實現 SIP 伺服器，以及要實現哪些 SIP 伺服器的功能。SIP 的基本伺服器包括註冊伺服器(registrar server)、代理伺服器(proxy server)以及轉向伺服器(redirect server)，而這三種伺服器幾乎涵蓋了 SIP 伺服器的所有功能。本論文將提出用同儕節點來實作這三種伺服器的設計方法以及同儕節點的部署與運作方式。

本研究將在 general-purpose 的 P2P 平台 JXTA 上實作上面提到的設計。

1.3 Thesis Outline

接下來的文章將分成五個章節：第二章介紹基本的背景知識，包括 SIP、P2P SIP、JXTA 與 MjSip 等。第三章說明本研究如何在 JXTA 平台上設計與實作整個 P2P SIP 服務，包括設計的概念與實作的細節。第四章為本研究所設計的 JXTA SIP 的運作過程，除了 JXTA SIP 本身的基本運作，也考慮了與 conventional SIP 互動的運作方式。第五章是利用 JXTA SIP 進行註冊(registration)、會話建立(session setup)、會話結束(session termination)等幾項 SIP 基本操作的環境設定與測試結果。第六章總結本研究的貢獻與未來可行的方向。

Chapter 2 Background

2.1 SIP

SIP(Session Initiation Protocol)是以 HTTP(Hypertext Transport Protocol)為基礎的文字編碼協定，主要功能在點對點通訊(peer-to-peer communication)的會話發起(session initiation)。SIP 使用類似 HTTP 的主從式架構(client-server transaction model)，SIP 客戶端(user agent client, UAC)會產生 SIP 要求(request)給 SIP 伺服器端(user agent server, UAS)，SIP 伺服器端會對 SIP 要求產生回應，根據互傳給對方的 SDP 內容 (Session Description Protocol, 會話描述協定)決定相互溝通時要用的媒體格式。

SIP 網路的基礎元件包括使用者代理人(user agent)、代理伺服器(proxy server)、轉向伺服器(redirect server)註冊伺服器(registrar server)與定位服務(location service)。

SIP 中的關鍵就是如何找出 AoR(Address of Record)與對應的 contact address，一個 AoR 就代表著一個使用者，但是該使用者實際的位置卻可能會不停地改變，而使用者的 AoR 與使用者所在位置的對應關係都會被記錄在定位伺服器裡，UA 就需要藉由代理伺服器或轉向伺服器，以及定位服務等才能連繫到對方，進而透過 SIP 建立媒體會話(media session)。

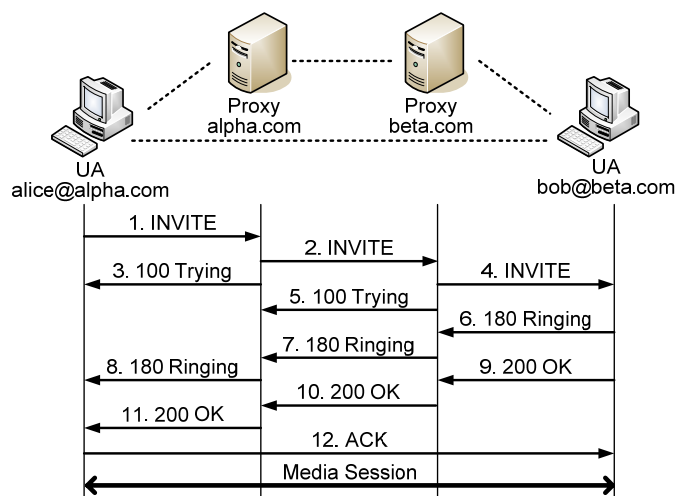


圖 1：SIP 的基礎梯型架構

2.2 Peer-to-Peer SIP

P2P SIP(Peer-to-Peer Session Initiation Protocol，同儕網路會話發起協議)是由 IETF P2P SIP Working Group 所制定的一套通訊協定，它的目的是要讓 SIP 能夠藉由 peer-to-peer 的技術找出 SIP 要求的目的地、提供 SIP 訊息的傳輸以及其他 SIP 相關的功能。

P2P SIP 網路裡的節點可分為 P2P SIP Peer 與 P2P SIP Client。P2P SIP Peer 是在 P2P SIP 網路中提供其他 peer 儲存、傳輸與尋找服務的 peer。而 P2P SIP client 是在 P2P SIP 網路中不提供路由、儲存與查詢功能的節點。P2P SIP Peer Protocol 定義了 P2P SIP Peer 之間的溝通方式；P2P SIP Client Protocol 定義 P2P SIP Peer 與 P2P SIP Client 的通訊協定。P2P SIP Peer Protocol 提供了維護整個同儕網路、儲存與擷取資訊以及遞送訊息的機制。圖 2 為 P2P SIP 同儕網路的參考模型。

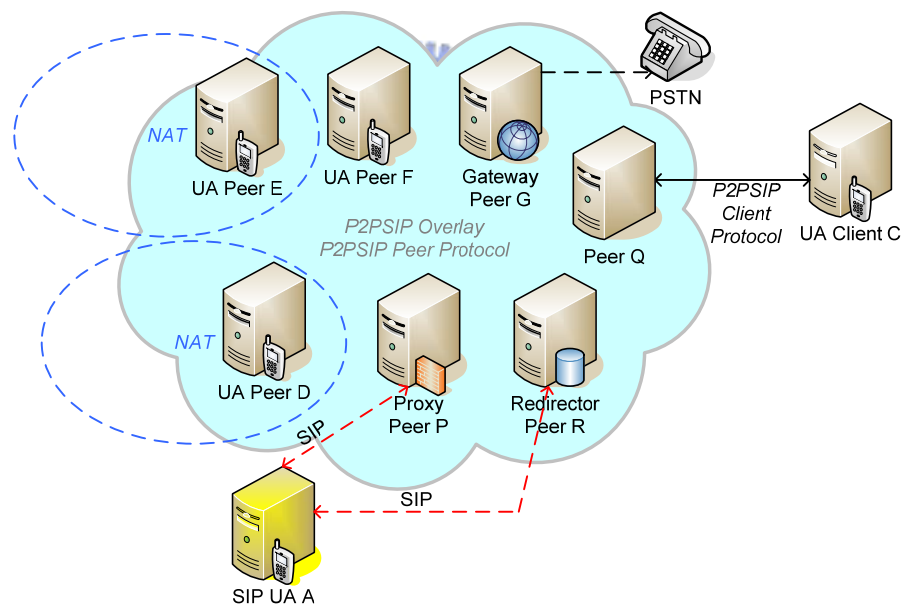


圖 2：P2P SIP 同儕網路參考模型

P2P SIP 是一個雙層的架構，如圖 3 所示，在 SIP 層下面加上 P2P SIP 來負責尋找資源。當 SIP UAC 要和 SIP UAS 建立會話連線時，SIP UAC 會先向下方的 P2P SIP 層的 P2P SIP client 詢問 SIP UAS 的 contact，接著 P2P SIP client 會再詢問 P2PSIP peer。P2P SIP peer 會用 DHT 的技術找到持有 SIP UAS contact 的 P2P SIP peer，跟它要求該筆資料後傳回給 SIP UAC 的 P2PSIP client，然後往上再傳給 SIP UAC。此時兩個 UA 就可以直接用 SIP

訊息互相溝通，接下來的步驟就與傳統的 SIP 一樣。

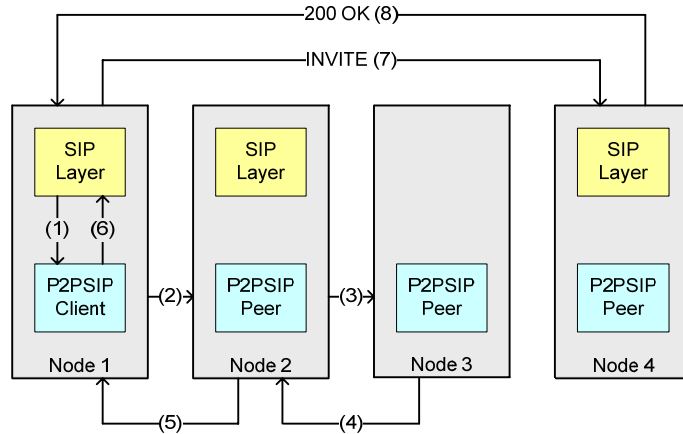


圖 3：P2P UA 之間建立會話連線的步驟

下面介紹 dSIP 與 RELOAD，兩種不同的 P2P SIP Peer Protocol 實現途徑。

2.2.1 dSIP

dSIP 是 P2P SIP Peer Protocol 的一種實現方式，它利用 DHT(distributed hash table)[8,9] 來管理整個 P2P SIP 網路，提供資源註冊(resource registration)與資源搜尋(resource discovery)的功能，並且使用 SIP 訊息作為 P2P SIP Peer Protocol 的編碼基礎。為了能夠適當地實現 P2P SIP Protocol，dSIP 將 SIP 擴充，加入一些新定義的 header。因為可以直接支援普通的 SIP，dSIP 不需要額外地設計 P2P SIP Client Protocol，所以 dSIP 裡就沒有 P2P SIP Client Protocol 的存在。

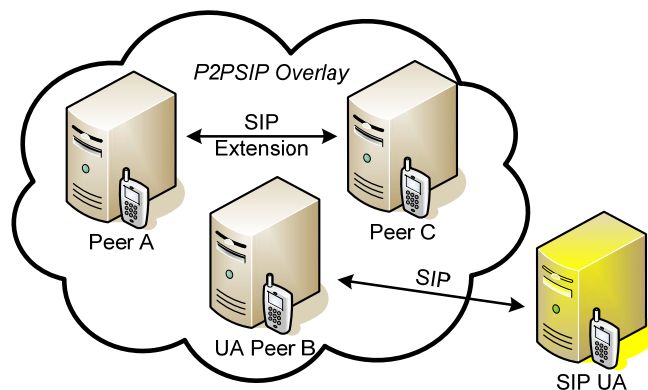


圖 4：dSIP 示意圖

dSIP 的設計可以允許使用不同的 DHT，並且都支援一個基本的 DHT 演算法 Chord，以

保持相容性。

2.2.2 RELOAD

RELOAD(RESource LOcation And Discovery)是由 dSIP 演變而來的另一種 P2P SIP Peer Protocol 實現途徑。它一樣利用 DHT 來達到資源註冊與資源搜尋的功能，並提出了交換 P2P 訊息的架構，也允許抽換 DHT 的演算法，但是至少要能支援 Chord。

RELOAD 與 dSIP 的最大的差別在於，RELOAD 為了避免 SIP 的誤用，將 SIP 從 P2P SIP Peer Protocol 中完全移出，使 P2P SIP Peer Protocol 與 SIP 相互獨立。

RELOAD 目前已有的商品化的實作¹。

2.3 JXTA

JXTA[10,11]是一個專為同儕網路而設計的開放式網路平台，其目標是提供基本元件、服務(services)及協定(protocols)，協助開發以同儕節點群組(peer group)為主的應用程式。

“JXTA” 這個詞是來自 juxtapose，意為並列(side by side)。這個名字的意義在於，相對於現今傳統分散式運算架構，如客戶端-伺服器架構(client-server)或是以網路為基礎的計算架構(web-based computing)，JXTA 針對的 peer-to-peer 是一個並行等價的概念。它提供了一系列基本的開放式協定(open protocol)和一個發展同儕網路應用程式的平台。

¹ SIPeerior Technologies P2PSIP Core and Endpoint Development Kits，<http://www.sipeerior.com/products.php>

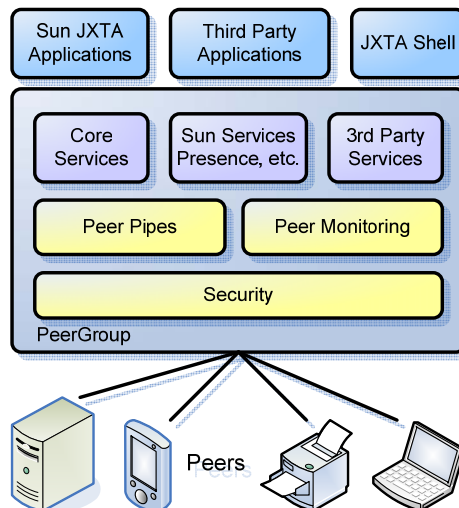


圖 5：JXTA 架構圖

下面介紹 JXTA 中的幾個基本單元：

2.3.1 Advertisement

公告(advertisement)是用來描述 JXTA 網路裡各種網路資源的 XML 文件，所描述的網路資源包括同儕節點、同儕節點群組、管線(pipe)、及服務等。同儕節點要先搜尋到該項資源的公告，再藉由公告裡的資訊來取得並使用該項資源。

