國立交通大學

資訊學院資訊科技（IT）產業研發碩士班

碩 士 論 文

以 P2P 技術為基礎之巨量發佈系統

– 群組發佈研究

The Study of the Massive Deployment System based on P2P Technology

– Grouped Deployment

研 究 生：吳冠翬

指導教授：吳毅成　教授

中 華 民 國 九 十 八 年 十 月

以 P2P 技術為基礎之巨量發佈系統

－ 群組發佈研究

The Study of the Massive Deployment System based on P2P Technology

－ Grouped Deployment

研 究 生：吳冠翬　　Student：Kuan-Hui Wu

指導教授：吳毅成　　Advisor：Yi-Cheng Wu

國 立 交 通 大 學

資訊學院資訊科技（IT）產業研發碩士班

碩 士 論 文

A Thesis

Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master

in

Industrial Technology R & D Master Program on

Computer Science and Engineering

Oct 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年十月

以 P2P 技術為基礎之巨量發佈系統
– 群組發佈研究

研究生：吳冠翬　　指導教授：吳毅成

國立交通大學資訊學院產業研發碩士班

摘要

最初由檔案下載所發展而來的同儕（Peer-to-peer, 或 P2P）發佈技術，為了適用於不同的應用領域，需要有所修正。以線上遊戲產品為例，遊戲更新檔通常有其先後順序，可以利用階段式下載的方式進入遊戲，可讓玩家更快進入遊戲。因此，我們可藉由加入檔案群組以及群組優先權機制，以滿足所需。

本系統目標是改良傳統 P2P 檔案發佈技術，針對檔案群組與優先權機制，提出一套解決方案。系統將以我們實驗室過去所開發之 Massive Deployment System（MDS） 系統為基本架構，加以改良。本論文將針對檔案分群與優先權機制進行研究與探討並分析效能。

The Study of the Massive Deployment System based on P2P Technology – Grouped Deployment

Student: Kuan-Hui Wu Advisor: Yi-Cheng Wu

Industrial Technology R & D Master on
Computer Science College
National Chiao Tung University

Abstract

This paper describes a new architecture to improve the download ability in massive deployment system for network gaming, which can provide a priority-based grouped deployment service. This new architecture system contains four components, such as, server, super seeder, host and client. This thesis focuses on the design of host, client and performance analysis.

誌謝

首先感謝我的指導教授吳毅成博士，給我相當多的指導，由於他寶貴的意見和指教，才能順利完成本論文。

再來要感謝實驗室的同學和學弟妹，因為有他們的陪伴，讓這兩年研究所生活中除了煩悶與壓力外，還多了歡笑和活力。

另外感謝群想網路科技股份有限公司，提供我實驗的數據，感謝群想全體員工的幫忙，尤其是庭嘉，提供了許多協助與意見，讓我的論文得以順利完成。

最後感謝我的家人，在求學過程中給予的照顧與關懷，特別感謝我的女友家茵，在這段過程中不論成功或失敗總是默默的支持我，給我撐下去的信心與動力，讓我沒有後顧之憂，謹以此文獻給所有關心我的人。

目錄

圖目錄

表目錄

第一章、緒論

本章會說明點對點同儕（Peer-to-peer, 或 P2P）目前在網路上的應用，並提出本論文的動機與目的，最後會說明整篇論文的大綱。

1.1 研究動機與目的

近年來，隨著寬頻網路環境的日益成熟，硬體設備的長足進步，促成更多網路應用。如高速、低延遲的網路，使得檔案發佈、網路電話服務、影音串流服務、大型多人線上遊戲等蓬勃發展，並成為現代家庭休閒育樂中不可或缺的角色。這些網路應用的發展，也帶給業者新的挑戰。大量的頻寬需求，造成使用傳統主從式架構（Client/Server）系統的壅塞，降低了系統的可用性。為了解決此問題，業界與學界開始引入點對點同儕架構，以提升系統應付高同時在線使用者的可擴充性。P2P 的技術，首先廣泛運用於檔案發佈。而在其他如影音串流與線上遊戲，則因為有不同的功能需求，需要進行改良。

以線上遊戲產品為例，在遊戲更新後，會再重新開放的瞬間產生大量頻寬需求，如何降低頻寬需求並減少玩家進入遊戲前的等待時間。是線上遊戲公司經常面對之問題。綜上所述，P2P 架構是目前業界與

學界用來解決主從式架構所面臨可擴充性問題的良好解決方案。

然而，最初由檔案下載所發展而來的 P2P 發佈技術，為了適用於不同的應用領域，需要有所修正。以線上遊戲產品為例，遊戲更新檔通常有其先後順序，可以利用階段式下載的方式，讓玩家更快進入遊戲。因此，我們可藉由加入檔案群組及群組優先權機制，以滿足所需。

本論文目標是改良傳統 P2P 檔案發佈技術。針對檔案群組與優先權機制，提出一套解決方案。系統將以實驗室過去所開發之 Massive Deployment System（MDS）[1] [2]系統為基本架構，加以改良。

1.2 論文章節說明

第二章將會介紹 P2P 相關技術背景以及發展現況；而第三章會對本系統的架構與功能運作方式進行介紹；第四章說明本系統實作方式；第五章展示實驗設計與結果的分析；最後第六章會提出本研究的結論以及未來工作與展望。

