

同儕式行動代理人檔案推播平台

P2P-like File Push System Using Mobile Agents

游伯恩

王宗一

鍾哲民

n96931267@nckualumni.org.tw

成功大學工程科學系

wti535@mail.ncku.edu.tw

Adam145love@yahoo.com.tw

摘要

寬頻網路時代的來臨，傳統的推播系統專注於電子報、RSS(Rich Site Summary)等小資料量的資訊傳遞，高資料量的推播系統並不多見。

目前網路中大流量的檔案傳輸以同儕式檔案傳輸(P2P File Sharing)佔最大宗，其架構的優點在於使用者需貢獻自己的網路上傳能力來換取更佳的下載速度，縮短下載的時間。在此架構下，整體環境的總上傳能力將會比主從式檔案傳輸(Client-Server File Sharing)高上許多。

因為行動代理人技術具有跨平台、可移動之特性，可以降低伺服器端的負載，所以本論文提出一個結合行動代理人的同儕式檔案推播平台，探討進行高資料量推播之可行性及優缺點為何？系統實作是以MACE(Mobile Agent Carrier Environment)作為行動代理人的運載環境，實驗的量測結果顯示將主從式行動代理人推播架構改進為同儕式行動代理人推播架構，將可以面對未來高資料量的資訊推播需求。

關鍵詞：推播技術、P2P 檔案分享、行動代理人

Abstract

As the broadband network technique is coming, classic push system focus on provide small data, like e-newspaper, RSS(Rich Site Summary), etc. But the push system that provides big data is rare.

Currently, the maximum network utilization in internet is peer-to-peer file sharing, its advantage is user need to contribute the upload bandwidth of self to get higher download bandwidth. In this file

transfer type, the file spread is better than Client-Server.

Because the characteristic of the Mobile Agent technique are cross platform, mobility, etc, it can reduce the sever load. So this paper proposed a P2P-like File Push System Using Mobile Agents, to see the feasibility on big data push system. Experiment result shows this system is better than Client-Server Mobile agent push system, and it can satisfy the push of big data.

Keyword: Push Technology、P2P File Sharing、Mobile Agent

一、研究動機

目前國內外皆已積極推動推播技術(Push Technology)的實際應用，例如，新竹風城、台北市公車已裝置無線廣告推播服務，日本 Sears 與 NTT Communication 合作建構無線廣告推播服務系統，美國與新加坡則主要裝置於大賣場內進行產品促銷，上海運用在計程車、樓宇間進行廣告的推播，預計未來將普遍應用於大賣場、地鐵、車站、車廂、超商、咖啡吧之內。

另一嶄露頭角的推播技術為 RSS，RSS 是一種用於新聞頻道、網誌和其他 Web 內容的資訊交換規範，源起於網景通訊公司(Netscape)的推播技術。RSS 讓網站能夠製造內容摘要，並提供 RSS 客戶端供訂閱的使用者閱讀摘要，免除上網瀏覽網頁所耗費的時間。

推播改變了以往資訊取得的主(Website)從(User)關係下的觀念，因推播系統大多是透過伺服器推播資訊給使用者，所以資訊主體的儲存、搜集、產生及使用興趣的判斷也都是在伺服器端，而使用

者的角色便只是純粹的接收端，跟以往 Http request/get 最大的不同在於推播只需少數的變更興趣的行為便可獲得接近期待的資訊，更可以省下時間與精力。

推播技術，基本上有下列三種優勢：

1. 提醒式的 Push: 有了 Push 技術，網站便能主動出擊，避免在使用者漫無邊際的搜尋瀏覽過程中被遺忘了。舉例來說，台灣的博客來網路書店 (<http://www.books.com.tw/e-page/>)，你也可以在博客來電子報專區填寫你想訂閱的種類，以後只要有符合的新資訊，便會主動的通知你。

2. 資訊式的 Push: Push 技術也可以提升服務品質，主動將資訊送給使用者。舉例來說，網站發行網站報紙刊物，每天免費將資訊主動寄給你，免你上網查詢之塞車之苦。如東森新聞網站 (<http://www.ettoday.com/case/express/>) 就提供了"新聞快遞"的服務。

3. 量身定製的 Push: Push 技術也可以達到量身定製 (Customization) 的目的。舉例來說，中華電信 (<http://demo.bossing.biz/CHT/Web/BusinessCat.php?CatID=65>) 也有提供客戶端新聞播送服務 (NewsCast Service): 利用推播 (PUSH) 技術，依照客戶端的需求，將新聞節目以即時、預訂的方式或是隨選 (On-Demand) 的方式，主動推播送到客戶端。

行動代理人 (Mobile Agents) 是一可在異質網路系統 (Heterogeneous network system) 內從一主機 (Host) 遷移 (Migrate) 至另一主機並和其他代理人及分散資源系統 (Distributed resources) 溝通及交互作用 (Interact) 之軟體程式。它們特別適合用來發展分散式及網路上的應用系統。

本論文之動機為針對目前的檔案推播系統，在多用戶且大資料量的推播需求增加的情況下，探討舊有的推播機制會有何缺點，並思考若是將行動代理人及 P2P File Sharing 的概念加入後，能否提供更佳的解決方案。在經過實驗後，證明使用同儕式行動代理人推播技術有三項的優點：

1. 面對不同資料量的推播內容，可以提供比行動代理人推播系統更佳的性能。

2. 透過行動代理人的逐站推播，可以有效減少資訊內容提供者的網路輸出流量。

3. 當參與任務的使用者越多，則完檔速度會越快。

二、 相關研究

2.1 行動代理人

行動代理人是可從某一主機遷移 (Migrate) 到另一主機並和其他代理人溝通與存取異質網路中資源的軟體模組。而自治型行動代理人則可完全自主地選擇要在何時及要往何地作遷移。而當其決定要作遷移的時候，行動代理人會先暫停其執行，並且遷移至新寄主處，而且並在其新寄主處恢復其執行的工作，因此特別適合用來發展分散式的應用系統。

一般來說，代理人 (Agent) 可以看成是一個獨立的模組 (module) 或程式 (program)。而每個代理人可以單獨執行使用者所賦予的任務，或與其他代理人做互動 (Interact) 及會談 (Meet)，以達到分工合作 (collaboration) 之目的。

以下為利用行動代理人建置或是支援同儕式網路 (Peer-to-Peer Network) 的相關研究，分別描述如下，[9] 證明了使用行動代理人在同儕式網路上可有效降低流量，[5] 使用數種不同目的之代理人來對同儕式網路進行最佳化，[13] 以 JXTA 實做代理人系統，利用行動代理人來解決同儕式網路中的安全性議題，[2] 以行動代理人建置一檔案分享的環境，提出搜尋的機制，也就是透過派送搜尋代理人，逐站詢問是否有用戶端願意提供使用者想要的資源資源，缺點是網路節點眾多，代理人不可能走完，因此這種搜尋方式不保證找到想要的檔案。

2.2 推播技術

消費者已經鮮有耐性等待漫長的查詢

而且最後還看到一堆沒有用的資料(或不是你想要的資料),因此推播技術(Push Technology)應運而生。它是改變以往的主(Website)客(User)關係,由被動接收到主動要求,網路上的 User 透過推播技術可"自主地只選擇網站上某個單元",而不需要瀏覽層層網頁,這就是方便的推播技術,而具有推播功能的網站就被稱為 Active Channel。