2.3.2 Peer Group

同儕節點群組(peer group)是一群擁有共同特徵的同儕節點的集合。共同特徵包括遵循相同的策略(policies)、提供或需要類似的服務、擁有相似的興趣傾向等。同儕節點群組是由同儕節點自行組織而成的，使用者、服務開發者和網路管理員可以動態地產生新的同儕節點群組，將同儕節點依照不同的需求及互動來劃分。JXTA 協定只負責描述同儕節點如何發佈、尋找、加入及監視同儕節點群組，不會控制任何同儕節點群組產生的時間、地點及原因。同儕節點群組可以建立自己的成員資格策略(membership policy)，從完全公開到需要驗證的高度安全性及保護性都可以設定。每一個同儕節點可以同時屬於多個同儕節點群組，提高了許多彈性。

2.3.3 Rendezvous Super-Peers

集結節點是一個擁有額外功能的同儕節點，它會快取(cache)一些同儕節點的公告的索引(index)，並且傳遞「發現要求」(discovery request)，幫忙其他同儕節點尋找資源。當一個同儕節點加入同儕節點群組時，它會自動地搜索集結節點，並且與集結節點交換資訊。若沒有找到任何集結節點，它會動態地成為該同儕節點群組的集結節點。每一個集結節點都會維護一份清單，裡面記錄著它所知道的其他集結節點以及它所服務的同儕節點，集結節點之間會利用非嚴謹同步的 DHT(loosely-consistent distributed hash tables)[12]來協尋網路上的資源。同儕節點會將「發現要求」傳給集結節點，集結節點則會幫忙傳遞這個要求給其他的集結節點與同儕節點，直到有同儕節點能夠回應這個要求或者要求逾時為止。

以圖 6 為例，下半部為實體網路(physical network)，上半部為 JXTA 虛擬網路(virtual network)。同儕節點 Peer A 發出一個「詢問」(query)給集結節點 Rdv 1，Rdv 1 再將它傳給其他的集結節點。其中集結節點 Rdv 2 找到能夠回應這個「詢問」的同儕節點，便轉送給同儕節點 Peer B。最後 Peer B 收到後依據裡面的資訊，直接回應 Peer A。

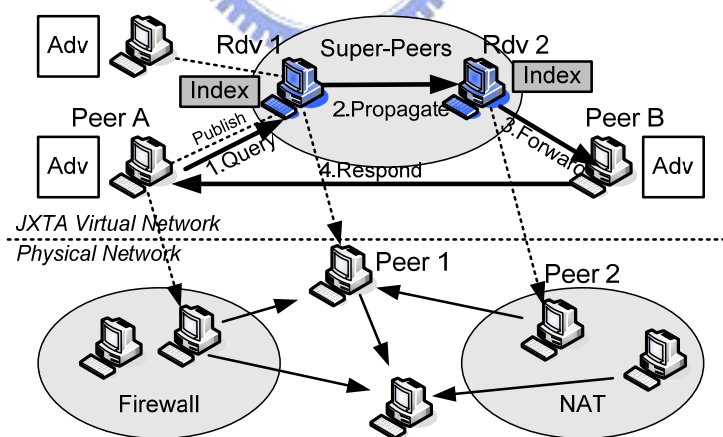


圖 6：JXTA 集結節點運作圖

2.3.4 Pipe

管線(pipe)是服務與應用程式之間用來傳送及接受訊息(message)的虛擬通道(virtual

communication channel)，它將訊息的實體傳遞方式隱藏起來並且提供抽象介面 (abstraction)，使服務及應用程式以簡單的輸出與輸入方式操作。管線的建立必須要有一個輸入管線(input pipe)和一個以上的輸出管線(output pipe)，輸入管線的資訊會藉由管線公告(pipe advertisement)型式發佈出去，讓其他同儕節點找到，並利用管線公告的資訊動態地產生一個輸出管線與其輸入管線連接，而其他同儕節點傳入輸出管線的訊息，都會被傳送到連接著的輸入管線，產生該輸入管線的服務或應用程式便可取得訊息。

2.4 MjSip

MjSip[13,14]是由 Parma 大學¹與 Roma Tor Vergata 大學²共同發展的一個套件。MjSip 用 Java 實現了支援 RFC 3261 SIP 規範的 SIP stack，同時提供 SIP stack 的 API 與實作，並且實作了 SIP 系統中使用者代理人、註冊伺服器、代理伺服器以及定位服務等幾個基礎元件。在本論文的設計裡會使用 MjSip 作為 SIP stack 的部份，並且參考它的 Server 相關類別，來設計 JXTA SIP 的 SIP Server。

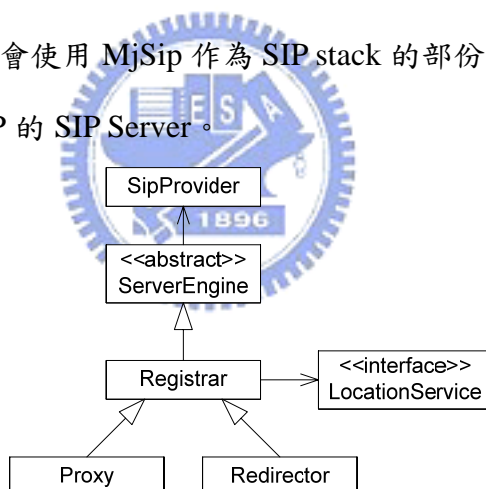


圖 7：MjSip Server 相關類別圖

¹ University of Parma (Università degli Studi di Parma)，<http://www.unipr.it/>。

² University of Roma Tor Vergata(Università degli Studi di Roma Tor Vergata)，<http://web.uniroma2.it/>。

Chapter 3 Design and Realization

3.1 Rationale

針對 Research Approach 所提出的設計方法，下面將會闡述我們的理由。

3.1.1 Peer-to-Peer (P2P) Middleware Layer for SIP

本論文發展 P2P 中間層來支援 SIP 服務，有兩個主要的原因。誠如之前提過的，定位服務(location service)是 SIP 中非常關鍵的功能，像 P2P SIP 裡的一個重點也是在設計定位服務。註冊伺服器(registrar server)會向定位服務登錄使用者的 AoR(Address of Record)與其 contact address，而代理伺服器(proxy server)與轉向伺服器(redirect server)會拿使用者的 AoR 向定位服務詢問使用者的 contact address。簡單來說定位服務就如同一個目錄服務(directory service)，而 P2P 架構的特色之一就是非常適合實現分散式目錄(distributed directory)，像是 DHT(distributed hash table)等。因此，使用 P2P 架構提供的分散式目錄來支援定位服務是絕對可行且非常適合的方式。

另一個原因在於同儕節點的部署所帶來的好處。以原有的 SIP 而言，要建立一個 SIP 網路架構，需要部署許多專用的 SIP 伺服器來服務為數更多的 SIP UA(user agent)。然而，專用伺服器的建置與維護卻不是件容易的事，但 UA 一定要透過 DNS 伺服器或 SIP 伺服器才能夠互通溝通。

而在 P2P 的架構中，沒有專用的伺服器，每一個同儕節點都可以成為一個 SIP 伺服器，若不是 SIP 伺服器，則該同儕節點也能幫忙尋找其他有提供 SIP 伺服器功能的同儕節點，來因應 UA 的需要。因此，若 P2P SIP 架構經過適當地部署，UA 只要找到任何一個同儕節點，就可以與其他 UA 進行連線。在我們的設計中，一個 UA 也可以同時是同儕節點，在這種情況下，UA 甚至不需要找尋求其他伺服器的協助，直接利用並存的同儕節點提供的 SIP 伺服器功能即可。

再者，P2P 系統裡，同儕節點存在的機率與廣泛性一定比原有 client-server 架構裡的 SIP 伺服器還要高出許多。而且要從頭開始建置一個 P2P 架構來提供 SIP 服務，也勢必比從頭建置整個伺服器架構要容易得多，甚至要建構一個全球性的結構也是絕對可行的。因此我們選用 general-purpose P2P 架構來支援 SIP 絕對是一個非常好的選擇。

3.1.2 Choice of JXTA

本論文選用 JXTA 作為底層的 P2P 架構，主要是因為 JXTA 為一個很常用且非常完整的 P2P 發展平台。若是只為了要支援 P2P SIP 而選擇一個 P2P 架構，JXTA 未必會是最適合的平台。以 RELOAD 為例，它已經定義了一個支援 P2P SIP 所需要的最基本架構，相對於我們的做法，RELOAD 算是輕型(lightweight)的架構。倘若日後還要實現 SIP 以外的其他機制，例如整個多媒體系統，則類似 RELOAD 的輕型架構就會不敷使用。再者，若要從頭開始建立一個新的、全球性的 P2P 多媒體平台，正確的方法應該是用一個 general-purpose 的 P2P 架構來支撐它，而且這個架構必須是很普遍與常用的。在這種情況下，更應該選用 JXTA，因為它不但提供了最全面且完整的 general-purpose P2P 架構，JXTA 也是發展多年逐漸趨於成熟、且已經被運用在許多應用上的平台。JXTA 的普遍性、跨平台以及模組化的設計，並且目前 JXTA 平台上尚未有實際可行的 SIP 服務模組。然而 JXTA 本身也有些不容忽視的缺點，如 Rendezvous Network 會有資料不同步(Data Inconsistency)與資料無法取得(Unavailability)等問題。不過這些問題可以藉由可能已經可實作用的 scalable consistent DHT¹(e.g. Chord)的導入來解決，但 DHT 的實作已超過本研究的範疇，故不在此討論。

3.2 System Architecture

本論文將 JXTA SIP 設計成一個 JXTA Service，讓同儕節點啟始這個 JXTA Service 後就能夠提供 SIP 伺服器的服務。JXTA SIP 是一個中間層(middleware)，包含 P2P Location

¹ METEOR Distributed Hash Table， <https://jxta-meteor.dev.java.net/>。

Service 與 P2P Transport 兩個介面，P2P Location Service 提供搜尋 URI 與 contact 的 binding 的功能，而 P2P Transport 則用來進行 SIP 訊息的傳輸。JXTA SIP 利用下層 JXTA 的 Discovery Service 實作一個 P2P Location Service，也會透過 JXTA 的 Pipe Service 實作 P2P Transport 以提供同儕網路上傳輸的能力，並且使用上層的 MjSip 處理與傳輸 SIP 訊息，進而完成 SIP 伺服器的工作。

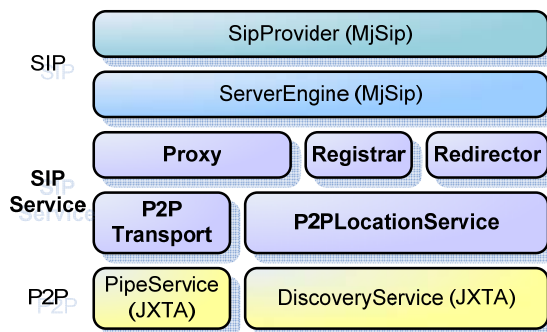


圖 8：JXTA SIP 架構圖

3.3 Implementation

首先會描述 JXTA SIP 的三種啟始模式。接著針對本研究的重點 SIP Service，分別探討它的 SIP 層和 Peer-to-Peer 層的設計與實作。最後介紹 SIP Service Finder 的概念。更詳細的實作細節，請參閱 JXTA SIP 的 API。

3.3.1 Modes of Instantiation

Main 是啟始整個 JXTA SIP 的起點，它會初始化一個 JXTA Peer，加入 JXTA Net Peer Group，接著根據參數的不同，有底下三種不同的運作模式：

- 如圖 9a，執行 SIP Service 成為一個 JXTA SIP Peer，提供 SIP Registrar、Proxy 或 Redirect Server 的功能，並參與同儕網路。
- 圖 9b 和前一個模式的不同之處，在 SIP Service 啟動完成後，會接著跑起 MjSip 的 Command Line UA，並將 SIP Service 形成的 JXTA SIP Peer 視為 outbound proxy，同時運行 SIP Server 與 UA。

- 圖 9c 所示的模式不會執行 SIP Service，而是透過 SIP Service Finder 搜尋 JXTA SIP Peer，找到後如同前一個模式，執行 MjSip 的 Command Line UA，並將找到的 Peer 設定為該 UA 的 outbound proxy。