第二章、背景說明

本章節將介紹與本論文相關之各種 Peer-to-peer（P2P）的檔案資源發佈系統及 Massive Deployment System（MDS）[1] [2]系統架構介紹。首先會在 2.1 節介紹幾個著名的 P2P 檔案分享（file sharing）系統，並針對其中 BitTorrent[3] [4] [5] [6] [7]協定分享流程作較詳細之介紹，接著在 2.2 節介紹 MDS 系統之架構與元件說明。

2.1 P2P 檔案分享系統

隨著網際網路蓬勃發展，各種網路服務，不管軟體、硬體，皆有長足的進步。其中，最引人注目的軟體技術之一，即是點對點同儕（Peer-to-peer，或 P2P）架構，目的是用以取代或整合於主從式架構。P2P 架構主要核心概念在於將部分或全部頻寬與運算需求，轉移至使用者的電腦，從而降低主從式架構中伺服器的負擔。當使用者越多，可使用的資源也越多，因而可以有效解決主從式架構的可擴充性問題。

P2P 架構的概念雖然早在 1969 年即見於文獻上，但其廣泛應用可從音樂共享軟體 Napster 在 1999 年的興起來視為開端。當然，除了檔案分享的應用外，業界與學界也不斷尋找利用 P2P 架構改善系統

可擴充性問題的機會。目的是為了能讓 P2P 架構更適應企業既有軟硬體資源。

一般來說 P2P 架構所指的是用戶端與用戶端的能力相等，所擁有的能力就是用戶端自己本身原本擁有的能力。相較於主從式（Client/Server）模式，P2P 模式即表示每個用戶端同時扮演客戶端與伺服器兩種角色，每個存在於同一個 P2P 架構中的節點（Peer）皆是如此。

P2P 架構大致分為二類，分別介紹如下：

- 純粹 P2P

  這種模式下的點對點架構，並沒有中央伺服器，各個用戶端與用戶端之間的資訊傳輸屬直接對傳，知名的 Gnutella[8]專案即是此模式。

- 混合 P2P

  此種模式最大的不同之處，是多了中央伺服器，伺服器中可能會保存各個用戶端連線的資訊概況、用戶端欲分享的檔案或其他伺服器的位址…等，端看該點對點架構是如何實作的。

上述二種模式，互有利弊，視應用情形套用。就現在市面上的點對點架構來看，混合型點對點架構有較多專案採用，如 BitTorrent、eMule[9]。本論文所開發之系統即是一種混合 P2P 模式的應用。

2.1.1 Napster

1999 年由 Shawn Fanning 所創造，是第一個被大眾所使用的音樂分享服務。Napster[10]的架構很簡單，用戶端上線後即會連至一伺服器，目的就是在伺服器端留下「擁有檔案清單」，若是某個用戶端希望下載檔案時，會將該檔案關鍵字發送至伺服器，伺服器再回傳哪些用戶端擁有這個檔案。

可惜的是，在 2001 年七月，因為版權等問題，Napster 在法庭上敗訴，被迫關閉伺服器，服務被迫終止。

2.1.2 KazaA

在 Napster 結束服務之後，另一個名為 KazaA[11]的服務竄起，相較於 Napster，其不同點在於 KazaA 並沒有中央伺服器，而是導入了超級節點(Super Node)的概念，其他並非超級節點的用戶端，在 KazaA 中，就稱作普通節點(Ordinary Node)，因為沒有中央伺服器，用戶端連上 KazaA 網路時，會先連至一超級節點，跟 Napster 一樣，用戶端

也會跟超級節點提供「擁有檔案清單」與「欲下載檔案關鍵字」來進行檔案傳輸。

值得一提的是，超級節點與超級節點之間會互相分享所有擁有的用戶端資訊，如此的好處為，並不會因為用戶端所連至之超級節點失效無法繼續服務後，而自 KazaA 網路離線，用戶端可連線至其他超級節點。可惜因為當時 KazaA 對於檔案驗證的機制薄弱，導致於有心人士利用此一弱點散布許多假檔，從 KazaA 網路上下載的檔案，大部分都會下載到名不符實的檔案，使得 KazaA 逐漸淡出市場。KazaA 的開發團隊日後利用獨特的超級節點技術，研發出了知名網路通訊軟體 Skype[12]，號稱當網路頻寬足夠時，能夠提供比傳統電話還清晰之音質，也確實如此。

2.1.3 eMule

eMule 的前身是 eDonkey，最大的特色是能夠將檔案切割為多個小片段，以利充分利用網路資源，並且改善了 KazaA 關於檔案驗證方面的弱點，利用 MD4 數學運算，針對每一個分享檔案做了識別運算，稱之為檔案切細值。雖然 eMule 已經將檔案切割成小片段，但是他擁有續傳功能，所以同一個分享檔案，可能是由多個用戶端提供而組成，大大地提昇了傳輸的效率。

eMule 最主要得問題則是使用者還是必須連線至一台 eDonkey 相容伺服器，以進行資源搜索，當伺服器可能因為法律、政策、網路…等因素，被迫服務終止時，eMule 的下載將會受到極大的阻礙。