目前在網際網路上最普遍,而且也是大家最熟悉的推播技術就是 E-Mail,另外是一些電子報、廣告信等等,現在有許多號稱有推播能力的網站(如 Pointcast、Netcaster...等),其實這些網站大部分並非真正的將資料送到客戶端的主機上,而是網站提供給訂閱者應用程式,事實上就是應用客戶-伺服器(Client-Server)的模式,利用一些自動更新下載的軟體來取代使用者自己去更新最新的資訊,而這並非是所謂的推播技術,而是將連結到伺服器端及下載資料回來本地端主機(Local Host)這些動作隱藏起來,讓使用者以為是推播的技術,這些應用程式(applications)有時被稱為 smart pull 應用程式。而真正的推播技術應是由 Server 端將使用者所需的資料送到使用者的主機上或是使用者指定的地方去,這樣才能稱為推播技術,例如:E-Mail、多點傳送(Multicast)及廣播(Broadcast)這些技術等等。圖 1 為 pull/smart pull 與 push 之比較。

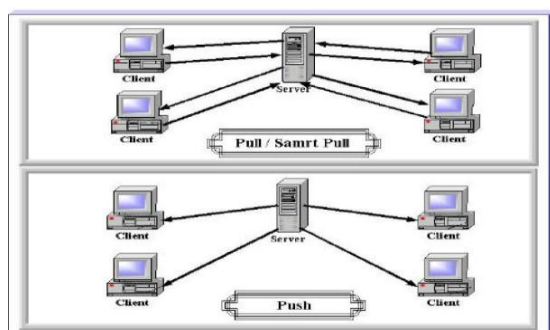


圖 1 pull/smart pull 與 push 的比較

2.3 MIME(Multipurpose Internet Mail Extensions)

電子郵件是分散式系統中最常被使用的應用系統。打從開始起,簡單郵件傳輸協定 SMTP(Simple Mail Transfer Protocol)

便已是 TCP/IP 協定集合中的一員。但是 SMTP 只能傳送簡單的文字訊息。在近幾年,人們希望能利用電子郵件來傳送聲音、影像及視訊等各種資料。為了滿足這個需求,一個建立在 SMTP 上之新的電子郵件標準及被推出。此新的標準即為多用途網際網路郵件延伸(Multi-Purpose Internet Mail Extension—MIME)。如圖 2 所示。

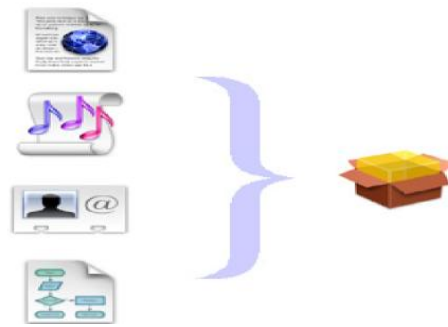


圖 2 封裝多個檔案到 Mail 中

MIME 使用物件包裝的方式,將各種不同格式的檔案打包好再予傳送,所以應用的範圍相當廣泛。MIME 允許數種不同的編碼格式,寄信者可以根據要寄的資料格式選擇編碼方式。圖 3 為 MIME 之傳輸碼方式,由於本系統只利用到 Base64 的編碼方式,所以在下一節中我們只介紹 Base64 編碼技術,另外幾項便不詳細介紹。

編碼方式	描述
7 bit	資料均被表示成由 ASCII 字元所組成之文字行。
8 bit	可含非 ASCII 字元之文字行。
Binary	可含非 ASCII 字元及長一點的資料行(不受 SMTP 傳輸之限制)。
quoted-printable	將資料編碼的方式為若被編碼資料的大部份為 ASCII 字元,則編碼後的格式仍維持人類所能理解之格式。
Base64	將資料由 6 位元之輸入區塊對應為 8 位元之輸入區塊,所有字元均為 ASCII 之可列印字元。
X-token	非標準之編碼方式。

圖 3 MIME 之傳輸碼方式

2.4 Base 64 傳送編碼方式

Base64 為一個二進位資料編碼的常用方法,以使資料不會受到郵件傳輸軟體之傷害。在目前支援 MIME 的郵件系統,必然會支援此一編碼、解碼方式。Base64 相

當適合二進位檔案的傳送，任何檔案格式均可透過 Base64 編碼夾帶於電子郵件中。圖 4 說明如何將 6 個位元對應到字元的方式。此字元集含有文數字及”+”與”/”。而”=”字元則被當作填補數字。

6位元 之值	字元編 碼方式	6位元 之值	字元編 碼方式	6位元 之值	字元編 碼方式	6位元 之值	字元編 碼方式
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

圖 4 Base64 之編碼方式

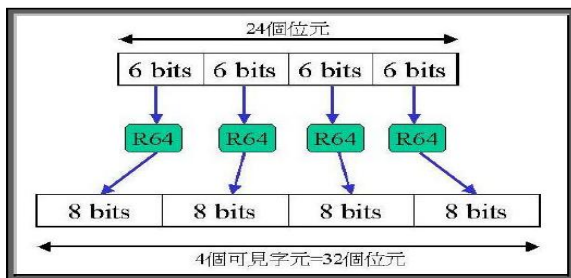


圖 5 將二進位資料編碼成 Base64 格式之方式

圖 5 說明了此簡單的對應方式。二進位之輸入資料會以一次一個位元組的方式被處理，在此區塊中的每組 6 個位元會被對應成一個字元，在圖中，這些字元會被表示成含有 8 個位元的數值。因此，每個 24 位元的輸入會被擴展成 32 位元的輸出。

此對應方式的一個重要特色即為代表此 65 個字元之 6 個位元的值在所有常用之字元集中均是相同的。例如：在 ASCII 系統中，”E”的值為 01000101，而在 EBCDIC 系統，其值為 11000101。在這兩種情形中，其右手邊的 6 個位元是相同的。因此若由 Base64 格式轉回二進位資料，只需取出每個字元之右邊 6 個位元即可。經由此編碼方式，使得 MIME 可適用於任何型式的檔案。

2.5 P2P File Sharing

P2P 網路近來的興起來自於 Napster[10]、Gnutella[6]、KaZaA[7] 及 BitTorrent[3, 4]等 P2P 檔案分享軟體出現。而近年來的量測研究也顯示，P2P 網路應用產生的網路流量已達到整體網路環境負載的四分之一，甚至超過了 WWW 的交通量[8, 12]。因此，P2P 網路在現今網路應用中的重要性是不言而喻的。

目前 BT (BitTorrent) 為 P2P 網路中目前最熱門且高效率的檔案傳輸技術之一，並被稱為第三代的 P2P 技術。BT 在 2001 年由美國軟體工程師 Bram Cohen 發明，它跟傳統的 P2P 有兩大分別：沒有中央搜尋系統、多點對多點傳輸。BT 亦沒有採用分散伺服器來追蹤檔案的搜尋，因此在網路上不會像 Gnutella 造成大量的搜尋流量。由於 BT 並不提供搜尋的機制，因此 BT 透過一個副檔名為”torrent”的檔案，將上傳者 (Seed) 和下載者 (Peer) 連結起來。BT 的另一特點是多點對多點傳輸。傳統 P2P 雖然將伺服器的的工作分流，但仍然未脫離單點對單點的下載方式，如果 Peer 一多就需要排隊下載，仍然會減低下載效率。但 BT 則善用了用戶在下載時沒用到的上傳頻寬，做到在下載的同時亦在進行上傳。換句話說，在 BT 的環境下，同一時間的任務參與者越多，總體上傳頻寬則越高。在此篇論文將著重於探討 BitTorrent-like System 下的架構在推行動代理人檔案推播系統上的可行性。