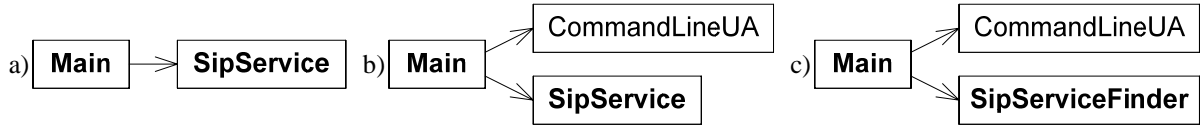


圖 9：JXTA SIP 運作模式

3.3.2 SIP Layer

SIP Service 的 SIP 層主要是用來處理與傳輸 SIP 訊息。如圖 10 所示，本論文利用 MjSip 提供的 SIP Provider 作為 SIP 傳輸層(transport layer)。MjSip 的 Server Engine 提供了 SIP 伺服器最基本的運作，本論文設計的 Server Engine Implementation 繼承了 Server Engine，它會依據 SIP 訊息的內容，把 SIP 訊息傳遞給 Proxy、Registrar 或 Redirector 其中之一。Registrar 會根據 REGISTER 要求向 P2P Location Service 進行更新。Proxy 和 Redirector 會向 P2P Location Service 查詢訊息該傳送至哪裡。Proxy 會透過 P2P Transport 介面將訊息交由 P2P 層傳送出去，以及接收從 P2P 層收到的訊息。Redirector 則會利用 SIP Provider 將 contact address 回傳給 UA，讓 UA 把訊息導向其他位置。

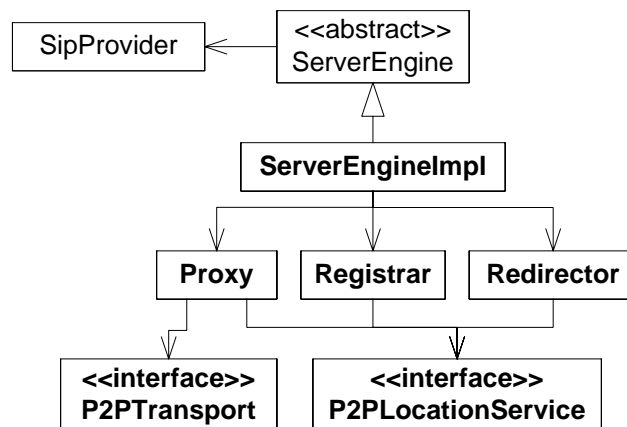


圖 10：ServerEngineImpl 類別圖

3.3.2.1 SIP Message Dispatcher: Server Engine Implementation

Server Engine Implementation 負責將收到的 SIP 訊息傳給適當的 SIP Server 來處理，它實作了 Server Engine 的四個抽象方法(如圖 11)：

1. `processRequestToLocalServer` 被呼叫時(即收到 Request 是要傳給 Local Server)，就會把 Request 交給 Registrar 進行註冊(Registration)。
2. `processRequestToRemoteUA` 被呼叫時(即收到的 Request 是要傳給遠端的 UA)，它會將 Request 交由 Proxy 進行 Request Forwarding 或讓 Redirector 回傳 Redirection 訊息「302 Moved Temporarily」。若 Proxy 和 Redirector 兩者並存時，兩者的選擇方式不在本研究的討論範圍內，目前的實作是優先傳給 Proxy 處理。
3. `processRequestToLocalUser` 被呼叫時(即 Request 是要傳給本地端的 UA)，就交由 Proxy 處理。
4. `processResponse` 被呼叫時(即收到 Response)也一律傳遞給 Proxy 進行下去。

Server Engine Implementation 還提供函數以啟始 Registrar、Proxy 和 Redirector。

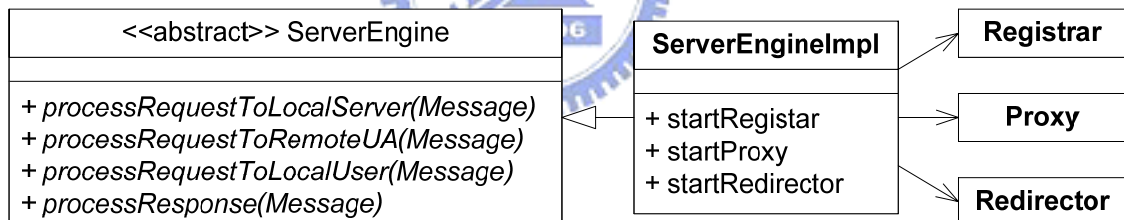


圖 11：ServerEngine 與 ServerEngineImpl 繼承關係類別圖

3.3.2.2 Registrar

Registrar 用來提供 SIP 註冊的服務。MjSip 的 Registrar 在建構時會產生 Location Service。本論文設計的 Registrar 並不會自行產生 Location Service，而是直接使用先前建立好的 P2P Location Service。其他細節的運作，例如依據 Registration 更新 P2P Location Service，皆與 MjSip 的 Registrar 類似。

3.3.2.3 Proxy

Proxy 會透過同儕網路提供 UA 之間的傳輸服務。Proxy 收到的訊息若是要傳給本地端 UA，它會向 P2P Location Service 查詢該 UA 的 contact address，並透過 MjSip 的 SIP Provider 把訊息送給 UA。

若訊息的目的地為遠端的 UA，則 Proxy 會呼叫 P2P Location Service 的 get Remote User Binding，以取得遠端 UA 的 binding 資訊。再將訊息與 binding 資訊傳給 P2P Transport 進行 forwarding。為了能夠接收經由 P2P Transport 傳遞過來的訊息，Proxy 還實作了 P2P Transport Listener。(如圖 12 所示)

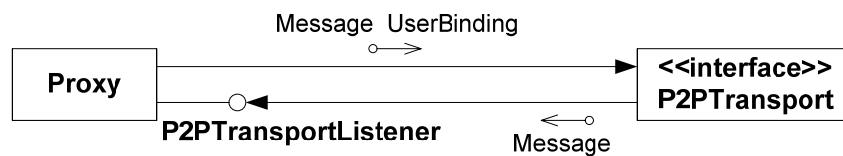


圖 12：Proxy 與 P2P Transport 的訊息傳遞方式

3.3.2.4 Redirector

Redirector 會提供 SIP 轉向服務，把 UA 的要求重新向導其他地方。Redirector 收到訊息後會向 P2P Location Service 查詢目的地 UA 的 contact address，並回應一個 Response Code 為「302 Moved Temporarily」的訊息給發送者。

3.3.3 Peer-to-Peer Layer

SIP Service 的 P2P 層會透過同儕網路進行資源搜尋與訊息傳遞，以提供上層 Location Service 與傳輸的功能。圖 13 為 SIP Service 相關的類別圖，SIP Service 會先加入 JXTA SIP Peer Group，然後分別跑起 P2P Transport 以透過同儕網路傳輸訊息、P2P Location Service 提供 Location Service 的功能、Server Engine Implementation 以啟始整個 SIP 層。

本論文設計了兩種 P2P Transport 的實作，Pipe Transport 與 BiDi Pipe Transport，兩者都是利用 JXTA 的 Pipe Service 產生 JXTA Pipe 以收發訊息。Pipe Transport 使用 JXTA 的

Input Pipe 接收訊息，用 Output Pipe 送出訊息。而 BiDi Pipe Transport 則是用 JXTA 的 Bidirectional Pipe(JxtaBiDiPipe)收送訊息。

P2P Location Service 的實作 JXTA Location Service 會使用 JXTA 的 Discovery Service 發佈與查詢 binding。

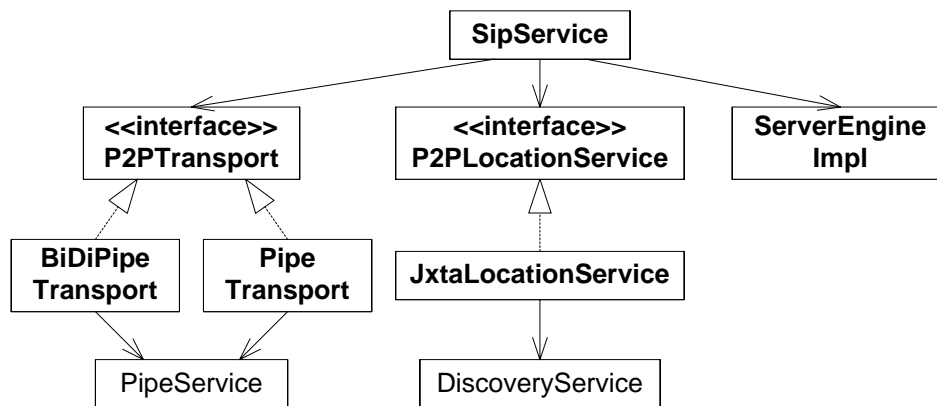


圖 13：SIP Service 類別圖

3.3.3.1 SIP Service

1. SIP Service 會先尋找 JXTA SIP Peer Group 的 Advertisement，利用該 Advertisement 加入其中。若沒有找到 JXTA SIP Peer Group 的 Advertisement，則它會自行建立 JXTA SIP Peer Group，產生 JXTA SIP Peer Group 的 Advertisement，並將 Advertisement 發佈出去。
2. 然後產生一個 P2P Location Service，預設是使用本論文實作的 JXTA P2P Location Service 來提供 P2P Location Service。
3. 若要提供 Proxy 的功能，就會建立 P2P Transport 來透過同儕網路收發訊息。此時會根據參數選擇使用 Pipe Transport 或 BiDi Pipe Transport，兩者的差別將會在 3.3.3.4 裡說明。
4. 產生 Server Engine Implementation 以提供 SIP 層的功能，並將 P2P Location Service 與 P2P Transport 傳給它使用。
5. 最後將所提供的 SIP Server 的功能以 Advertisement 的形式發佈到 JXTA Net Peer

Group 中，讓 SIP Service Finder 能找這個 JXTA SIP Peer。

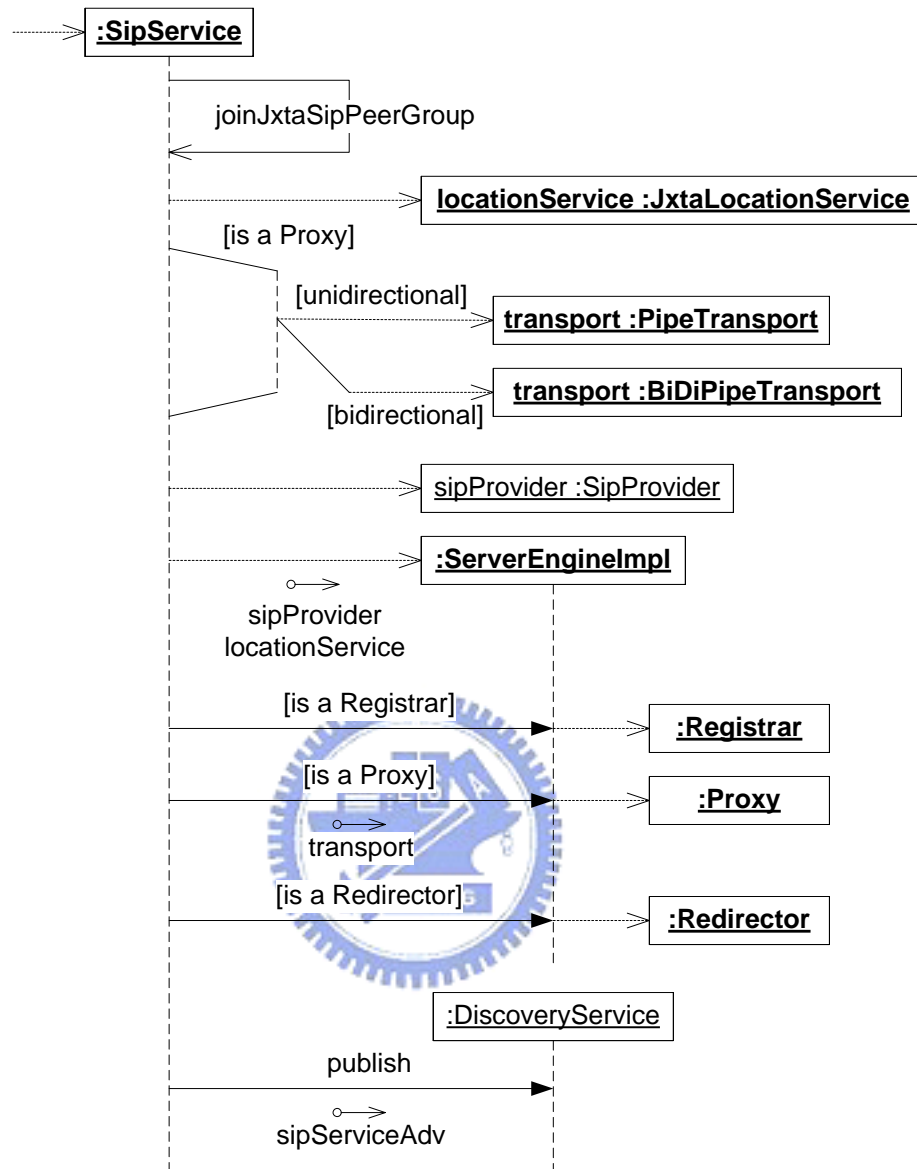


圖 14：SIP Service 啟始動作循序圖

3.3.3.2 User Binding

本論文設計了 User Binding 類別來記錄 AoR(Address of Record) URI 與 contact address 的對應關係。每個 User Binding 裡包含一個代表 AoR 的 URI 以及一個 Contact Header List，記錄著一至多個的 contact address 資訊。