2.1.4 BitTorrent

BitTorrent 可說是目前世界上最流行的 P2P 檔案資源下載軟體，在其架構中，有一個特殊的 Tracker 角色，上面擁有某使用者所提供的 Torrent 檔案資訊，該使用者在將此 Torrent 檔案散佈給其他使用者，其他用戶端即可透過此 Torrent 檔案，利用其中所包含的資訊進行檔案下載。

BitTorrent 系統[13]中，主要元件定義如下：

- Files — 所欲分享發佈的檔案資源，可是單一檔案，或多層目錄檔案結構。例如遊戲公司某款線上遊戲產品。
  - Piece — BitTorrent 系統會將原始檔案切割為多個 Piece 片段，以利交換。當 Client 擁有所有的 Piece 後，才能夠擁有完整的檔案。
  - Block — 在 BitTorrent 系統實際傳輸上，檔案的傳送，是以比 Piece 更小的 Block 來傳送。當 Clinet 與有某 Piece 之

所有 Block 後，才算是完整擁有此片 Piece。

- Tracker — 提供某個 BitTorrent 系統中，用戶端的連線名單，使用戶端可以透過此名單，與其他用戶端相連並交換檔案片段。

- Metainfo File — 為 BitTorrent 系統針對發佈檔案所產生之簽章，其中包含了欲下載此發佈檔案之相關資訊，如 Tracker 位址、發佈檔案大小、發佈檔案分割成片段之 hash 值、片段大小、註解欄位等。

- Peers — 即是整個 BitTorrent 系統中，所有用戶端的統稱，又分為以下兩個角色。
  - Client — 尚未擁有所有片段（完整檔案）的用戶端。
  - Seeder — 擁有所有片段（完整檔案）的用戶端。

2.2 BitTorrent 系統發佈/下載檔案資源之流程

2.2.1 Metainfo File 發佈流程

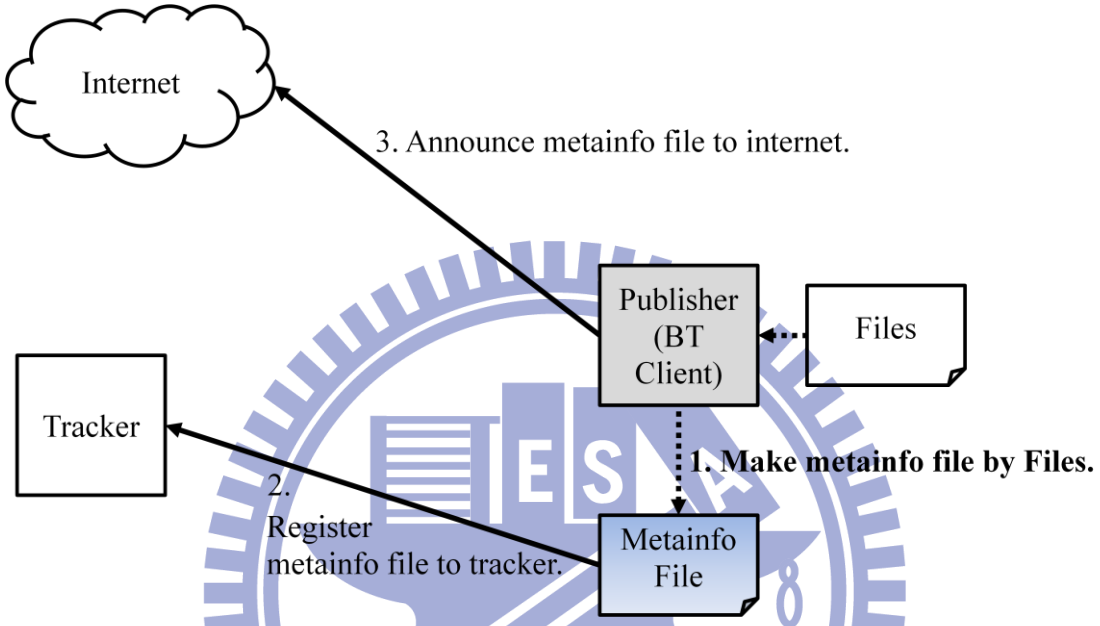

圖 1：Metainfo File 發佈流程步驟一

首先檔案資源擁有者，欲發佈資源時，會先將發佈資源讀入，並且依照 BitTorrent Protocol 製作出一 Metainfo File（Torrent File），如圖 1 所示。