BT 的運作方式如圖 6 所示，在 BT 中，用戶之間透過一個 Tracker 伺服器所回傳之部分使用者清單來找到彼此，然後進一步互相溝通並且傳輸資料。Tracker 的功用類似一個索引伺服器，能夠追蹤目前到底有多少使用者同時在下載同一個檔案，當客戶端對 Tracker 伺服器提出下載要求時，就會獲得回傳的部分其他用戶名單，進而與其它用戶直接溝通並進行下載。需注意的是 Tracker 並不負責資料的傳送，而僅告知客戶端該跟哪些人提出下載要求。當使用者在進行下載的時候，同時也正在上傳資料給其他用戶。在 BT 中，正在下載而還

沒完成的用戶稱之為下載者 (Peer)，而已經完成的用戶則稱之上傳者或種子 (Seed)。

接下來簡介 BT 的片段散播流程，如圖 6 所示：

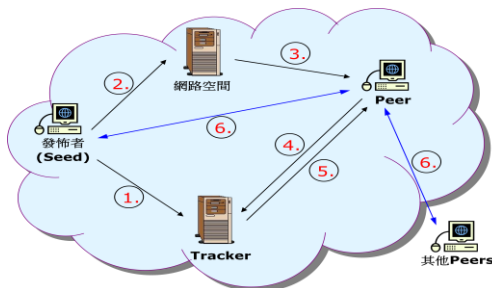


圖 6 BitTorrent 散佈片段之流程圖

1. 發佈者將要發佈的檔案做成 *.torrent 檔並且自己當 Seed，然後將 *.torrent 檔發佈到 Tracker 上。

2. 發佈者接著將 *.torrent 檔放在網站或是網路空間供有興趣之 Peer 下載。

3. 想要下載這個檔案的 Peer 須先下載有興趣的 *.torrent 檔。

4. Peer 根據 *.torrent 檔內包含的資訊得知 Tracker URL，接著向 Tracker 索取同樣下載這個檔案的 Peer 清單。

5. Tracker 回傳部分使用者清單。

6. 根據 Peer 清單向 Seed 與其他 Peer 溝通並且下載檔案。

我們以 BT 作為改善行動代理人推播系統的研究對象，參考其資源描述文件及 Tracker 機制。在[4]這篇文章中，載有 BT 協定的運作方式：資料提供者及下載者都須安裝 BT 軟體；原始檔案提供者要將其檔案公佈出來時，可利用 BT 程式建立 torrent 檔，而 torrent 是對要分享的檔案做描述的文件，內含 Tracker URL、檔案名稱、檔案大小、分段大小等等，接下來原始檔案提供者需將此 torrent 公佈在網路空間中供有興趣的使用者下載，一般常見的方式為將 torrent 發佈在網路論壇中，並對 torrent 做額外的描述，如以文字描述或是加上圖片，供下載者判斷是否要下載。那下載者在 BT 進行下載任務前，須先下載 torrent 檔，再利用 BT 開啟此 torrent 檔，經由 torrent 中紀錄的 Tracker URL 資訊，向 Tracker 要求

下載者清單來找到其他下載者，進而跟清單中的其他下載者相互溝通完成檔案的下載。

Tracker 與下載者之間是使用 HTTP 協定溝通，下載者會告知 Tracker 想要下載哪個檔案，接著 Tracker 會回傳下載者一份清單，內容為下載同一份檔案的其他 Peers 資訊，由於 Tracker 只負責上述的資訊交換，所以 Tracker 的網路及主機能力需求不高。另外 Tracker 傳回的 Peer 清單是使用隨機選取的方式，採用隨機選取的方法就較不會讓某幾台 Peer 或 Seed 的負載過高，進而達到負載平衡的目的。儘管 Tracker 負擔的事情並不多，但卻是 Peer 與 Peer 之間發現互相的唯一路徑，所以一旦 Tracker 當機就會造成 Peer 之間無法互相發現。

三、系統架構

在這一節將詳細介紹系統設計的動機及概念。首先會闡述系統設計之目的與概念，進而訂定系統的功能需求。並依功能需求來規劃及設計系統的架構與組織，及詳細敘述系統中組成元件的功能與系統之運作模式。

3.1 系統設計之動機與目的

所謂的推播系統，初期是以主從式架構為主，簡單來說就是資訊主體都是由伺服器端發送到訂閱的客戶端，但這樣的架構其擴展性不佳，容易在用戶端數量增加的狀況下造成伺服器服務品質下降，後來有行動代理人支援推播技術[1]的提出，利用行動代理人可在不同節點之間遷移之能力來進行推播的任務。對伺服器來說，只需派送少數的行動代理人便可以服務多個用戶端，達到減輕伺服器端的負擔及流量。上述的推播系統，其推播的內容資訊實體都是存放在伺服器端，由伺服器端決定使用者所接收的資訊，因此用戶端無法享受到其他人所持有的資源。另外，當推播的內容不再侷限於小資料量的資訊時，如文件、網頁及圖片等，而是大資料量的內容，如網路短片、軟體映像檔及電影預告等的話，那代表單一任務行動代理人在節點的

間遷移必須花費比以往更多的流量及時間，因此上述的推播系統架構皆不適合用來推播大資料量的內容。

寬頻網路的時代已來臨，全球網路流量中又以 P2P 軟體的使用佔最大宗，P2P 的架構讓同一社群的使用者彼此互相分享所持有的資源，例如 CPU 計算能力、儲存空間、資料內容等，讓使用者可以享受到原本無法負擔的資源。參考到目前 P2P File Sharing 中最熱門的 BitTorrent，本研究提出了同儕式行動代理人推播機制的想法，除了保有先前行動代理人推播系統的優點（降低伺服器端流量）外，此同儕式推播機制更適合推播大資料量的資訊內容，且提供用戶端之間互相分享私人持有的資源。

透過本同儕式行動代理人推播平台的建置，預期可以達到以下優點：

1. 當任務資料量越大時，行動代理人毋須攜帶完整檔案進行推播，而改以推播切細過的檔案片段，因此每個片段推播代理人繞送的時間並不會因資料量大小而改變。另外用戶端可在同一時間接收多個用戶端所推播的相異片段，因此可以比行動代理人推播系統提供更好的效率。

2. 同一下載任務的參與用戶端數量增加時，則總上傳頻寬能力也會隨之增加，代表的是任務的完成時間將會因用戶端的參與數量多寡而遞減。

3.2 代理人執行環境

以下簡述本同儕式行動代理人檔案推播平台所參考之基礎建設(Infrastructure) MACE(Mobile Agent Carrier Environment)[14]系統的架構與各組織的功用。如圖 7 所示，整個 MACE 系統依其功能可分為下列主要的部分：

代理人發送模組 (Agent Lunch Module) 就如同字面意思是創造及發送代理人到網路上的模組。被送出去的代理人漫遊在網路上然後將一些服務項目攜帶到**代理人伺服器模組 (Agent Server Module)**。服務協定(Service Protocol)中，代理人發送模組是**代理人產生器 (Agent Creator)**的實作。當需要使用某些服務時，

使用者可以查閱**目錄服務模組 (Directory Service Module)** 來找出較適合或較喜愛的服務且下載相關的 agentlets。而**行動存取模組 (Mobile Accessing Module)** 就是行動平台將代理人發送到網路或是從網路上接收代理人的一個橋樑。