P2P Location Service 要可以更新與查詢 User Binding，而 P2P Transport 要藉由 User Binding 的資訊才能透過同儕網路傳遞訊息。因此根據不同的 P2P Transport 與 P2P

Location Service 實作方式，可以將 User Binding 擴充成所需要的樣子，只要以繼承 User Binding 的方式擴充上層 SIP Layer 的運作都不會受到影響。

本論文所實作的 Pipe Transport 與 BiDi Pipe Transport 就會需要 Pipe Advertisement 來建立 Pipe 傳遞訊息。因此，我們設計了 JXTA User Binding 這個類別，它繼承了原有的 User Binding，再加上 Pipe Advertisement 的資訊。每個 JXTA User Binding 會有對應的 User Binding Advertisement，而 JXTA User Binding 還要記錄對應的 User Binding Advertisement 的 ID，用途會在 3.3.3.3 說明。最後所設計出來的資料結構如圖 15 所示：

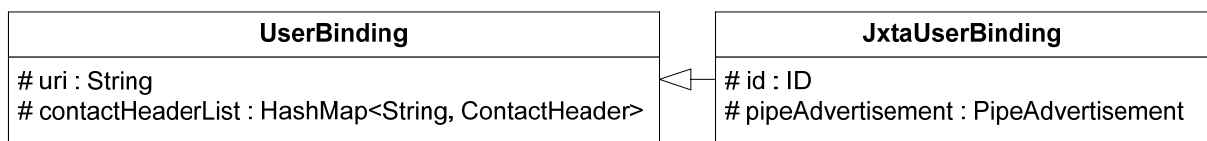


圖 15：User Binding 與 JXTA User Binding 繼承關係類別圖

3.3.3.3 Peer-to-Peer Location Service

P2P Location Service 介面的主要功能就是提供更新 User Binding 與查詢 User Binding 的函數原型。本論文利用 JXTA 的 Discovery Service 實作了 P2P Location Service，稱為 JXTA Location Service，後面就是描述實作的細節。

JXTA Location Service 在建構時，要傳入 Pipe Advertisement 與 JXTA SIP Peer Group 為參數。所有透過這個 Location Service 產生的 User Binding 都會包含這個 Pipe Advertisement，當其他的 JXTA SIP Peer 取得該 User Binding 後，就能藉由其中的 Pipe Advertisement 與對應的 P2P Transport 建立連線。JXTA Location Service 會利用 JXTA SIP Peer Group 拿到 Discovery Service 與 Peer Group ID。Peer Group ID 被用來產生 User Binding Advertisement 的 ID，而 Discovery Service 則是來發佈與搜尋 User Binding Advertisement。

JXTA User Binding 必須要轉成 Advertisement 的型態，才可以透過 Discovery Service 將這個資源發佈到 JXTA 同儕網路上或者從中搜尋。因此我們設計了 User Binding Advertisement，它裡面包含了一個 JXTA User Binding 並繼承了 JXTA 的 Advertisement，

讓 URI 變成可以被搜尋比對的屬性。

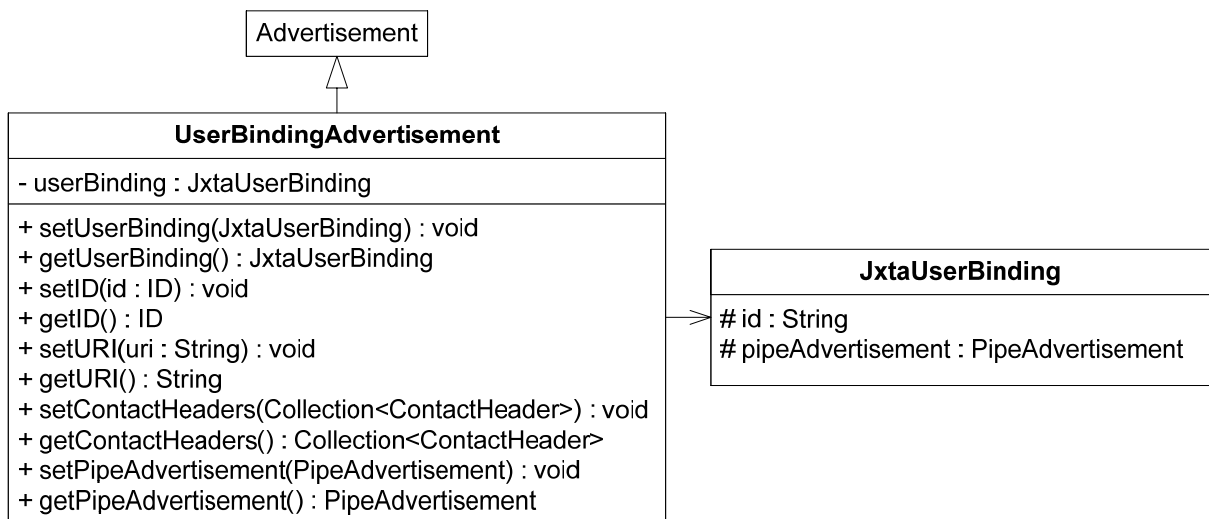


圖 16：User Binding Advertisement 類別圖

JXTA Location Service 更新本地端 UA 的 contact address 的步驟如圖 17：

1. JXTA Location Service 會先檢視是否已有這個 UA 的 User Binding 資料，若沒有則建立一個 JXTA User Binding。
2. 把 Contact header 加到 User Binding 裡，若已有相同的 contact address 存在，就會覆蓋，則 expires 等其他資訊就會更新。
3. 建立 User Binding Advertisement，利用 JXTA SIP Peer Group 的 Peer Group ID 演算出來的 ID 設為此 Advertisement 的 ID。
4. 將更新過的 User Binding 加到 User Binding Advertisement 中。
5. 透過 Discovery Service 把 User Binding Advertisement 發佈出去。
6. 檢視 JXTA User Binding 的 ID，若不為 null，表示有舊的 User Binding Advertisement 存放在本地快取(local cache)裡，利用 Discovery Service 把這個 ID 的 Advertisement 從本地快取中移除。
7. 最後，將 User Binding 裡的 ID，設定成先前發佈出去的 User Binding Advertisement 的 ID，以供下次更新時，可以依據這個 ID 清除過時的 User Binding Advertisement。

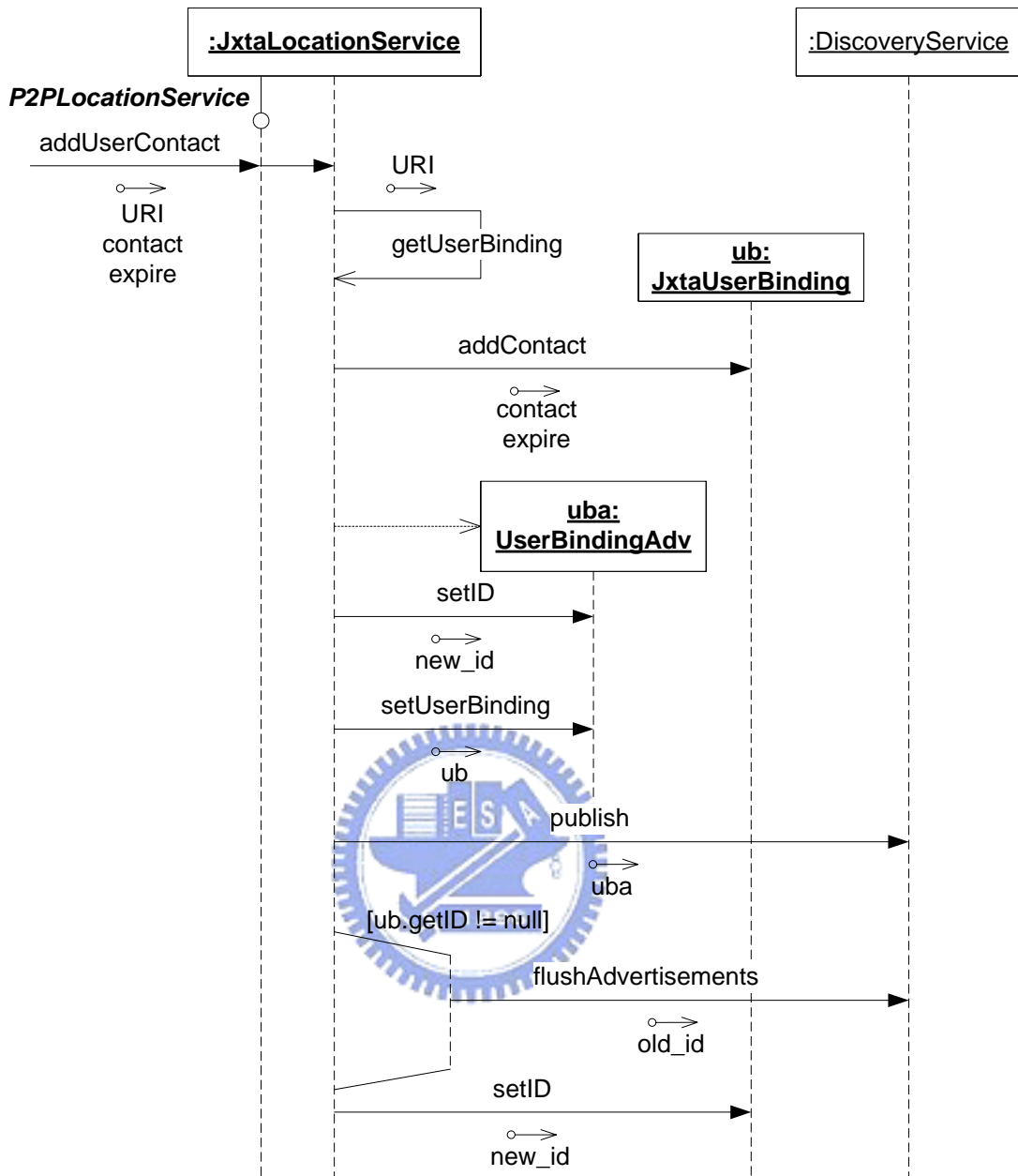


圖 17：JXTA Location Service 更新 contact address 的循序圖

透過 JXTA Location Service 查詢 User Binding 時，先檢查欲查詢的 User Binding 是否為本地端 UA，若是本地端 UA 則直接呼叫 `getUserContactURLs`，以取得未過期的 Contact Address。如果不是本地端 UA，就要執行 `get Remote User Binding` 來進行搜尋。在 `get Remote User Binding` 的設計裡面，會需要呼叫 JXTA Discovery Service 的 `get Remote Advertisements`，這個方法會透過 Rendezvous Super Peer 搜尋 advertisement。但是 `get Remote Advertisements` 不會立即回傳結果，而是非同步的方式回傳尋找到的

advertisement，為了克服這個問題，我們設計了 Query Listener，它實作了 JXTA 的 Discovery Listener，可以監聽 Discovery Service 回傳的訊息，它還提供了 get User Binding 這個方法，讓別人可以取得 Query Listener 收到的 advertisement 裡的 User Binding。我們亦在其中設計了 synchronization 的機制，get User Binding 方法會等待 Query Listener 收到回傳訊息後才會繼續執行下去，把 User Binding 回傳，否則就一直等待下去，直到 timeout。下面針對 get Remote User Binding 的步驟作說明：

1. JXTA Location Service 首先會呼叫 JXTA 的 Discovery Service 檢查想要搜尋的 advertisement 是否已在本地快取裡面。是的話，Discovery Service 會回傳 User Binding Advertisement。JXTA Location Service 再將 advertisement 裡的 User Binding 回傳就完成了。
2. 若 advertisement 並未在本地快取中，則 JXTA Location Service 會先建立一個 Query Listener，接著呼叫 Discovery Service 的 get Remote Advertisements，並把 Query Listener 當成參數傳入，然後再呼叫 Query Listener 的 get User Binding。
3. 因為有 synchronization 的機制，Query Listener 的 get User Binding 會讓目前的執行緒停住(wait)，在收到任何回傳訊息前這個方法都不會執行下去。
4. 等到 Query Listener 收到了剛才查詢的結果後，會通知(notify)暫停的執行緒，繼續完成 get User Binding 的動作，取得欲找的 User Binding 並回傳。