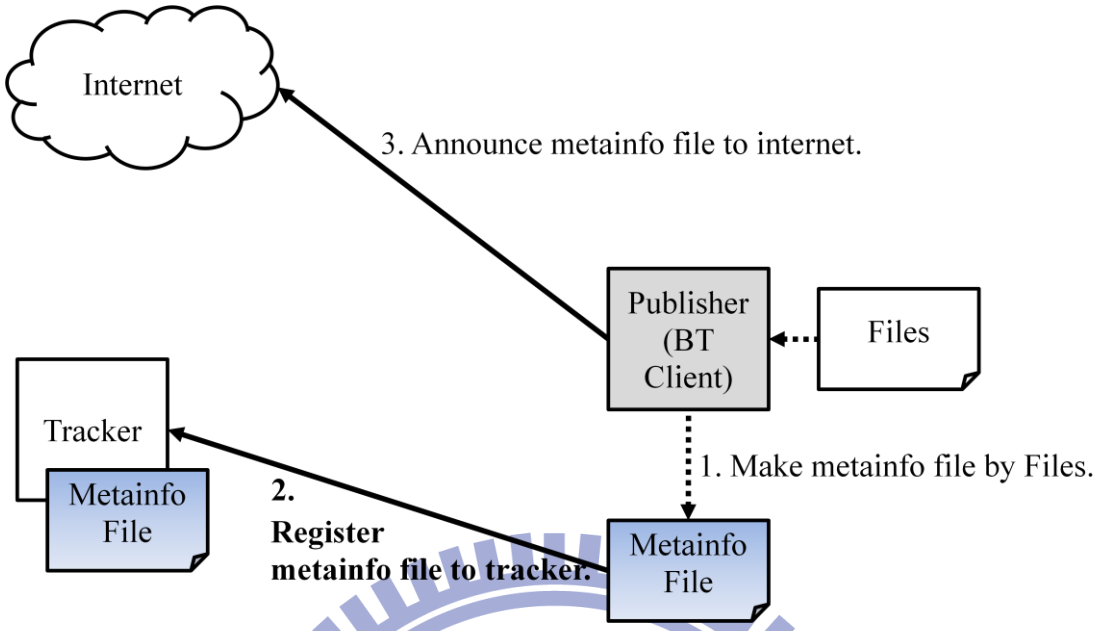

圖 2：Metainfo File 發佈流程步驟二

接下來檔案資源擁有者會向 Tracker 註冊，告知已產生一 Metainfo File，Tracker 就會保有一份此 Metainfo File 資訊，如圖 2 所示。

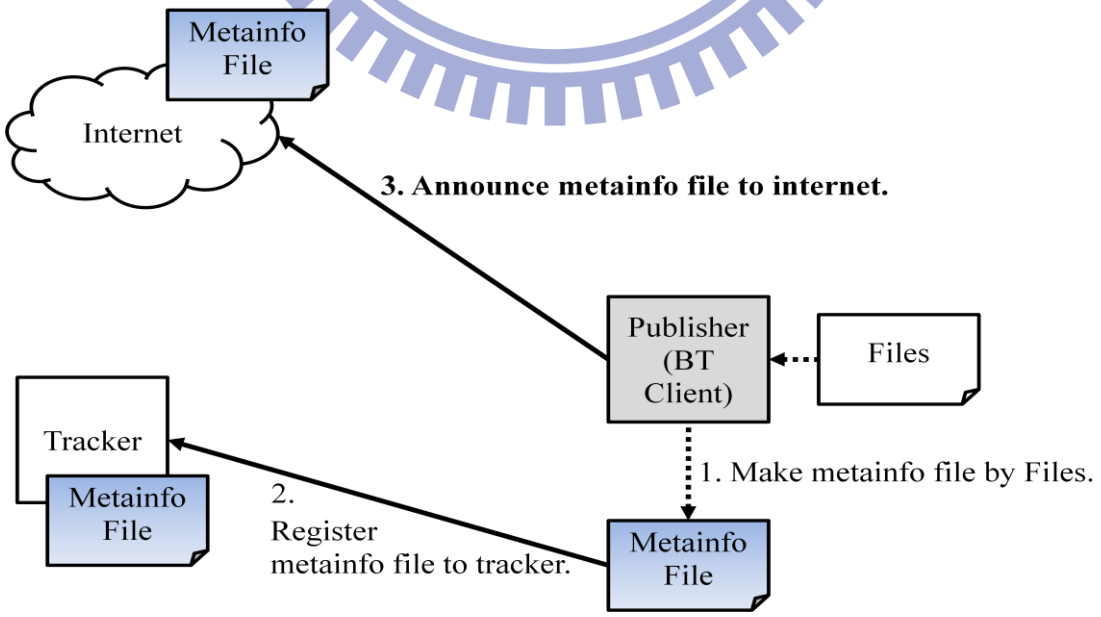

圖 3：Metainfo File 發佈流程步驟三

然後檔案資源擁有者就會向網路散佈所產生之 Metainfo File，其他用戶端即可透過網際網路下載此 Metainfo File，如圖 3 所示。

透過以上流程，即發佈了一 BitTorrent Metainfo File（Torrent File）至網際網路上，其他 BitTorrent 用戶端，只要取得了該 Metainfo File，即可與 Tracker 連線，取得 Seeder 與 Client 的位址清單，並與其進行連線，開始此 Metainfo File 中所紀錄檔案之下載。

2.2.2 BitTorrent Metainfo File 格式

announce	http://cycgame.com
comment	Demo Metainfo File
created by	Enhanced-CTorrent/dnh3.3.2
creation date	1253633301
info_hash	b59113d20717ff5986b9b397810edcf582d1f8d9
info	files: 0 → length: 344163, path: 0 → 啟動模組.exe; 1 → length: 320024, path: 0 → 核心模組.module
	name: 遊戲產品
	piece length: 262144
	pieces: 0 → ���C��JK��m�`[!5 1 → C��nX�Wo�iqZ�0 2 →  rboPMbh  L  J 