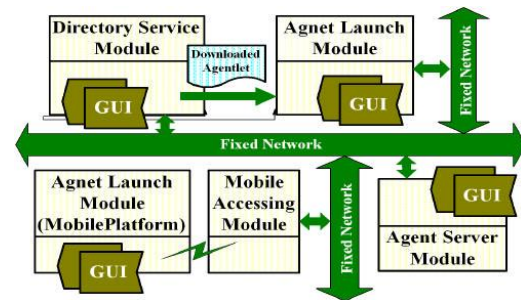


圖 7 MACE 系統概觀

3.3 系統概觀及架構

本系統主要成員有 Tracker 端及用戶端，其系統架構圖如圖 8 所示，以下將分別介紹及描述主要的模組功能。

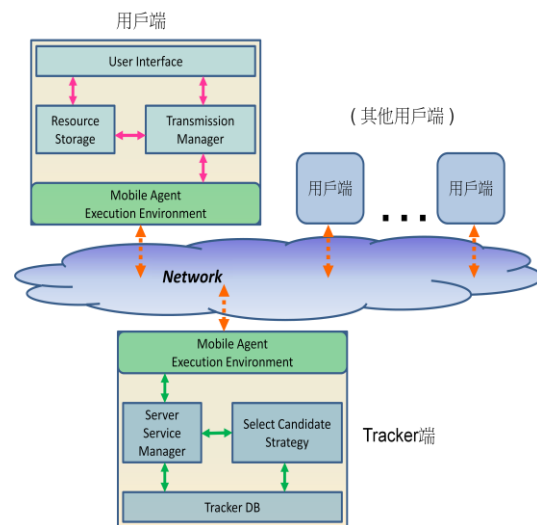


圖 8 同儕式檔案推播系統架構圖

- **用戶端介面 (User Interface)**：本同儕式推播系統的用戶端使用者介面，主要提供的功能有二，分別為 (1) 開啟新的資源描述文件以新增下載任務，(2) 選取特定資源，在此平台進行新的分享。

- **資源儲存體 (Resource Storage)**：用來儲存其他用戶端所推播的檔案片段或是從網頁下載的資源描述文件。

- **傳輸管理模組 (Transmission**

Manager)：掌管用戶端的網路輸入及網路輸出。當用戶端開啟資源描述文件以新增下載任務時，此傳輸管理模組將對此任務尚未完成之片段產生片段下載請求訊息，透過 MAEE (Mobile Agent Execution Environment) 派送至 Tracker 端。若用戶端收到其他用戶端所派送的推播代理人時，此傳輸管理模組剖析出代理人所攜帶的片段後，進行 MIME/Base64 解碼及 MD5 雜湊值驗證，驗證成功後才儲存至資源儲存體中並更改此任務片段狀態為已接收，最後還須判斷自己是否為此代理人繞送的最後一站，若為最後一站，則傳輸管理模組將銷毀此片段推播代理人。若非最後一站的話，則在儲存動作完畢後，將此片段推播代理人透過 MAEE 往下一站繼續繞送，直到最後一站才結束。

➤ **伺服器端服務管理模組 (Server Service Manager)**：負責接收用戶端所派送之片段下載請求代理人及片段完成回報代理人。當收到片段下載請求代理人時，叫用片段候選人選取策略模組挑選出最合適的候選人，最後派送工作通知訊息代理人給候選人，以進行指派的推播任務。另外當收到片段完成回報代理人時，即代表在此推播平台上，擁有某片段複本的擁有人又多了一個，將此訊息儲存至 Tracker 端紀錄資料庫。

➤ **片段候選人選取策略模組 (Select Candidate Strategy)**：將伺服器端服務管理模組所轉送之片段下載請求，轉為 SQL 命令，對 Tracker 端紀錄資料庫進行檢索，接著對回傳的資料集進行選取策略，選出適合此片段下載請求的最佳候選人。

➤ **Tracker 端記錄資料庫 (Tracker Database)**：紀錄 Tracker 端提供服務所需的資訊，包含用戶端及運行中的任務相關資訊，如：任務名稱、用戶端 IP 及用戶端目前工作負載量等等。

3.3.1 片段候選人選取策略

Tracker 端當收到客戶端所派送之片段下載請求代理人後，會交付片段候選人選取策略進行選取。此選取策略的主要目的

在平均片段擁有者的工作負載量。其工作負載量在資料庫資料表中以 JobCount 紀錄之。

當 Tracker 端選出候選人 A 並派送工作通知代理人後，將候選人 A 的 JobCount 加一。日後當 Tracker 端收到片段回報代理人後，攜帶結果為某用戶端 B 完成接收來自用戶端 A 的片段推播，則將候選人的 JobCount 減一。透過此選取方式，針對每一片段下載請求，片段候選人選取策略所挑選出的都是目前最少工作負載量的擁有者。

3.3.2 資源描述文件

資源描述文件 RDD (Resource Describe Document) 是用戶端分享資源的媒介，一份 RDD 代表的是此平台上目前有此一任務正在進行，RDD 內含的訊息有此任務的描述、檔名、大小、片段大小、片段總數及 MD5 雜湊值集合等，以下將會描述 RDD 的內容及格式，格式如圖 9 所示：

```
<?xml version="1.0" tandalone="yes"?>
<Attribute>
  <Type>RDD</Type>
</Attribute>
<File>
  <FileName>REM.mp3</FileName>
  <FileSize>6516536</FileSize>
  <PieceSize>102400</PieceSize>
  <PieceCounter>7</PieceCounter>
  <MD5HashValue>FF67B0E2AD1CDC7AE07A8CAB99E380D0, 0
  0F528DBAFFB3FDC38E1F9E1739DB8CD, DFABA20D81B50F89
  FE3ACA131392630F, 54B8F86876C162FCDCE77A19EB03FB4
  0, 30F40C336E5A3B7894D521D62F4456C3, AF17120935921
  77A2EC795CE7026D7CE, A3666879E9C946E85A0A65343009
  716A</MD5HashValue>
</File>
```

圖 9 資源描述文件 (RDD) 格式

3.4 系統使用之 XML 格式代理人

在本推播平台中，為了讓遊走在用戶端及 Tracker 端的行動代理人能被兩方解譯，因此制定以下數種相異的代理人格式，依不同的需求而有所差異。以下將逐一的介紹。

3.4.1 用戶端分享新任務代理人

本平台上的用戶端彼此可以進行資源的分享，當用戶端選定欲進行分享的檔案時，會由用戶端傳輸管理模組派送一代理

人通知 Tracker 端，在 Tracker 端上新增任務的相關訊息，內容有訊息的形態（1 代表的是客戶端分享新任務通知代理人；2 代表的是用戶端片段下載請求代理人；3 代表的是 Tracker 端工作通知代理人；4 代表的是客戶端的片段完成回報代理人。）、欲分享檔案的名稱、經過分段後的片段索引總數及分享資源用戶端的網路位址。此代理人格式說明如圖 10 所示下：

```
<?xml version="1.0" standalone="yes"?>
<Attribute>
  <Type>1</Type>
</Attribute>
<File>
  <FileName>test.avi</FileName>
  <PieceCounter>700</PieceCounter>
</File>
<Network>
  <IP>140.116.39.123</IP>
</Network>
```

圖 10 分享新任務代理人

3.4.2 用戶端片段下載請求代理人

用戶端透過開啟 RDD 來新增下載任務後，交付傳輸管理模組來處理整個下載任務的請求及接收。傳輸管理模組會針對目前此任務所缺的片段定期的對 Tracker 端產生片段下載請求代理人，內容有訊息型態、用戶端欲下載之任務檔案名稱、欲下載的片段及提出此片段下載請求的用戶端網路位址。此代理人格式說明如圖 11 所示：

```
<?xml version="1.0" standalone="yes"?>
<Attribute>
  <Type>2</Type>
</Attribute>
<File>
  <FileName>test.avi</FileName>
  <PieceIndex>3</PieceIndex>
</File>
<NetWork>
  <IP>140.116.39.116</IP>
</NetWork>
```