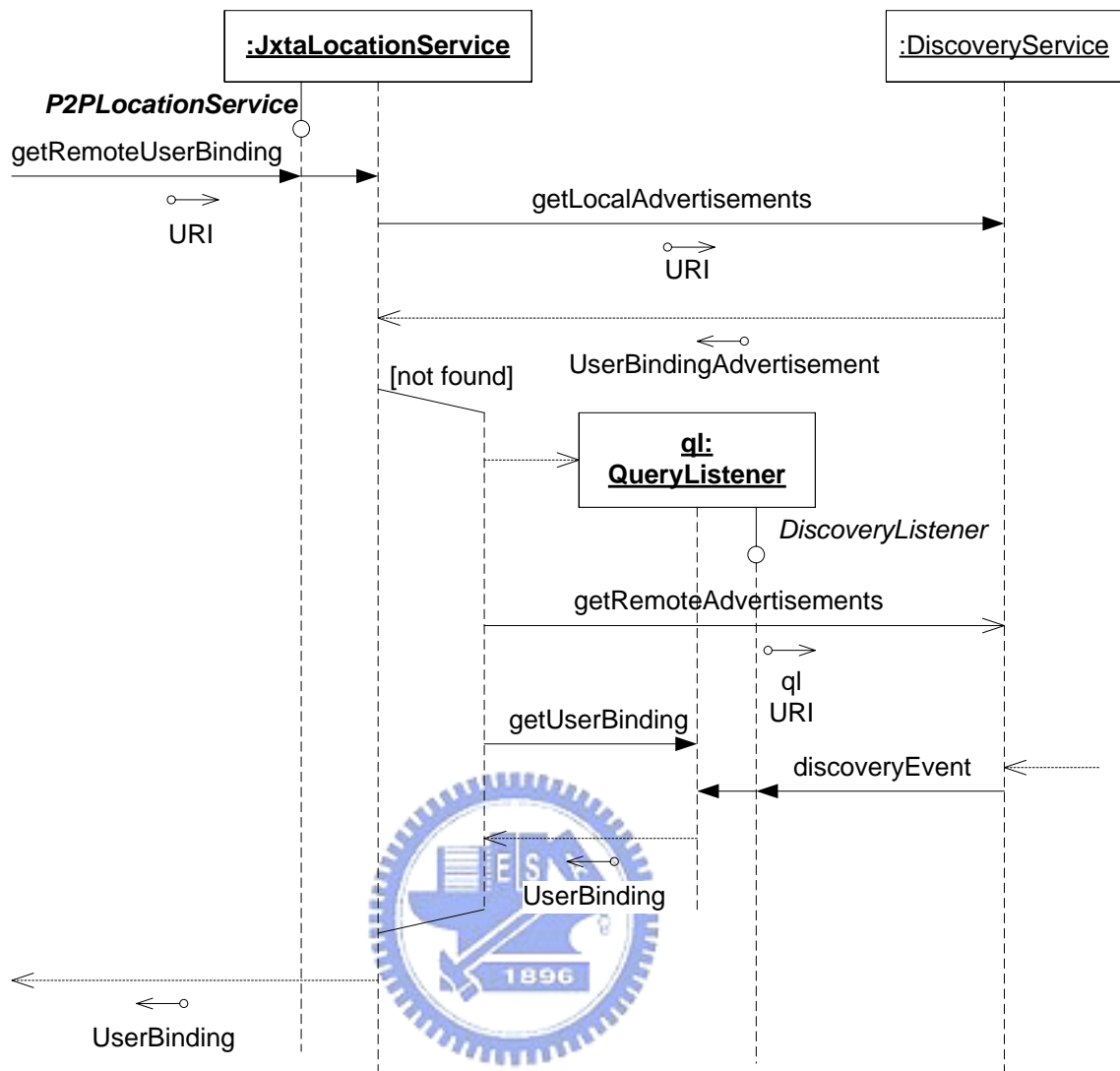


圖 18：JXTA Location Service 搜尋遠端 User Binding 的循序圖

3.3.3.4 Peer-to-Peer Transport

P2P Transport 是透過同儕網路傳輸訊息的一個介面，呼叫 send Message 方法就可以將訊息送出。要接收經由 P2P Transport 收到的訊息，就要實作 P2P Transport Listener 介面，並且設定成 P2P Transport 的 listener，當 P2P Transport 收到訊息時，會呼叫 listener 的 message Received 方法，把訊息傳給上層。Proxy 要實作 P2P Transport Listener 且使用 P2P Transport 來達到 message forwarding 的功能。

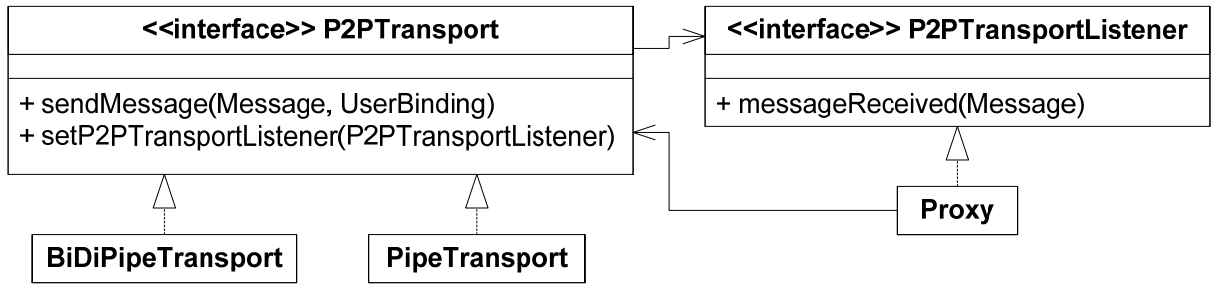


圖 19：P2P Transport 與 P2P Transport Listener 相關類別圖

本論文設計了兩種不同的類別來實作 P2P Transport。首先介紹 Pipe Transport 的建立與傳輸訊息的過程：

1. Pipe Transport 的建構子會有 Pipe Advertisement 與 Pipe Service 兩個參數，它利用 Pipe Advertisement 與 Pipe Service 建立一個 Input Pipe 來等待訊息的傳入。
2. JXTA Location Service 則會把該 Pipe Advertisement 放入 JXTA User Binding，再包成 User Binding Advertisement 並發佈出去。
3. 其他的 JXTA SIP Peer 取得此 User Binding Advertisement 後，會把裡面的 User Binding 和想要傳送的訊息透過 P2P Transport 的 send Message 一起交給 Pipe Transport 處理。
4. 這個 Pipe Transport 拿出 JXTA User Binding 裡的 Pipe Advertisement，再利用 Pipe Service 建立對應的 Output Pipe。
5. 接著用 Message Converter 把 SIP 訊息轉換成 JXTA 訊息從 Output Pipe 送出。
6. 則先前的 Input Pipe 並會收到這個 JXTA 訊息，並傳送給 Pipe Msg Listener，也就是原本的 Pipe Transport。
7. 最後，該 Pipe Transport 使用 Message Converter 把 JXTA 訊息轉換回原來的 SIP 訊息，再呼叫 message Received 把 SIP 訊息傳給 P2P Transport Listener。

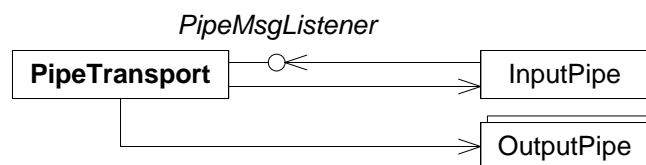


圖 20：Pipe Transport 類別圖

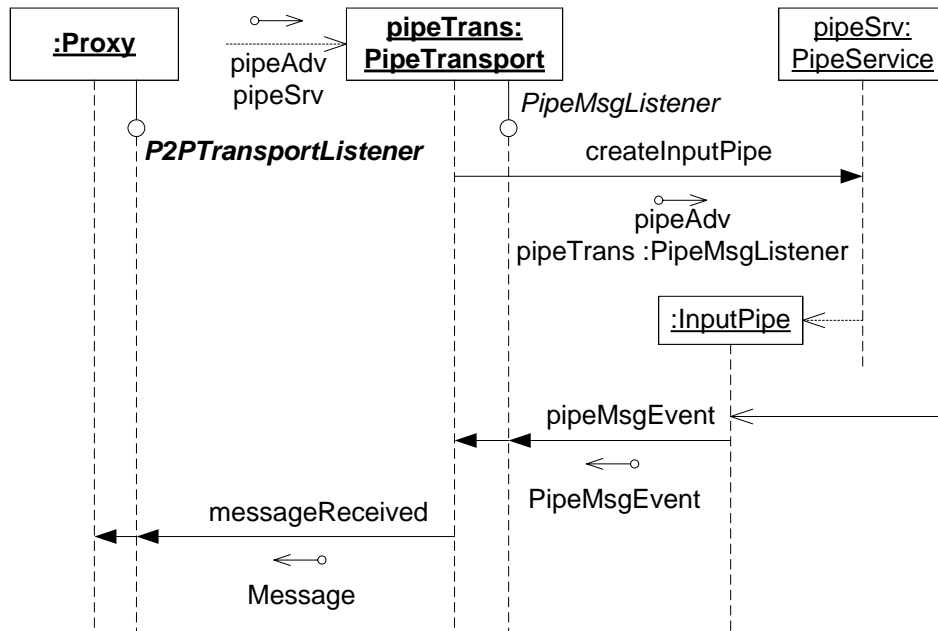


圖 21：Pipe Transport 的建立與接收訊息的循序圖

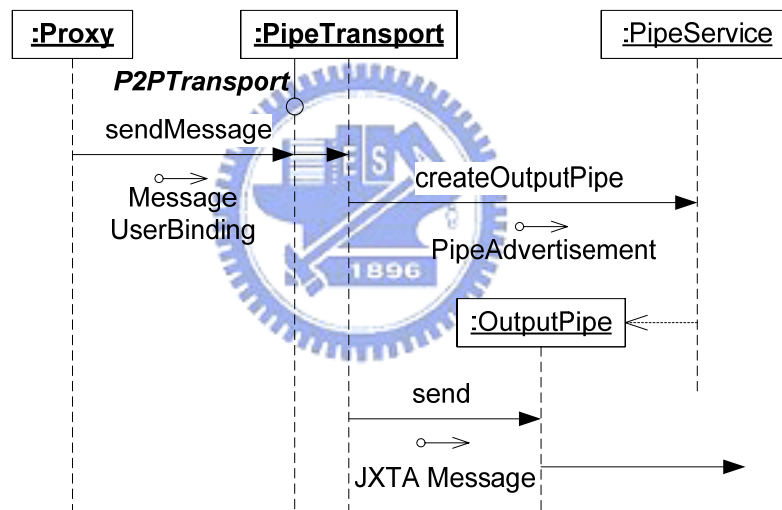


圖 22：Pipe Transport 傳送訊息的循序圖

由於 JXTA 的 Input Pipe 與 Output Pipe 類似 UDP 的機制無法保證訊息傳遞的可靠性，為了提供可靠的訊息傳遞，本研究利用 JxtaBiDiPipe 實作了另一個類別 BiDi Pipe Transport。在建立 JxtaBiDiPipe 時能設定它是否為 reliable，若設定為是，則 JxtaBiDiPipe 能保證訊息傳遞的可靠性，使 BiDi Pipe Transport 可以提供可靠的傳輸連線。BiDi Pipe Transport 會建立 BiDiPipe Server 來監聽是否有 Peer 要求建立 JxtaBiDiPipe，若有的話則利用 BiDiPipe Server Listener 的介面通知 BiDi Pipe Transport 取得該 JxtaBiDiPipe，BiDi Pipe Transport 還會實作 Pipe Msg Listener 來接收從 JxtaBiDiPipe 傳

來的訊息。要利用 BiDi Pipe Transport 傳送訊息時，它會先建立 JxtaBiDiPipe，然後透過 JxtaBiDiPipe 把訊息送出。接著圖 24 與圖 25 描述 BiDi Pipe Transport 運作的細節。