表 1：BitTorrent Metainfo File 格式

Metainfo File（Torrent File）由 Bencode 編碼組成，以下介紹幾個常用欄位。

- **announce**：記錄 Tracker 的位址
- **comment**：此 Metainfo File 的註解，可由 File 製造者自行填寫
- **created by**：記錄此 Metainfo File 的製造者資訊

- **creation date**：記錄此 Metainfo File 的製造日期
- **info_hash**：此 Metainfo File 之 Hash 值
- **info**：內含多種資訊，包含所欲發佈檔案之檔案結構、片段大小與各個片段之 Hash 值

2.2.3 BitTorrent 基本連線架構

BitTorrent 基本連線架構分為以下兩類，如圖 4：

- Tracker HTTP Protocol（THP）

  此為 Peer 與 Tracker 之間的連線，採用 HTTP 連線方式。

- Peer Wire Protocol（PWP）

  此為 Peer 與 Peer 之間的連線，採用 TCP 連線方式。

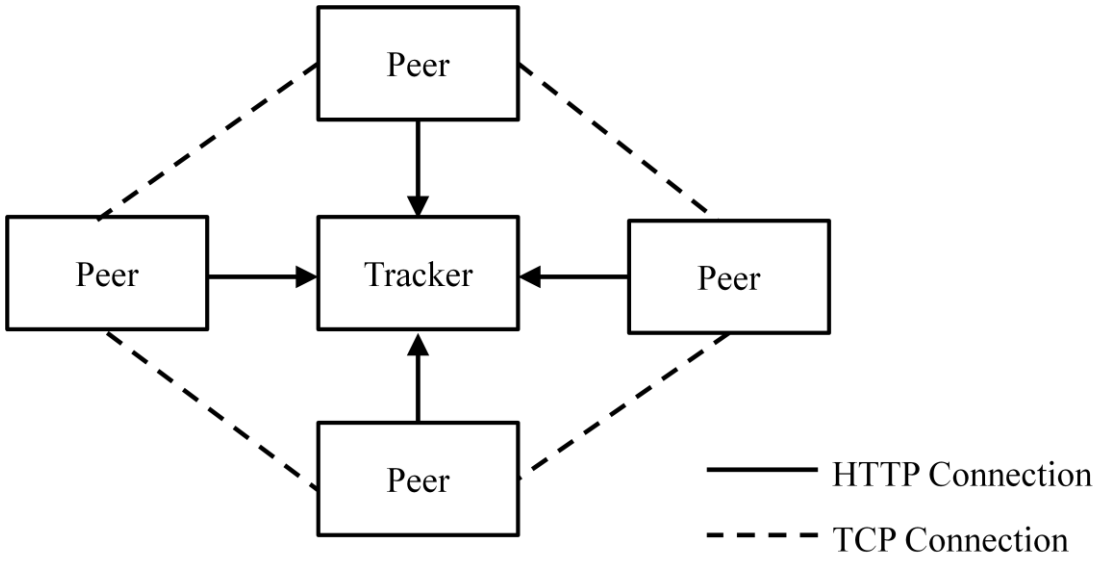

圖 4：BitTorrent 基本連線架構

在經過以上所述之發佈 Metainfo File 流程，Peer 取得欲下載資源檔案之 Metainfo File 後，便可透過其中所記載的 Tracker 位址，利用 Tracker HTTP Protocol 與 Tracker 連線，如圖 5。連線後便可取得 Peer List，如圖 6。再透過 Peer List 中所記載之其他 Peer 位址，利用 Peer Wire Protocol 與其他 Peer 相連，接著進行一連串的片段交換，逐漸完成資源檔案之下載，如圖 7。