圖 11 片段下載請求代理人

3.4.3 Tracker 端工作通知代理人

Tracker 端在收到用戶端的片段下載請求代理人後，透過片段候選人選取策略挑選出適合的片段推播候選人，接著派送工作通知代理人給選出的候選人，請候選人對目的地進行指定的任務片段推播，工作通知內容為訊息型態、此推播工作的任務檔案名稱、檔案片段索引、此工作通知

息的目的地候選人網路位址及曾經跟 Tracker 端要求過同一片段的用戶端。如圖 12 所示。此代理人格式說明如下：

```
<?xml version="1.0" standalone="yes"?>
<Attribute>
  <Type>3</Type>
</Attribute>
<File>
  <FileName>test.avi</FileName>
  <PieceIndex>7</PieceIndex>
</File>
<NetWork>
  <IP>140.116.39.117</IP>
  <Destination>140.116.39.108, 140.116.39.112, 140.116.39.125</Destination>
</NetWork>
```

圖 12 工作通知代理人

3.4.4 用戶端片段推播代理人

當用戶端從 MAEE 取出 Tracker 端所派送之工作通知代理人後，解譯並取出所夾帶的資訊（任務名稱、片段索引及目的地 IP），將任務片段進行 MIME/Base64 編碼（如圖 13 所示）後填入片段推播代理人，對目的地 IP 進行逐站推播。

在解說此型態代理人格式前先介紹 MIME/Base64 編碼與解碼，在相關研究章節中，已介紹到我們採用 MIME/Base64 編碼方式來對檔案片段進行編碼與解碼，由於本系統之行動代理人是由一堆自行定義標籤所組成的 XML 文件，假若我們想把檔案片段交付代理人攜帶進而推播繞送，那必須先行轉換成可讓行動代理人攜帶的文字碼。另外 Base64 這項編碼技術可應用於任何形式的檔案，包含聲音、影像等二進位資料。在圖 13 中，將針對一個 test.wmv 的影像檔進行 MIME/Base64 的編碼，將編碼後的結果將會填入代理人的 <PieceContent> 標籤欄中。

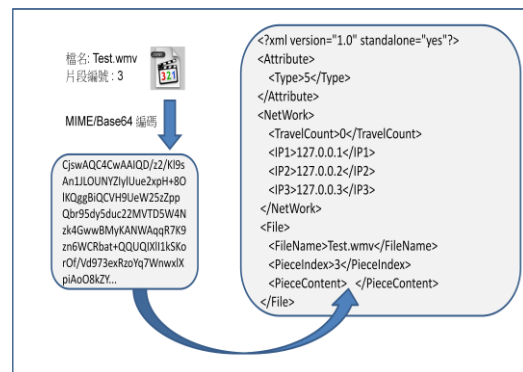


圖 13 片段推播代理人

3.4.5 用戶端片段完成回報代理人

當用戶端從 MAEE 佇列中取出其他用戶端所派送之片段推播代理人後，進行解譯及 MD5 Hash Value 驗證後，必須回報 Tracker 端，代理人內容說明送出訊息的用戶端目前是某一任務片段的複本擁有人。此代理人格式說明如圖 14 所示：

```
<?xml version="1.0" standalone="yes"?>
<Attribute>
  <Type>4</Type>
</Attribute>
<File>
  <FileName>test.avi</FileName>
  <PieceIndex>5</PieceIndex>
</File>
<NetWork>
  <IP>140.116.39.117</IP>
</NetWork>
```

圖 14 片段完成回報代理人

3.5 系統運作模式

本文之同儕式推播系統運作可細分為五個部份，透過以下的五個流程，用戶端可以在此平台上透過行動代理人來完成散佈資源給其他用戶端及從其他用戶端獲取資源。

3.5.1 用戶端分享新資源之流程

當用戶端想在此推播平台上分享自己持有的資源時，需對此資源產生一個資源描述文件，內容是描述此資源的名稱、大小、片段大小及片段總數，接著將此資源文件散佈到網路上以供其他用戶端下載使用，其流程如圖 15 所示：

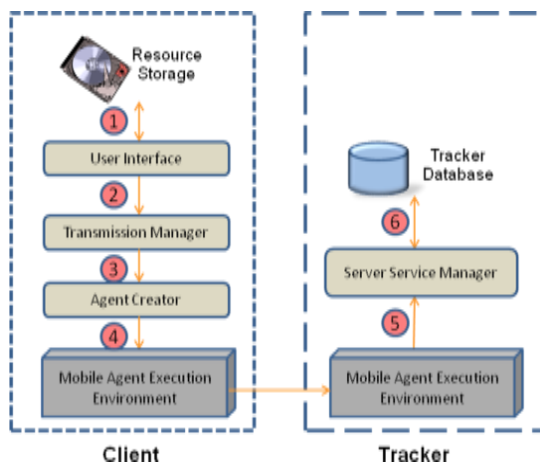


圖 15 用戶端分享新資源之流程

1. 用戶端透過使用者介面選取欲分享的資源檔案。
2. 待選定檔案後，將檔案的名稱、片段總數及用戶端自身 IP 告知**傳輸管理模組**。
3. **傳輸管理模組**進行產生資源描述文件，此文件將會放在一公開已知的網路上供其他用戶端下載。另外也會產生新任務通知訊息，交付給 Agent Creator。
4. 將**傳輸管理模組**交付的新任務通知訊息包裝成代理人格式，透過 MAEE 傳送到 Tracker 端。
5. 取出代理人送來之片段總數、任務檔案名稱及擁有者 IP。
6. 透過**伺服器端服務管理模組**，將所得之資訊轉為 SQL 命令，對 **Tracker 端紀錄資料庫** 進行新增的動作。

3.5.2 用戶端進行下載任務之流程

用戶端可以透過使用者介面開啟已下載的資源描述文件來新增一個下載任務，此任務將由傳輸管理模組控管，透過監視此任務尚未完成的片段來定期產生片段下載請求，而片段下載請求會交由 Agent Creator 包裝成代理人格式後透過 MAEE 傳送到 Tracker 端，以達到進行下載任務的目的。其流程如圖 16 所示：

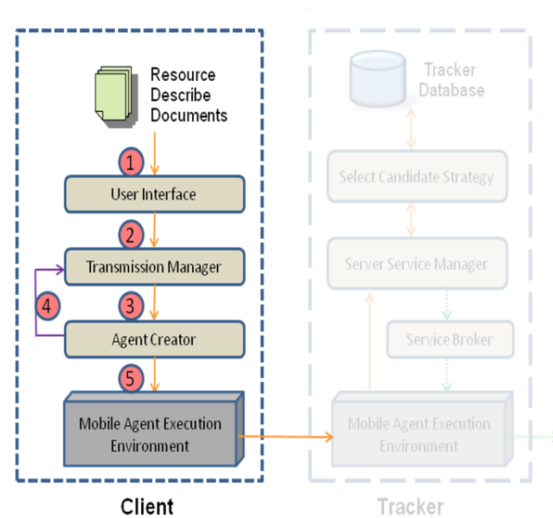


圖 16 用戶端進行下載任務之流程

1. 透過**使用者介面**開啟已下載之資源描述文件。
2. 產生新的下載任務，交付**傳輸管理模組**監控。
3. **傳輸管理模組**透過監視未完成下載的片段來定期的產生片段下載請求訊息，直到任務的所有片段都已完成接收及驗證。
4. 若任務仍有片段尚未完成接收，則繼續下一次的片段下載請求。
5. **傳輸管理模組**產生的片段下載請求透過 Agent Creator 包裝成代理人格式，透過 MACE 傳送至 Tracker 端。