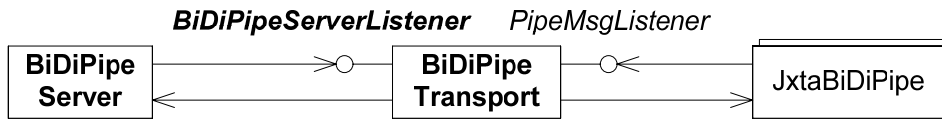


圖 23：BiDi Pipe Transport 相關類別圖

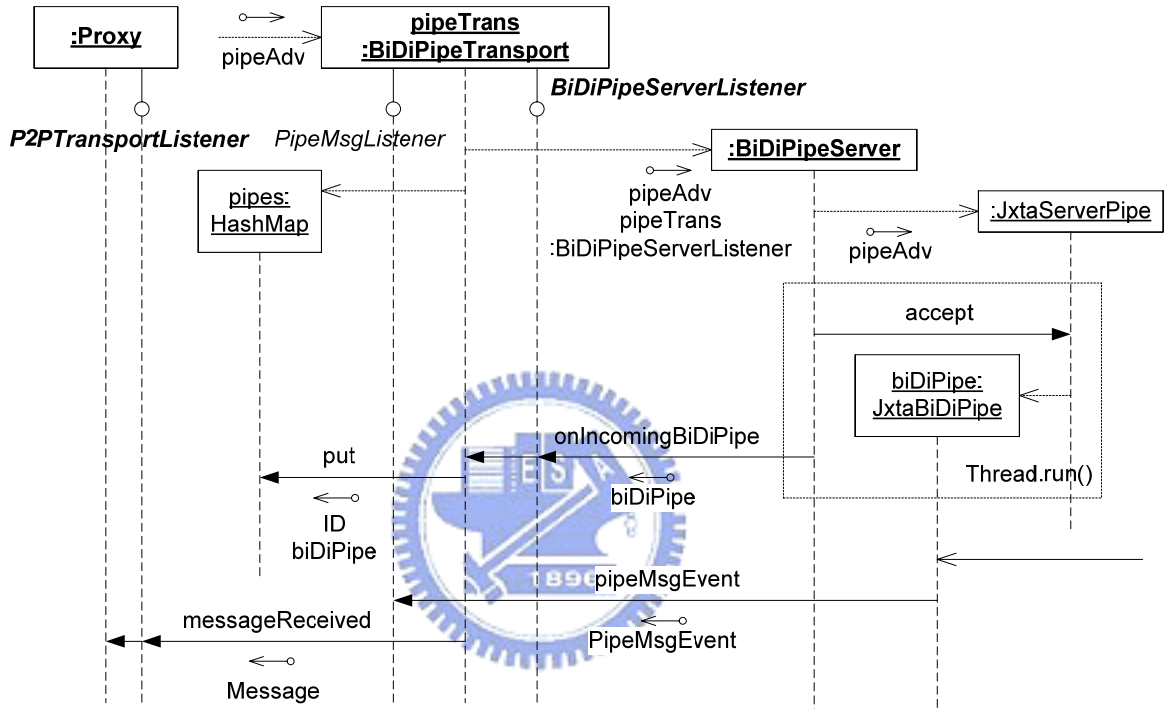


圖 24：BiDi Pipe Transport 的建立與接收訊息的循序圖

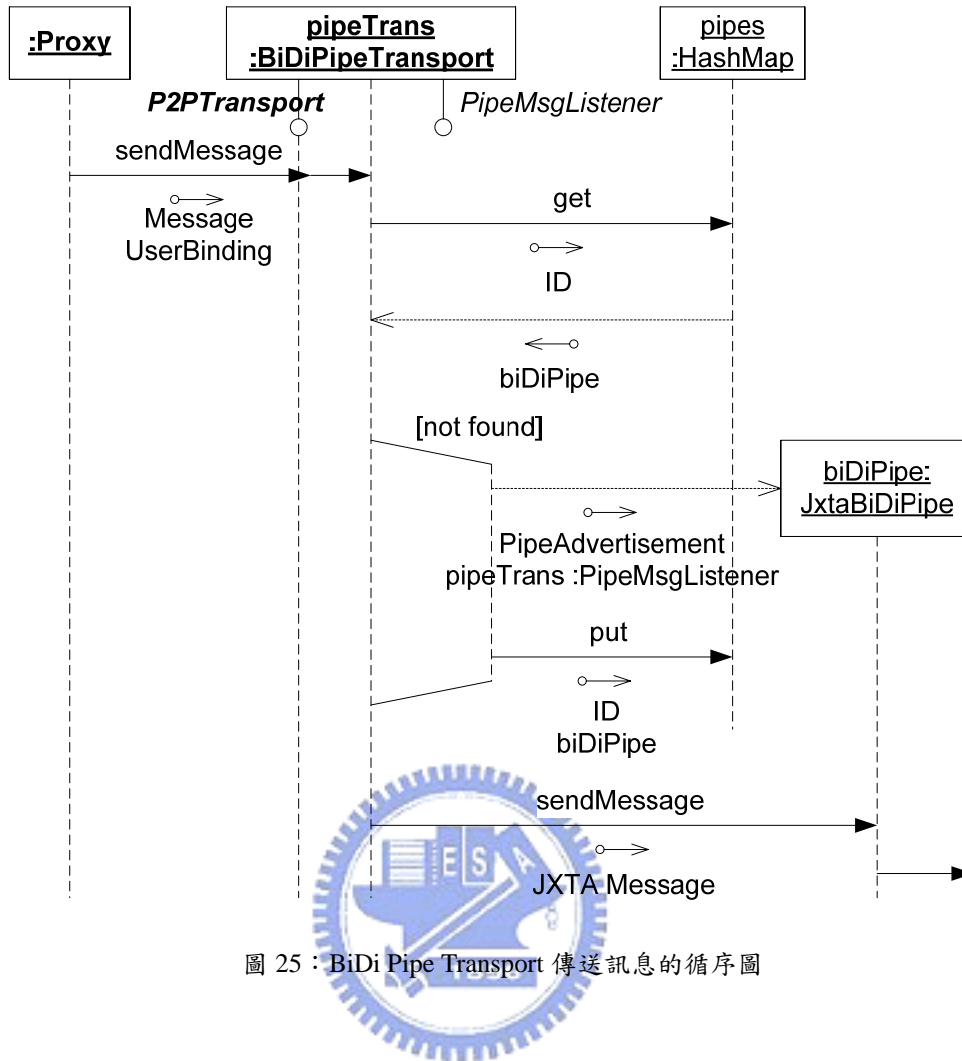


圖 25：BiDi Pipe Transport 傳送訊息的循序圖

3.3.3.5 Message Conversion

SIP 訊息必須轉換成 JXTA 訊息，才能透過 JXTA 的 Pipe 傳遞。我們將整個 SIP 訊息視為單一字串，把這個字串當作參數用來產生 JXTA String Message Element，再把 Message Element 放入一個新建的 JXTA 訊息中，就能夠藉由 Pipe 傳遞了。

而收到 JXTA 訊息的那一方，只要利用事先定義好的 name space 與 name 把存有 SIP 訊息的 JXTA String Message Element 從 JXTA 訊息中取出，再用裡面的字串產生 SIP 訊息就能恢復成原本的 SIP 訊息。

3.3.4 SIP Service Finder

SIP Service Finder 啟動後，會在 JXTA Net Peer Group 搜尋 JXTA SIP Service

Advertisement，一旦找到後，就可以取得 JXTA SIP Peer 的資訊，最後根據這些資訊設定上層 UA 的 outbound proxy，則該 UA 就可以透過其他的 JXTA SIP Peer 取得 SIP Server 的服務。



Chapter 4 Operation

在本論文的設計中，可以使用任何支援 RFC3261 的 UA 與我們的 JXTA SIP Peer 溝通，亦可利用 JXTA SIP Peer 自動產生的 MjSip UA，對整個 JXTA SIP Peer 的運作方式都是相同的。而且無論 UA 與 JXTA SIP Peer 是在相同的主機上還是各別運行在不同主機裡，也都不會有影響。因此後面的探討將不會特別針對何種 UA，並假設 UA 與 JXTA SIP Peer 跑在不同的主機上。

在下面運作中，皆假設 UA 已將 JXTA SIP Peer 設定為 outbound proxy，若是使用 JXTA SIP 自行建立的 UA，則 outbound proxy 會自動設定完成。

4.1 Registration

UA 向 JXTA SIP Peer 進行註冊的動作如下(見圖 26)：

1. UA 先向它的 outbound proxy 即 JXTA SIP Peer 發出註冊要求。
2. JXTA SIP Peer 的 SIP 層收到後註冊要求後，會依據 contact address 等資訊產生 JXTA User Binding。然後由 P2P 層產生 User Binding Advertisement 儲存在本地快取裡，並把公告索引發佈到 Rendezvous 網路上。
3. 最後再由 JXTA SIP Peer 的 SIP 層回傳 200 OK 的訊息給 UA，就完成了註冊的動作。

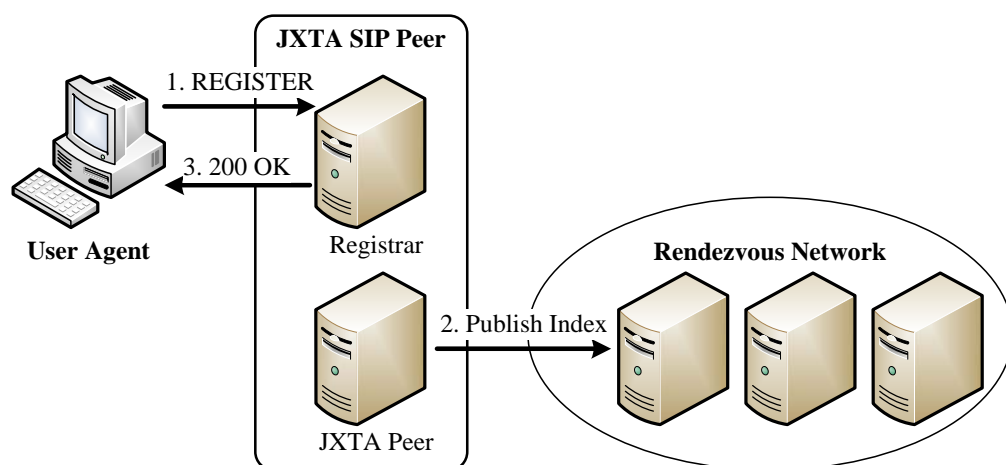


圖 26：Registration 的運作方式

4.2 Proxy: Message Forwarding

UA 利用 JXTA SIP Peer 的 Proxy 收發訊息的過程如下：

1. UA A 發出要求訊息給 JXTA SIP Peer A。
2. JXTA SIP Peer A 根據 Request-URI 裡的 SIP URI 去搜尋符合的 User Binding Advertisement，先檢查本地快取，若沒有則向 Rendezvous Super-Peer 發出搜尋要求 (discovery request)。

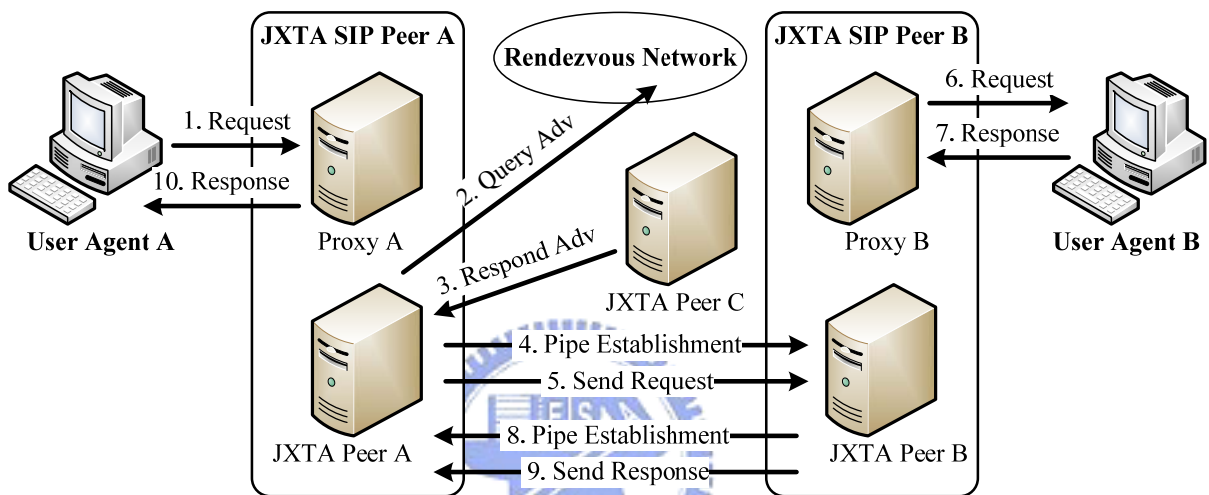


圖 27：Proxy: Message Forwarding

3. 擁有該 User Binding Advertisement 的 JXTA Peer C 會把公告傳回給 JXTA SIP Peer A。
4. 接著 JXTA SIP Peer A 利用 JXTA User Binding 裡的 Pipe Advertisement，與 UA B 的 outbound proxy JXTA SIP Peer B 建立 pipe 連線。JXTA SIP Peer B 可能與 JXTA Peer C 為同一個 Peer。
5. 透過建立好的 pipe 把要求訊息送給 JXTA SIP Peer B。
6. JXTA SIP Peer B 會查詢本身的 P2P Location Server，得知 UA B 的位置後，將要求訊息傳遞給它。
7. UA B 將回應訊息傳送給它的 outbound proxy 即 JXTA SIP Peer B。
8. 如同步驟 4，JXTA SIP Peer B 會利用 JXTA User Binding 建立一個 pipe 連到 JXTA SIP Peer A。
9. 接著 JXTA SIP Peer B 利用 pipe 將回應訊息傳送給 JXTA SIP Peer A。