以下流程圖將說明上述步驟：

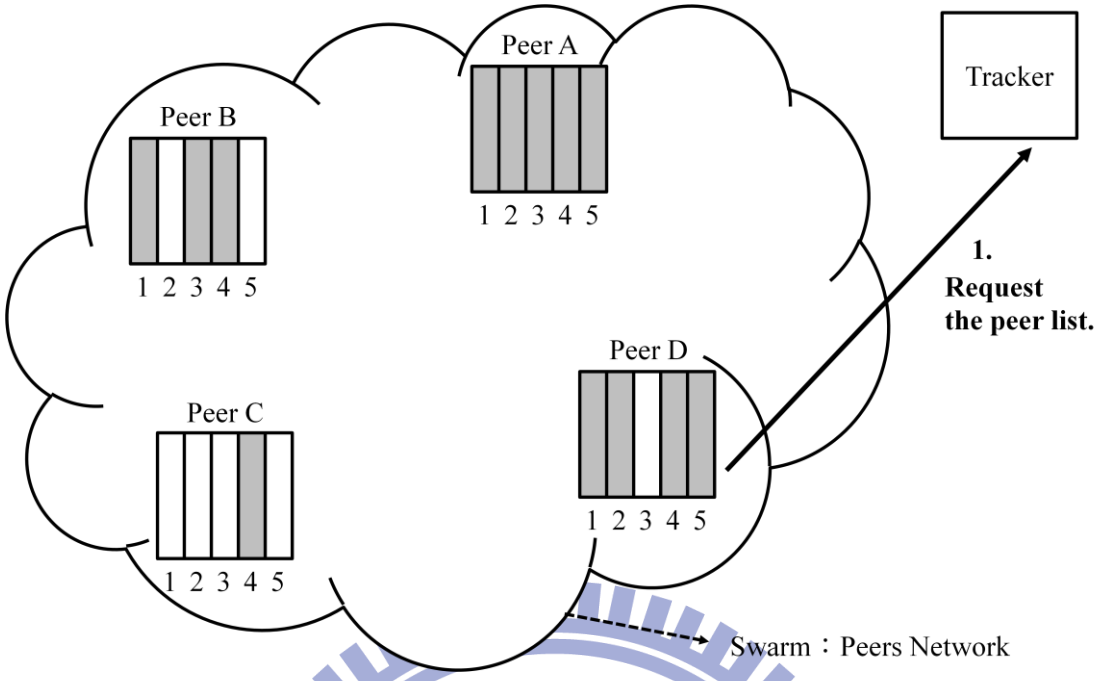

圖 5：Peer D 透過 Metainfo File 連至 Tracker，請求 Peer List 連線清單

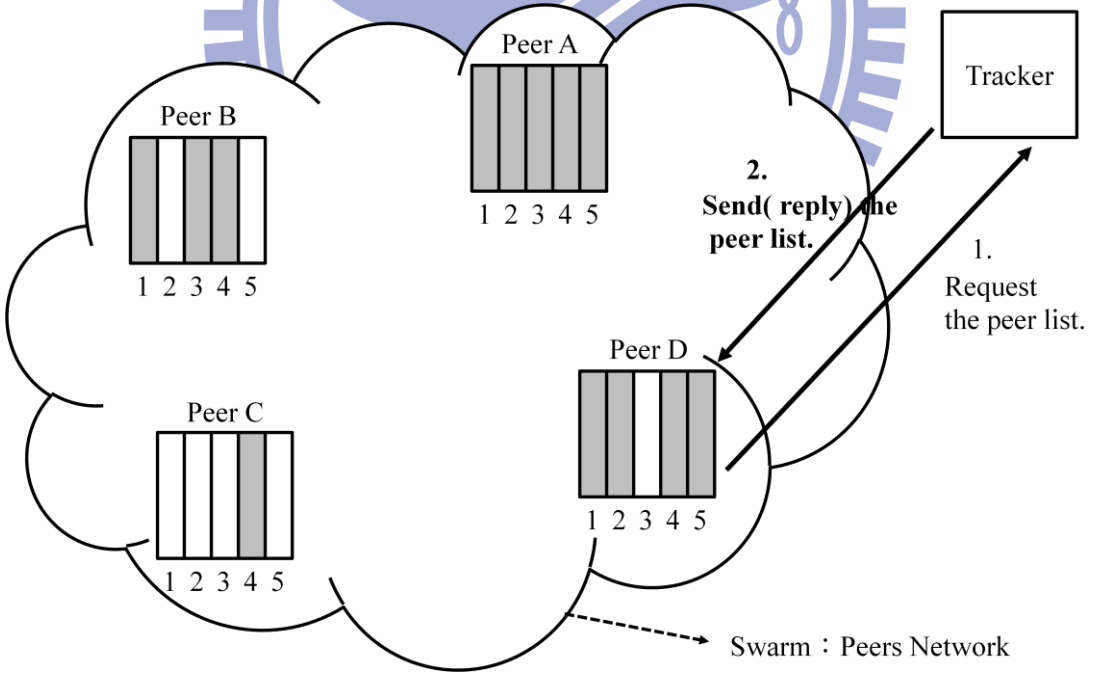

圖 6：Peer D 自 Tracker 處取得 Peer List 連線清單

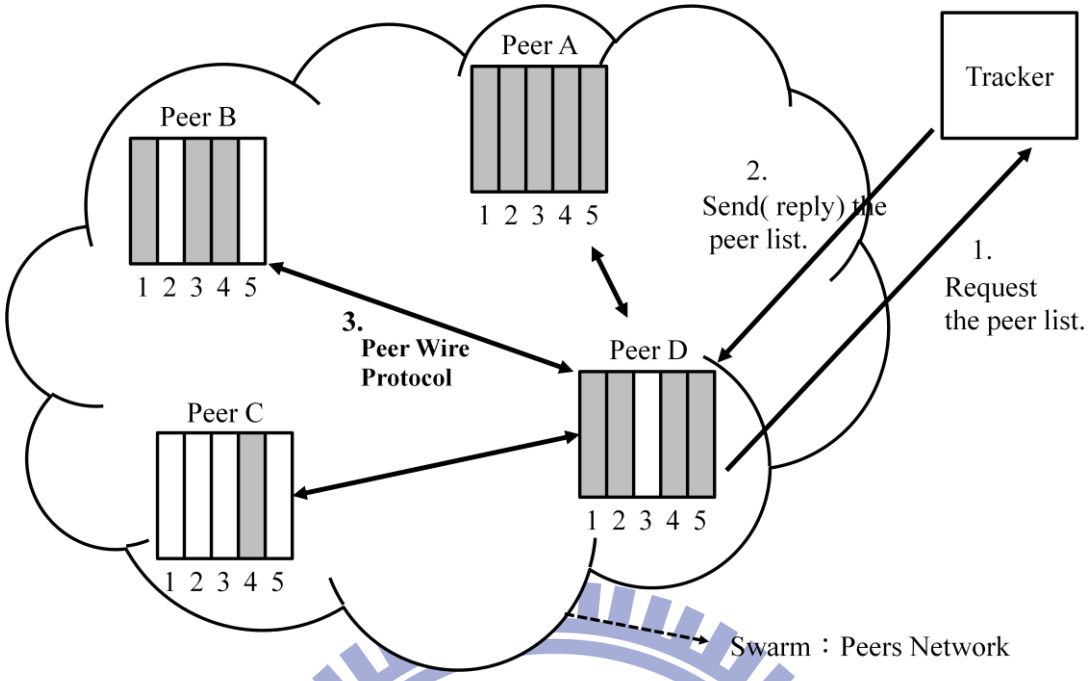

圖 7：Peer D 透過 Peer List 用 PWP 協定與其他 Peers 互連並交換片段

經由以上步驟，Peer D 已可與其他同屬同個 Peers Network 的 Peers 進行檔案片段的交換，已達成檔案資源下載的目的。

2.2.4 Peer Wire Protocol

本節將詳細描述 BitTorrent 系統中，Peer 與 Peer 之間的通信協定（PWP），PWP 是架構在 TCP 連線之上，裡面包含了幾種不同的訊息，讓 Peer 之間可以互相建立連線、交換片段資訊、交換狀態資訊，以便於檔案片段的交換，以下就是 PWP 中主要的訊息：

2.2.4.1 Handshake

圖 8：Handshake message