3.5.3 Tracker 端處理用戶端下載請求支流程

Tracker 端在收到用戶端的片段下載請求代理人後，**伺服器端服務管理模組**取出代理人夾帶之參數，交付**片段候選人選取策略模組**進行處理，挑選出最適合此片段下載請求代理人的片段推播候選人，挑選出後，**伺服器端服務管理模組**會產生工作通知訊息，經由 Agent Creator 包裝成代理人格式透過 MAEE 送出，目的地為先前挑選出的片段推播候選人。其流程如圖 17 所示：

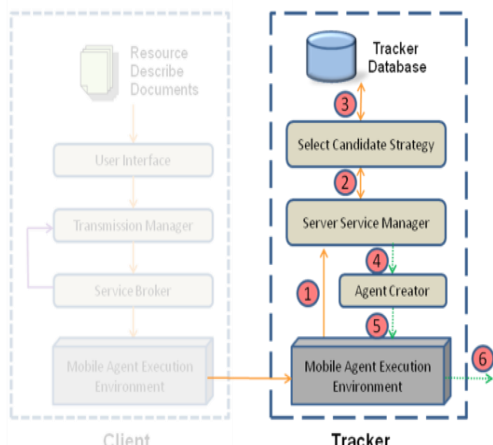


圖 17 Tracker 端處理用戶端下載請求支流程

1. **伺服器端服務管理模組**從 MAEE 佇列中取出片段下載請求代理人。
2. 取出片段下載請求代理人夾帶之參數後，交付**片段候選人選取策略模組**進行處理。

3. **片段候選人選取策略模組**將參數轉為 SQL 命令，對 Tracker 端紀錄資料庫進行檢索，挑選出最合適的片段候選人。
4. **伺服器端服務管理模組**產生工作通知訊息，內容主要為通知挑選出的片段推播候選人該對哪些用戶端們進行某個檔案片段的推播。
5. 將工作通知訊息透過 Agent Creator 包裝成代理人格式，接著透過 MAEE 送出。
6. 目的地為先前挑選出的最合適片段推播候選人。

3.5.4 用戶端進行推播任務之流程

同儕式推播系統中，跟以往推播系統不同的地方在於用戶端不再只是提出服務請求，也需提供服務給其他用戶端，而在同儕式推播系統裡，用戶端所提供的主要服務就是推播片段給其他用戶端。接下來將會介紹用戶端如何達成推播的服務，其流程如圖 18 所示，說明如下：

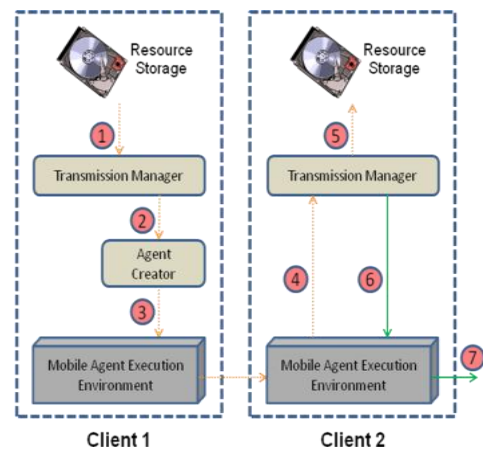


圖 18 用戶端進行推播任務之流程

1. 已知 Client 1 用戶端剛剛收到 Tracker 端所傳送之工作訊息通知代理人，**傳輸管理模組**取出代理人夾帶之參數，參數中包含要進行推播的檔案片段索引及目的地用戶端。**傳輸管理模組**在此步驟中把要推播的檔案片段載入並進行 MIME/Base64 編碼，編碼成可以讓代理人攜帶的文字格式。
2. 把檔案片段編碼後的結果及目的地用戶端交付 Agent Creator。

3. 將**傳輸管理模組**傳送的訊息包裝成代理人格式，此代理人為凸播片段代理人。

4. Client 2 用戶端的**傳輸管理模組**從 MAEE 佇列中取出片段推播代理人。

5. 取出片段推播代理人所攜帶之編碼片段，進行解碼、驗證後寫入**資源儲存體**中。

6. **傳輸管理模組**透過片段推播代理人的<TravelCount>欄位可以得知自己是否為最後一站，假若是的話則直接銷毀此片段推播代理人。若非最後一站，則將此片段推播代理人透過 MAEE 往下一個目的地推播，直到最後一站為止。

7. 片段推播代理人繼續往下一個目的地進行推播。

3.5.5 用戶端回報片段完成之流程

Tracker 端如何得知用戶端之間彼此推播的結果狀態呢？方法是用戶端每當接收片段及驗證結束後，須產生一**片段完成回報代理人**到 Tracker 端，進行資料庫更新的動作，透過此方式，Tracker 端的紀錄資料庫中擁有目前平台中所有片段擁有人的資訊，再透過**片段候選人選取策略模組**依工作負載量的高低挑選出相異的片段推播候選人。接著描述用戶端產生**片段回報代理人**之流程，其流程如圖 19 所示，說明如下：

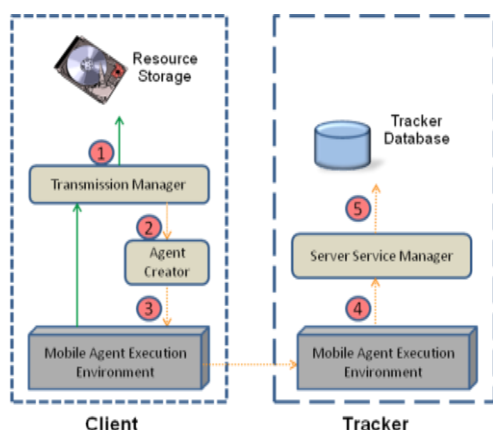


圖 19 用戶端回報片段完成之流程

1. 用戶端**傳輸管理模組**從 MAEE 佇列中取得片段推播代理人，取出代理人所夾帶的檔案片段，在存入資源儲存體之前須進行 MD5 雜湊值驗證，將驗證值跟資源

描述文件中的片段雜湊值比對，若相同則代表此片段式接收完全且無遭湊竄改，**傳輸管理模組**接著會產生一**片段完成回報訊息**；若相異的話則有可能是傳輸不完全或是遭受汙染，**傳輸管理模組**則會馬上銷毀此片段推播代理人並重新此片段的片段下載請求訊息。

2. 當片段 MD5 雜湊值驗證成功之後，**傳輸管理模組**會產生**片段完成回報訊息**。

3. 交由 Agent Creator 把**片段完成回報訊息**包裝成代理人格式，接著由 MAEE 傳送至 Tracker 端。

4. Tracker 端的**伺服器端服務管理模組**由 MAEE 取得**片段完成回報代理人**。

伺服器端服務管理模組取出代理人所夾帶之回報資訊，轉為 SQL 命令，對後端資料庫進行新增的動作。也代表此一檔案片段在本平台上又多了一位複本擁有者。

四、系統實作

本系統之客戶端及 Tracker 端的作業系統是 Windows XP SP2，集成開發環境是 Borland C++ Builder 6 SP4 with Indy 9 Socket Library，資料庫是 MySQL v5.0.24a。

本系統主要由用戶端及 Tracker 端所組成，用戶端的主要功能有三，一為分享資源給其他有興趣的用戶，二為接受其他用戶端所推播的任務片段，三為接受 Tracker 端所交付的推播工作通知訊息。Tracker 端的主要功能有二，一為處理用戶端的片段下載請求代理人，透過推播片段候選人選取策略來選出合適的候選人，接著產生推播工作通知代理人給此候選人，並更新此候選人的工作負載量，二為處理用戶端的片段完成回報代理人，對資料庫進行更新的動作。以下將會分別描述用戶端及 Tracker 端兩者之實作介面及功能。