10. 最後，JXTA SIP Peer A 遞送該回應訊息給要求發送者 UAA。

4.3 Redirection

JXTA SIP Peer 進行 redirection 的動作如下：

1. UA 傳送要求訊息給 JXTA SIP Peer。
2. JXTA SIP Peer 收到後，根據 Request-URI 裡的 SIP URI 去搜尋符合的 User Binding Advertisement，先檢查本地快取，若沒有則向 Rendezvous Super-Peer 發出搜尋要求 (discovery request)。
3. 擁有該 User Binding Advertisement 的 JXTA Peer 會把公告傳回給 JXTA SIP Peer。
4. JXTA SIP Peer 會從 JXTA User Binding 取出 contact address，用來產生 302 Moved Temporarily 的回應訊息回覆給 UA，告訴 UA 應該把要求訊息傳遞到哪裡。

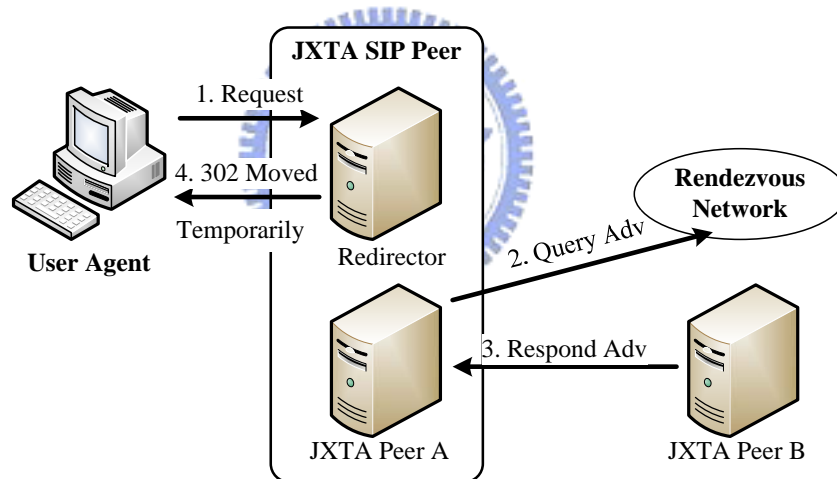


圖 28：Redirection 運作方式

4.4 Multiple Domain

JXTA SIP Peer 上的各種 SIP 伺服器，不會限制 UA 的網域是否要一致，可以讓 UA 使用任意的網域名稱進行註冊。只要是有透過 JXTA SIP Peer 註冊的 UA，它的 binding 就能夠被任何 JXTA SIP Peer 查詢到。因此無需任何的修改與設定，我們的 JXTA SIP Peer 可以讓各種不同網域名稱的 UA 使用。

4.5 Interoperability

4.5.1 Interoperation with Conventional SIP Domain

要讓 JXTA SIP Peer 形成的 P2P SIP 網域裡的 UA 與其他 SIP 網域的 UA 互通，需要如下的設定。JXTA SIP Domain 裡所有 UA 的網域名稱都要向 DNS 登錄，並且跟 DNS 設定某幾個 JXTA SIP Peer 會提供這些網域的 SIP 服務，這些 Peer 稱為 JXTA SIP Proxy Peer。建議的做法是直接限制 JXTA SIP Domain 中所能註冊的網域名稱，將允許被註冊的幾個網域名稱設定為全域的系統參數，由此 JXTA SIP Peer 可以分辨哪些網域不在它的管理範圍內。有了這樣的設定後，當一般 SIP Domain 的 UA X 要送訊息給 JXTA SIP Domain 的 UA Y 時，UA X 的 Proxy 會向 DNS 查詢負責 UA Y 所屬網域的 SIP 伺服器，就會查到了 JXTA SIP Proxy Peer。JXTA SIP Proxy Peer 再透過 P2P Location Service 查到負責 UA Y 的 JXTA SIP Peer，最後就可以傳訊息傳給 UA Y。相反的，當 UA Y 要傳訊息給 UA X 時，JXTA SIP Peer 會發現 UA X 的網域並不在 JXTA SIP Domain 裡面，因此不會向 P2P Location Service 查詢，而是直接以一般 SIP 的方式，利用 DNS 找出對方的 Proxy，然後傳遞訊息。如此一來 JXTA SIP Domain 的 UA 就能夠與其他 SIP 網域的 UA 互傳訊息，進而建立會話連線了。

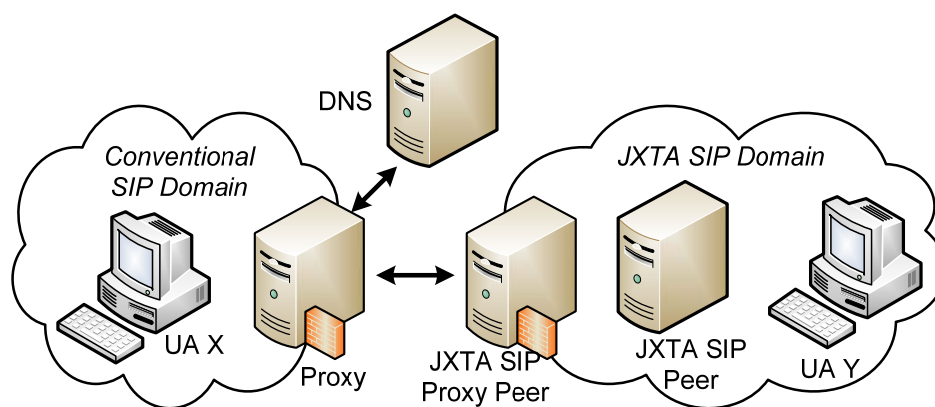


圖 29：Interoperation with Conventional SIP Domain

4.5.2 Interoperation with other P2PSIP Domain

只要其他 P2P SIP Domain 能夠支援與一般的 SIP Domain 互通的能力，則運用前一節敘述的方法，我們的 JXTA SIP Domain 就可以透過相同的方式與該 P2P SIP Domain 互相支援、傳遞訊息了。



Chapter 5 Experiment

5.1 Environment Setup

我們在五台電腦上執行 Wireshark¹，用來記錄及分析實驗中所有真實的網路封包傳輸。接著在其中三台主機上跑起 JXTA SIP Peer，提供 Registrar 與 Proxy 的功能，並且組成一個 JXTA SIP Peer Group。最後在另外兩台電腦上執行 MjSip 的 UA，一台將 Address of Record 設定為 alice@alpha.net(以下皆稱為 Alice)，並把 outbound proxy 設定成 JXTA SIP Peer A，另一台將 AoR 設為 bob@beta.net(以下皆稱為 Bob)，outbound proxy 則設定 JXTA SIP Peer B。並且把兩個 UA 都設定成自動接聽。

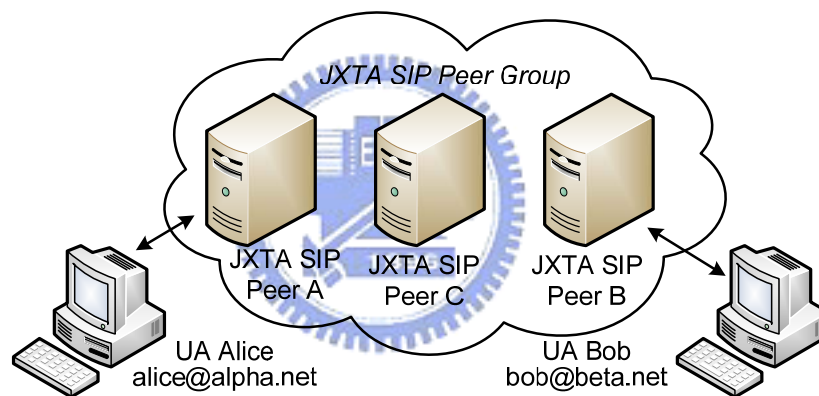


圖 30：實驗配置圖

實驗結束後，我們用 JXTA 的 DumpCm 擷取 JXTA SIP Peer 本地快取裡的 User Binding Advertisement 以及公告索引等資訊。

5.2 Results

我們測試了三項主要的 SIP 功能，分別為註冊、會話建立(session setup)以及會話中止(session termination)。利用 Wireshark 記錄到的 SIP 與 JXTA 訊息都會收錄在 Appendix B 裡，接下來的部份只會做敘述性的說明。

¹ Wireshark 是一個 network protocol analyzer，能夠記錄及分析實際的網路封包，<http://www.wireshark.org/>。

5.2.1 Registration

Alice 與 Bob 都會向他們的 outbound proxy 發出註冊的要求，而 JXTA SIP Peer 最後會將 User Binding Advertisement 存到本地快取中，並且把公告索引發佈到 Rendezvous Super-Peer 上，下面為 JXTA SIP Peer A 處理 Alice 的註冊時儲存在本地快取的 User Binding Advertisement，這裡只節錄了主要的部份，完整的內容請見 Appendix A。

```
1030
Adv/uuid-5EB1105307B9454A8DAEA5C7CF2AD75CF56D4F7C608E4EA2BA0BEA59F2B6D4
5401
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxtasip:UserBindingAdvertisement>
<jxtasip:UserBindingAdvertisement xmlns:jxta="http://jxta.org">
  <id>
    urn:jxta:uuid-5EB1105307B9454A8DAEA5C7CF2AD75CF56D4F7C608E4EA2B
    A0BEA59F2B6D45401
  </id>
  <name>
    alice@alpha.net
  </name>
  <contact>
    Contact: <sip:alice@140.113.216.104>;expires="Wed, 16 Jul 2008
    23:04:03 GMT"
  </contact>
  [truncated]
</jxtasip:UserBindingAdvertisement>
```

接著是 Rendezvous Super-Peer 所記錄的公告索引，下面為 Alice 的 User Binding Advertisement 的公告索引：

```
1028 Advnamealice@alpha.net
Advnamealice@alpha.net
Purn:jxta:uuid-59616261646162614E5047205032503352646D2D77D649D5AC0BE4B3
C8F83FE103,
Purn:jxta:uuid-59616261646162614E50472050325033D2F8AC3A00BA4CE4BE05A898
0C12251003
```

至此 Alice 的註冊動作就完成了。

5.2.2 Session Setup

Alice 與 Bob 都註冊完畢後，我們用 Bob 的 UA 向 Alice 發出邀請。Bob 的 outbound proxy JXTA SIP Peer B 透過 Rendezvous Super-Peer 順利找到 Alice 的 User Binding Advertisement，用其中的 Pipe Advertisement 與 JXTA SIP Peer A 建立起 Pipe，並且將接下來的 SIP 訊息都包成 JXTA 訊息從 Pipe 傳輸。Alice 的 UA 收到邀請後傳回 100 Trying、

180 Ringing 等訊息給 Bob，然後因為是自動接聽就會接著傳回 200 OK。Bob 從 JXTA SIP Peer B 收到 200 OK 的訊息後，會再發出 ACK 的訊息給 Alice。最後 Alice 收到該訊息，Bob 與 Alice 的 SDP 交換完成，兩邊 UA 的會話就建立起來了。

5.2.3 Session Termination

Alice 與 Bob 的會話建立完成後，我們讓 Alice 的 UA 結束會話，發出 BYE 的訊息給 Bob。經由 JXTA SIP Peer A 與 JXTA SIP Peer B 的傳送，Bob 收到該訊息並回傳 200 OK。最後 Alice 收到 200 OK，雙方的 UA 結束會話連線。



Chapter 6 Conclusion

6.1 Accomplishment

本研究的主要貢獻有下列兩點：

- 我們成功地將 P2P SIP 服務實現在一個 general-purpose 的 P2P 平台 JXTA 上，利用 Rendezvous Super-Peer 網路來達到 Location Service 的功能，並讓 JXTA 的 peer 能夠運用這個 Location Service 以提供 SIP Registrar、Proxy、Redirector 等服務，以及設計出 P2P SIP 同儕節點的運作模式。
- 我們也將實作出來的成果申請加入 JXTA™ Community Projects¹中，讓更多人能有機會利用本論文的研究成果，繼續進行研究與發展。

6.2 Future Work

本研究中仍然少數的設計尚未實作完成，如下所示：

- SIP Service Finder 在本研究中尚未被實現出來，但它的設計概念已非常地明確，未來可以將考慮實作這個功能。

此外，還有一些未來可行的研究方向，能使本研究更趨完整，敘述如下：

- 本論文並沒有實現 AAA(Authentication, Authorization, and Accounting)的機制，未來可以將 Halo[15]的設計實現在我們的系統中。
- 還有一些 SIP 網路單元並未在此論文中實現，如 Presence Server 以及 PSTN 閘道 (public switched telephone network gateway)等，都可以繼續進行實作。

最後，本研究申請加入 JXTA™ Community Projects 後，仍會持續地更新，以期能達到最完善的成果。

¹ jxta: JXTA™ Community Projects, <https://jxta.dev.java.net/>.