Peer 要與其他 Peer 第一次進行連線時，會進行 Handshake message 的傳遞。其主要目的是確認以下兩件事情：

- 雙方是否採用相同 Protocol，在此是 BitTorrent Protocol
- 彼此是否針對同一個 Metainfo File 檔案資源進行下載

待 Handshake 完畢之後，Peer 與 Peer 間即完成連線建立。

2.2.4.2 Bitfield

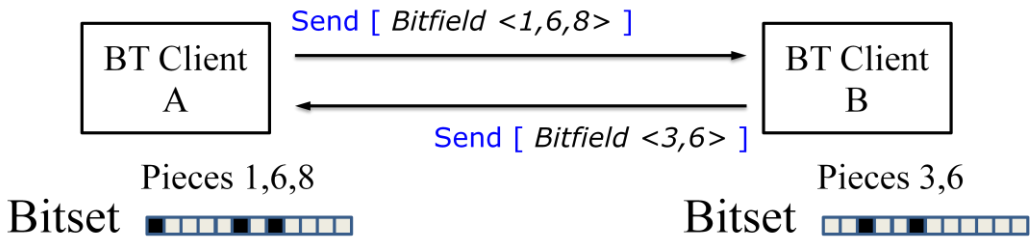

圖 9：Bitfield message

當 Peer 與 Peer 間進行 Handshake 完畢後，緊接著就會傳遞 Bitfield message，目的是為了告知對方自己所有擁有的片段資訊，自己也會收到對方所傳遞過來之片段資訊。

2.2.4.3 Interested (Uninterested)

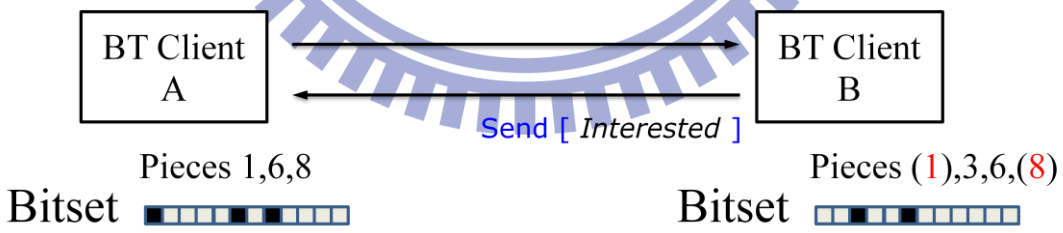

圖 10：Interested (Uninterested) message

藉由對方傳送的 Bitfield message，Peer 發現對方有自己所沒有之片段時，會發送 Interested message 給對方，反之則是 Uninterested message，Peer 狀態不改變，則不重新發送此訊息。