4.1 用戶端之實作畫面介紹

在用戶端的實作介面中分為兩個頁次，

第一個頁次請參考圖 20，主要的功能方塊有網路設定、製作資源描述文件、新增下載任務、任務列表及本機端工作訊息。網路設定方塊提供用戶端指定自己的 IP 位址。製作資源描述文件方塊提供用戶端在選定欲分享的檔案後，產生 RDD 及派送分享新任務通知代理人至 Tracker 端。新增下載任務方塊提供用戶端開啟下載的 RDD，於點擊後在任務列表中新增一新的下載任務。任務列表方塊在用戶端介面時開啟時載入所有未完成任務的狀態，若用戶端開啟 RDD 新增下載任務的話，則會在任務列表新增一任務紀錄。本機端工作訊息方塊顯示的主要為本機端的相關訊息，譬如：網路設定是否運行、RDD 的產生進度及任務是否完成。

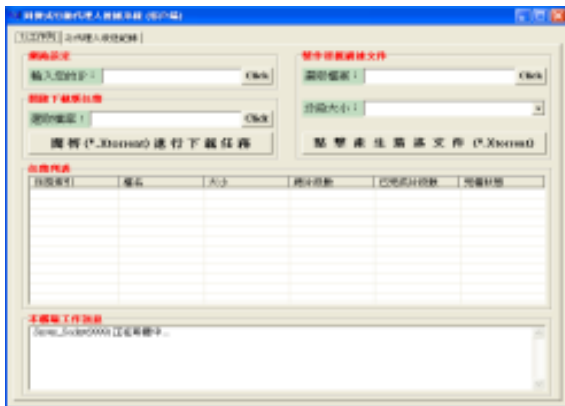


圖 20 用戶端實作介面一

用戶端的實作介面中第二個頁次請參考圖 21，主要是用戶端的代理人收送狀態，於上方的代理人接收紀錄中，主要在顯示此用戶端所接收到的代理人，包含 Tracker 端所送之工作通知代理人，或是其他用戶端所派送之片段推播代理人。下方的代理人派送紀錄，主要是顯示用戶端所派送出去的代理人，包括送至 Tracker 端的片段下載請求代理人，或是派送至其他用戶端的片段推播代理人。

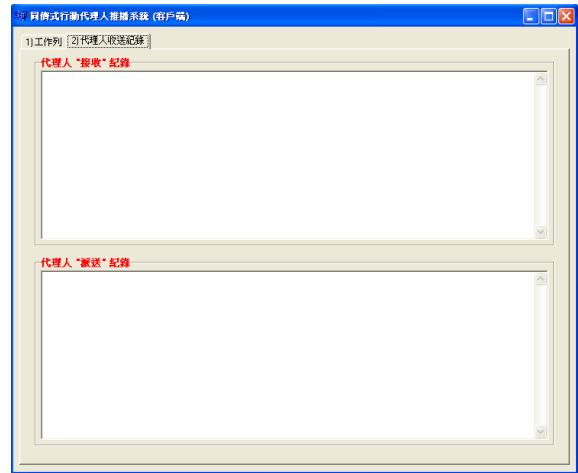


圖 21 用戶端實作介面二

4.2 Tracker 端之實作畫面介紹

Tracker 端在同儕式推播系統中扮演的是一個服務的角色，其實不需要介面，但為了方便了解 Tracker 端的服務內容，給定 Tracker 端一個訊息顯示介面，如圖 22 所示。Tracker 端的服務內容簡而言之就是三個，一為處理用戶端所派送之片段下載請求代理人，二為處理用戶端所派送之片段完成回報代理人，三為派送工作通知代理人給選定的片段推播候選人。實作方面則是使用佇列方式來存放待處理的代理人，在畫面左上方 (Job in Queue) 為 Tracker 端目前所接收到的代理人待辦任務；畫面右上方 (Job in Done) 為 Tracker 端每處理完一佇列中任務的任務就顯示通知。畫面左下方 (SQL Message) 為在 Tracker 端處理佇列中的待辦任務時，若有對後端資料庫進行檢索、新增或更新的動作時，將訊息顯示在此。右下方 (System Log) 顯示的訊息為 Tracker 端在派送工作通知代理人。

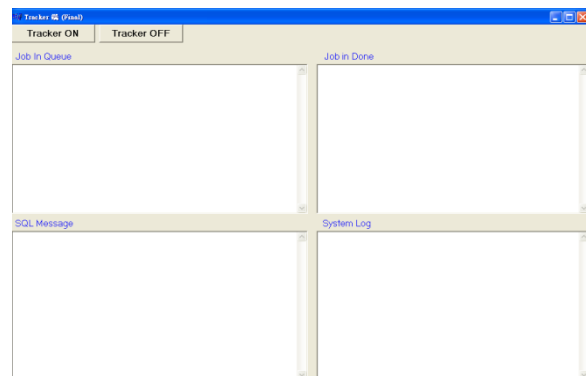


圖 22 Tracker 端訊息顯示介面

五、實驗設計與比較

本實驗將會以同儕式檔案推播平台為實驗組，以行動代理人推播系統為對照組，實驗的設計將會針對改變推播檔案片段的大小及不同的推播代理人繞送的深度。

5.1 資料量大小相異之任務完檔時間比較

實驗目標：進行資料量大小不同的推播任務，統一片段大小為 2MB；以行動代理人支援推播機制當對照組，而同儕式行動代理人檔案推播機制當實驗組，兩組之行動代理人繞送深度定為 3；最後比較所有用戶端對不同大小任務所耗費的平均完檔時間，驗證在資料量越大的狀況下，同儕式檔案推播機制會比行動代理人推播機制來的更有效率。

實驗設計：以六台相同規格之電腦當作實驗對象，其中五台為用戶端，剩餘一台為 Tracker 端。資料庫安裝在 Tracker 端，所有用戶端在同一時間新增下載任務，量測平均完檔時間。

結果分析：如圖 23 所示，實驗組織數據並無法大幅頂先對照組的原因在於參與的用戶端數量不夠多，當參與的用戶端越多，越有機會在同一時間接受到更多其他用戶端的推播。但已經可以得證實驗組的機制會比對照組來的有效率。

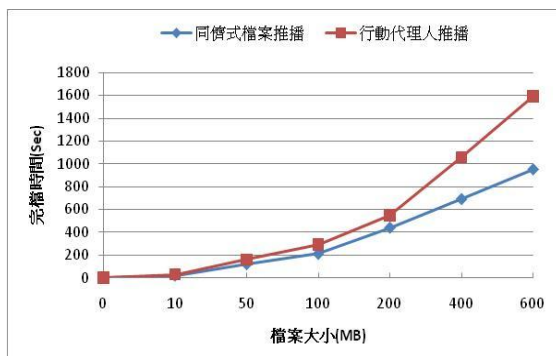


圖 23 資料量大小相異之任務完檔時間比較圖

5.2 同一任務不同繞送深度之完檔時間比較

實驗目標：片段推播代理人繞送深度為可變的參數，若繞送深度越淺，優點為

效率高，缺點為無法有效降低資訊提供者的流量。若繞送深度越深，優點為可以降低片段推播代理人的派送個數，缺點是越後面的目的地完檔時間越久，拉長完檔時間。此實驗的目的在相同任務下給定不同繞送深度，比較其效能差異。