Reference

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session Initiation Protocol,” RFC 3261, June 2002.
- [2] J. Rosenberg and H. Schulzrinne, “Session Initiation Protocol (SIP): Locating SIP Servers,” RFC 3263, June 2002.
- [3] D. Bryan, P. Matthews, E. Shim, and D. Willis, “Concepts and Terminology for Peer to Peer SIP,” Internet Draft draft-ietf-p2psip-concepts-01 (work in progress), November 2007.
- [4] J. Shi, Y. Ji, H. Zhang, and Y. Li, “A Hierarchical P2P-SIP Architecture,” Internet Draft draft-shi-p2psip-hier-arch-00, August 2006.
- [5] E. Shim, S. Narayanan, and G. Daley, “An Architecture for Peer-to-Peer Session Initiation Protocol (P2P SIP),” Internet Draft draft-shim-sipping-p2p-arch-00, February 2006.
- [6] D. Bryan, B. Lowekamp, and C. Jennings, “dSIP: A P2P Approach to SIP Registration and Resource Location,” Internet Draft draft-bryan-p2psip-dsip-00, February 2007.
- [7] C. Jennings, B. Lowekamp, E. Rescorla, J. Rosenberg, S. Baset, and H. Schulzrinne, “REsource LOcation And Discovery (RELOAD),” Internet Draft draft-bryan-p2psip-reload-03 (working in progress), February 2008.
- [8] M. Zangrilli and D. Bryan, “A Chord-based DHT for Resource Lookup in P2PSIP,” Internet Draft draft-zangrilli-p2psip-dsip-dhtchord-00, February 2007.
- [9] M. Zangrilli and D. Bryan, “A Bamboo-based DHT for Resource Lookup in P2PSIP,” Internet Draft draft-zangrilli-p2psip-dsip-dhtbamboo-00, February 2007.
- [10] jxta: JXTA™ Community Projects, <https://jxta.dev.java.net/>.

-
- [11] B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J.C. Hugly, E. Pouyoul, and B. Yeager, “Project JXTA 2.0 Super-Peer Virtual Network,” Sun Microsystems, Inc., May 2003.
- [12] B. Traversat, M. Abdelaziz, and E. Pouyoul, “Project JXTA: A Loosely-Consistent DHT Rendezvous Walker,” Sun Microsystems, Inc.
- [13] MjSip, <http://www.mjsip.org/>.
- [14] L. Veltri, “MjSip-Mini-Tutorial,” Department of Information Engineering at University of Parma, April 2005.
- [15] Fu-Kuo Tseng, “Halo: A Hierarchical Identity-Based Public Key Infrastructure for Peer-to-Peer Opportunistic Collaboration,” Institute of Computer Science and Engineering at National Chiao Tung University, June 2008.



Appendix A

JXTA SIP Peer 處理註冊要求時，儲存在本地快取中的 User Binding Advertisement：

```
1030
Adv/uuid-5EB1105307B9454A8DAEA5C7CF2AD75CF56D4F7C608E4EA2BA0BEA59F2B6D4
5401
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxtasip:UserBindingAdvertisement>
<jxtasip:UserBindingAdvertisement xmlns:jxta="http://jxta.org">
  <id>
    urn:jxta:uuid-5EB1105307B9454A8DAEA5C7CF2AD75CF56D4F7C608E4EA2BA0BE
    A59F2B6D45401
  </id>
  <name>
    alice@alpha.net
  </name>
  <contact>
    Contact: &lt;sip:alice@140.113.216.104>;expires="Wed, 16 Jul 2008
    23:04:03 GMT"
  </contact>
  <bidipipeadv>
    &lt;?xml version="1.0" encoding="UTF-8"?>
    &lt;!DOCTYPE jxta:PipeAdvertisement>
    &lt;jxta:PipeAdvertisement xml:space="preserve"
    xmlns:jxta="http://jxta.org">
    &lt;Id>urn:jxta:uuid-5EB1105307B9454A8DAEA5C7CF2AD75CD5A6AD46BF7C4DEAB1
    90B01C5A50484404&lt;/Id>&lt;Type>JxtaUnicast&lt;/Type>&lt;Name>JXTA SIP
    BiDiPipe&lt;/Name>&lt;/jxta:PipeAdvertisement>
  </bidipipeadv>
  <inputpipeadv>
    &lt;?xml version="1.0" encoding="UTF-8"?>
    &lt;!DOCTYPE jxta:PipeAdvertisement>
    &lt;jxta:PipeAdvertisement xml:space="preserve"
    xmlns:jxta="http://jxta.org">
    &lt;Id>urn:jxta:uuid-5EB1105307B9454A8DAEA5C7CF2AD75C4BFECDD15E854684B9
    ACBB19F134ACE904&lt;/Id>&lt;Type>JxtaUnicast&lt;/Type>&lt;Name>JXTA SIP
    UniPipe&lt;/Name>&lt;/jxta:PipeAdvertisement>
  </inputpipeadv>
</jxtasip:UserBindingAdvertisement>
```

Appendix B

在下列的實驗結果中，為了讓訊息的簡潔，我們可能會省略 SIP 訊息的 body 與對應的

Content-Length、Content-Type header 欄位。

Registration

Alice 在 User Agent 啟動時向 Proxy A 註冊。

```
REGISTER sip:alpha.net SIP/2.0
Via: SIP/2.0/UDP 140.113.216.104:5060;rport;branch=z9hG4bK91295
Max-Forwards: 70
To: "Alice" <sip:alice@alpha.net>
From: "Alice" <sip:alice@alpha.net>;tag=z9hG4bK42295646
Call-ID: 121784906138@140.113.216.104
CSeq: 1 REGISTER
Contact: <sip:alice@140.113.216.104>
Expires: 3600
User-Agent: mjsip stack 1.6
Content-Length: 0
```

JXTA SIP Peer A 收到回應 200 OK 給 Alice。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 140.113.216.104:5060;branch=z9hG4bK91295;rport=5060
To: "Alice" <sip:alice@alpha.net>
From: "Alice" <sip:alice@alpha.net>;tag=z9hG4bK42295646
Call-ID: 121784906138@140.113.216.104
CSeq: 1 REGISTER
Server: JXTA SIP
Contact: <sip:alice@140.113.216.104>;expires=3600
Content-Length: 0
```

接著 JXTA SIP Peer A 會產立 User Binding Advertisement，並發佈出去。

Session Setup

Bob 向 alice@alpha.net 發出邀請。

```
INVITE sip:alice@alpha.net SIP/2.0
Via: SIP/2.0/UDP 140.113.216.114:5060;rport;branch=z9hG4bK18756
Max-Forwards: 70
To: <sip:alice@alpha.net>
From: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
Call-ID: 440069755637@140.113.216.114
CSeq: 1 INVITE
Contact: <sip:bob@140.113.216.114>
Expires: 3600
User-Agent: mjsip stack 1.6
Content-Length: 159
Content-Type: application/sdp
```

```
v=0
o="Bob" <sip:bob@beta.net> 0 0 IN IP4 140.113.216.114
```

```
s=Session SIP/SDP
c=IN IP4 140.113.216.114
t=0 0
m=audio 21000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

JXTA SIP Peer B 收到 Bob 的要求，透過 Pipe 傳送給 JXTA SIP Peer A，它會收到以下的訊息，並傳送給 Alice。

```
INVITE sip:alice@140.113.216.104 SIP/2.0
Via: SIP/2.0/UDP 140.113.216.114:5060;branch=z9hG4bK18756;rport=5060
Max-Forwards: 70
To: <sip:alice@alpha.net>
From: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
Call-ID: 440069755637@140.113.216.114
CSeq: 1 INVITE
Contact: <sip:bob@140.113.216.114>
Expires: 3600
User-Agent: mjsip stack 1.6
Loop-Tag: d629edbe1477dc
Content-Length: 159
Content-Type: application/sdp
```

```
v=0
o="Bob" <sip:bob@beta.net> 0 0 IN IP4 140.113.216.114
s=Session SIP/SDP
c=IN IP4 140.113.216.114
t=0 0
m=audio 21000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Alice 的 UA 收到後，發出 100 Trying 的訊息給 JXTA SIP Peer A

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 140.113.216.114:5060;branch=z9hG4bK18756;
received=140.113.216.119;rport=5060
To: <sip:alice@alpha.net>
From: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
Call-ID: 440069755637@140.113.216.114
CSeq: 1 INVITE
Server: mjsip stack 1.6
Content-Length: 0
```

JXTA SIP Peer A 收到 Alice 的回應後，透過 Pipe 傳回 JXTA SIP Peer B，JXTA SIP Peer B

收到直接傳回給 Bob。接著 Alice 的 UA 會發出 180 Ringing 的訊息給 Proxy A，並且開始響鈴。

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 140.113.216.114:5060;branch=z9hG4bK18756;
received=140.113.216.119;rport=5060
To: <sip:alice@alpha.net>
From: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
Call-ID: 440069755637@140.113.216.114
CSeq: 1 INVITE
Server: mjsip stack 1.6
Content-Length: 0
```

同樣，JXTA SIP Peer A 收到後經由 Pipe 傳給 JXTA SIP Peer B，而 JXTA SIP Peer B 再回傳給 Bob，此時 Bob 的 UA 也會進入響鈴的模式。然後 Alice 接聽後，Alice 的 UA 就會傳送 200 OK 的訊息給 JXTA SIP Peer A。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 140.113.216.114:5060;branch=z9hG4bK18756;
received=140.113.216.119;rport=5060
To: <sip:alice@alpha.net>;tag=afee2211bc7a4b3b
From: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
Call-ID: 440069755637@140.113.216.114
CSeq: 1 INVITE
Contact: <sip:alice@140.113.216.104>
Server: mjsip stack 1.6
Content-Length: 159
Content-Type: application/sdp
```

```
v=0
o="Bob" <sip:bob@beta.net> 0 0 IN IP4 140.113.216.114
s=Session SIP/SDP
c=IN IP4 140.113.216.104
t=0 0
m=audio 21000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

JXTA SIP Peer A 同樣透過 Pipe 把訊息傳回給 JXTA SIP Peer B，JXTA SIP Peer B 再傳給 Bob。Bob 的 UA 收到後就會回傳 ACK 給 JXTA SIP Peer B。

```
ACK sip:alice@140.113.216.104 SIP/2.0
Via: SIP/2.0/UDP 140.113.216.114:5060;rport;branch=z9hG4bK71609
Max-Forwards: 70
To: <sip:alice@alpha.net>;tag=afee2211bc7a4b3b
From: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
Call-ID: 440069755637@140.113.216.114
CSeq: 1 ACK
Contact: <sip:bob@140.113.216.114>
Expires: 3600
User-Agent: mjsip stack 1.6
Content-Length: 0
```

JXTA SIP Peer B 藉由 Pipe 傳給 JXTA SIP Peer A，最後 Alice 的 UA 收到後，Bob 與 Alice 之間的 media session 就建立完成了。

Session Termination

延續剛才 Alice 與 Bob 的連線，此時 Alice 掛斷，Alice 的 UA 會發出 BYE 給 Bob。JXTA SIP Peer A 先收到這個訊息，經由 JXTA SIP Peer B 最後傳給了 Bob。

```
BYE sip:bob@140.113.216.114 SIP/2.0
Via: SIP/2.0/UDP 140.113.216.104:5060;branch=z9hG4bK84558;rport=5060
Max-Forwards: 70
To: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
From: <sip:alice@alpha.net>;tag=afee2211bc7a4b3b
```

Call-ID: 440069755637@140.113.216.114
CSeq: 1 BYE
User-Agent: mjsip stack 1.6
Loop-Tag: 3242267e52e0f1
Content-Length: 0

Bob 的 UA 收到後會回應 200 OK 的訊息。

SIP/2.0 200 OK
Via: SIP/2.0/UDP 140.113.216.104:5060;branch=z9hG4bK84558;
received=140.113.216.104;rport=5060
To: "Bob" <sip:bob@beta.net>;tag=z9hG4bK20581794
From: <sip:alice@alpha.net>;tag=afee2211bc7a4b3b
Call-ID: 440069755637@140.113.216.114
CSeq: 1 BYE
Server: mjsip stack 1.6
Content-Length: 0

當 Alice 的 UA 收到這個訊息後，兩者原本的連線就結束了。