實驗設計：以六台相同規格之電腦當作實驗對象，其中五台為用戶端，剩餘一台為 Tracker 端。資料庫安裝在 Tracker 端，所有用戶端新增相同的下載任務，任務大小為 50MB，在行動代理人繞送深度分別為 1、2、3 的條件下，量測其平均完檔時間。

結果分析：在深度越深的狀況下，可以由圖 24 得知完檔時間也會拉長，最不好的情況為本身為最後一個目的地，須多等待兩個繞送節點時間，好一點的為等待一個繞送節點時間，因此完檔時間會比深度一來的長。

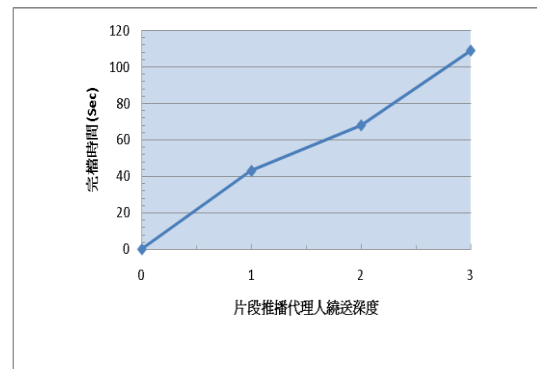


圖 24 同一任務不同繞送深度之完檔時間比較

5.3 同一任務下，與 BitTorrent 的 Traffic 比較

實驗目標：給定一相同的檔案去進行分享，分別以此平台推播的方式及使用 P2P 的方式去散佈，在此 P2P 使用 BitTorrent 的方式進行分享，量測完成散佈行為所需額外耗費的訊息流量。

實驗設計：分享一 100MB 的檔案，片段大小設為 2MB，共計有 50 個片段。紀錄所有用戶端個數(n)所花費的總吞吐量(X)，最後計算得出 $Traffic(T) = X - (100 \text{ MB} \times n)$ ，以圖表表示 T。

結果分析：BT 的訊息流量大多是發生在用戶端之間溝通彼此是否有無所需求的

片段時，另一個則是每隔一固定時間 BT 用戶端會對 Tracker 端提出下載用戶端清單，透過下載清單得知更多的來源，因此當用戶端參與數量越多時，用戶端之間的溝通流量會隨之增加。而本同儕式行動代理人推播系統之訊息流量大多發生在用戶端對 Tracker 端提出下載請求時，並不會對其他參與的用戶端進行溝通，因此其訊息流量接近線性成長。由圖 25 得知當 BitTorrent 在用戶端數目增加後，其訊息流量將會比本同儕式推播系統增加速度來的快。

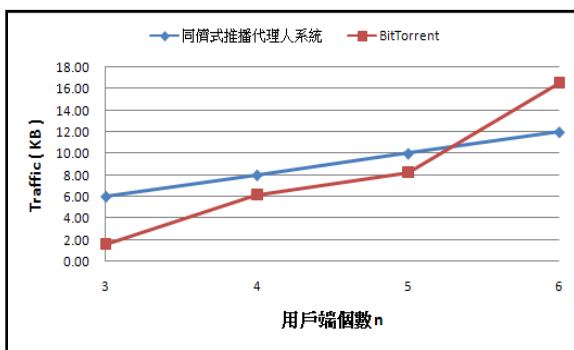


圖 25 同一任務不同參與用戶端之訊息流量比較

六、結論

本文所提出同儕式行動代理人檔案推播平台的概念，成功結合以下四項優點：

1. 利用行動代理人可以逐站繞送的特性來進行推播，可有效降低資源提供者的流量。

2. P2P File Sharing 的分段傳輸概念讓行動代理人攜帶的不是整個檔案而是切割過的片段，如此可以減低行動代理人在不同節點間繞送的流量，也可以透過派送多個攜帶相異片段的代理人來進行推播，達到在短時間內創造更多片段的複本。

3. 此平台中的檔案散佈是以片段複本的形態存在於所有的用戶端中，因此即便原始供檔者不再提供分享，用戶端依然可以透過持有其他片段複本的用戶端推播而完成整個檔案，因此比以往的推播系統擁有更佳的容錯率。

4. 用戶端之間可以彼此分享及獲取資源，不像先前的推播系統只能接受由伺

服器端所推播過來的資訊。

以 P2P 分段傳輸的概念融合行動代理人推播片段的想法，實現了同儕式行動代理人檔案推播平台，提供了一個傳輸方式給未來的高資料量需求。

七、未來展望

未來可以利用現有的發展成果為基礎，改善原有的缺點，並增加新的功能，使整個系統能更加完備。由於本系統是架構在行動代理人執行環境上，我們在未來將以改良系統的支援性及效率性來做考量，以期發展更好的系統效能，以下是未來發展可改進及建議的部分：

- 目前本推播平台並不支援行動裝置，因此未來若是想在行動裝置上享受同儕式推播的服務，則需針對行動裝置另外再開發用戶端介面。

- Tracker 端的實作方面，雖然可以依工作負載量來進行調整指派任務，但是並沒有參考到其他因素，例如用戶端的網路剩餘能力、計算能力及儲存空間等。

在本同儕式推播中，可讓不同的用戶端互相進行片段推播，但僅限於推播 Tracker 端所指派的片段及目的地，未來可以新增一行動代理人，專門負責進行搜集用戶端之間的片段擁有資訊。如用戶端 A 的任務為對用戶端 B 進行片段推播，則用戶端 A 同時派送兩種代理人，一為片段推播代理人，另一為片段擁有資訊搜集代理人，搜集完後須返回用戶端 A，那用戶端 A 接下來針對用戶端 B 所缺的片段再進行派送片段推播代理人，如此可以減輕 Tracker 端的資料庫運算，也可以更快速的完成下載任務。

八、參考文獻

- [1] 鍾昆學, "以行動代理人支援推播技術," 國立成功大學工程科學所碩士論文, 2001.
- [2] O. Babaoglu, H. Meling, and A. Montresor, "Anthill: a framework for the development of agent-based

- peer-to-peer systems," 2002, pp. 15-22.
- [3] Bittorrent, "<http://www.bittorrent.org/index.html>."
 - [4] B. Cohen, "Incentives Build Robustness in Bit Torrent," Workshop on Economics of Peer-to-Peer Systems, June 2003.
 - [5] P. Dasgupta, "Agent based peer-to-peer systems," 2002, pp. I-663-6 vol.1.
 - [6] Gnutella, "<http://www.gnutella.com/>."
 - [7] Kazaa, "<http://www.kazaa.com/us/index.htm>."
 - [8] E. P. Markatos, "Tracing a Large-Scale Peer to Peer System: An Hour in the Life of Gnutella," 2002, pp. 65-65.
 - [9] A. Mihalyi, "Optimizing peer-to-peer networks using mobile agents," 2004, pp. 194-199.
 - [10] Napster, "<http://free.napster.com/>."
 - [11] A. Puliafito, O. Tomarchio, and L. Vita, MAP: Design and implementation of a mobile agents' platform: Elsevier, 2000.
 - [12] S. Subhabrata, S. Oliver, and W. Dongmei, "Accurate, scalable in-network identification of p2p traffic using application signatures," in Proceedings of the 13th international conference on World Wide Web New York, NY, USA: ACM Press, 2004.
 - [13] L. Tie-Yan, Z. Zhi-Gang, and Y. Si-Zhen, "A-peer: an agent platform integrating peer-to-peer network," 2003, pp. 614-617.
 - [14] T.-I. Wang, "A Mobile Agent Carrier Environment with Mobile Computing Facilities," in IIP: International Conference on Intelligent Information Processing, The 16th IFIP World Computer Congress Beijing, 2000.