2.2.4.4 Choke (Unchoke)

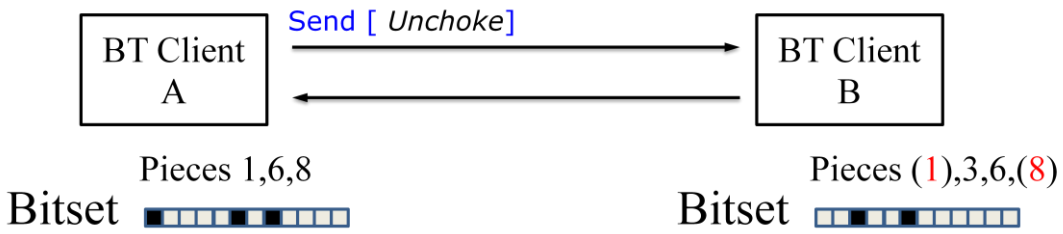

圖 11：Choke (Unchoke) message

Peer 會自所有對自己 Interested 的對象中，挑出欲上傳之對象，且對他發送 Unchoke message，於下所有連線對象，皆發送 Choke message。Peers 上傳對象狀態不改變，則不重新發送此訊息。

2.2.4.5 Request

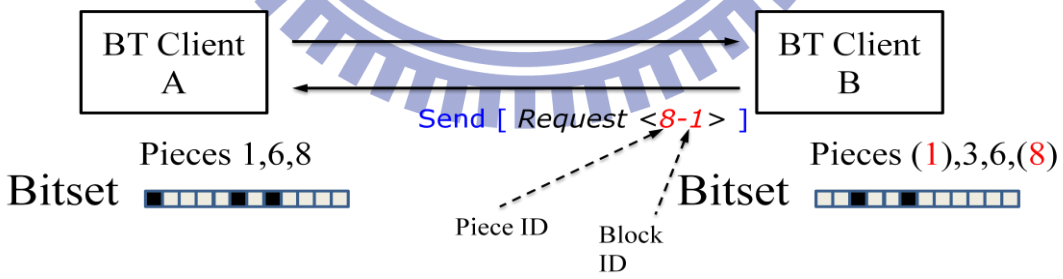

圖 12：Request message

當一 Peer 被 Unchoke 時，即代表可向對方發送 Request message，索取想要下載的子片段。

2.2.4.6 Piece

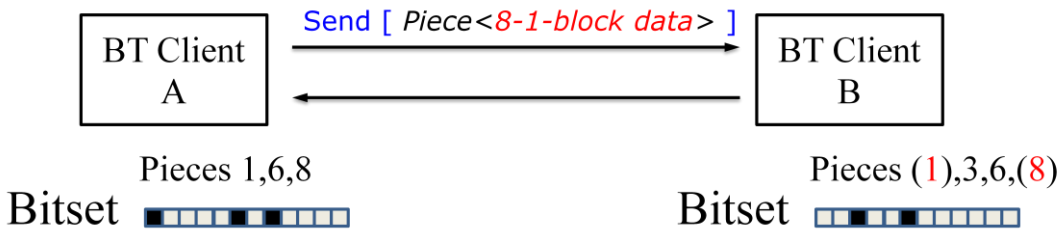

圖 13：Piece message

當收到 Request message，且對方也是在自身之 Unchoke 名單中時，Peer 就會將 Request message 所要求之子片段內容透過 Piece message 送給對方。此訊息雖名為 Piece，但實際上所傳遞之內容是子片段（Block）資訊。

2.2.4.7 Have

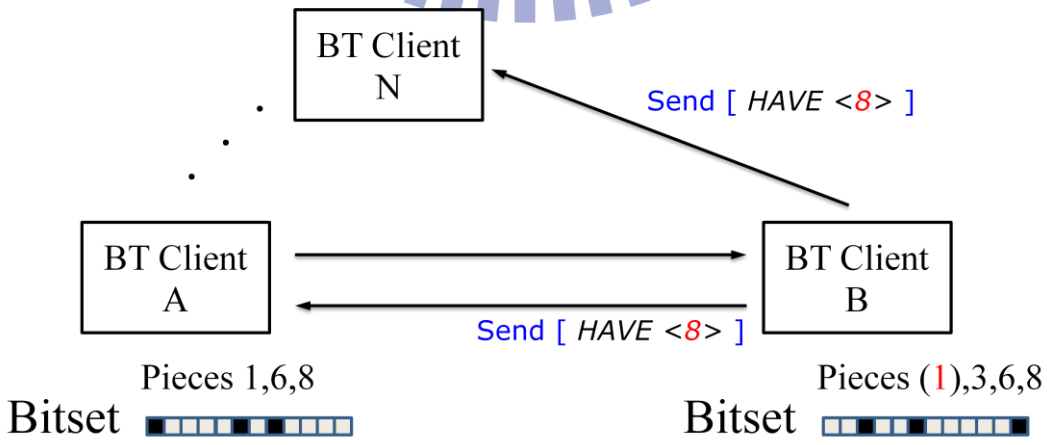

圖 14：Have message

當 Peer 透過多次 Request message 要求所需子片段後，待收集到一完整片段資訊後，會將此片段之 Hash 值與 Metainfo File 中所記錄之 Hash 值進行比對，若無誤，則代表完成此片段的下載，即會針對所有連線對象發送 Have message，通知對方自身已經擁有此片段。當 Peer 收到連線對象所送來之 Have message，也會將其中所記載之片段資訊更新至該 Peer 之片段資訊中，如此，就可知道自己所有連線對象中，完整的片段資訊。

由以上所描述之 Peer Wire Protocol 可了解到 Peer 與 Peer 之間事如何建立連線，與交換片段／狀態資訊，最後完成資源檔案下載。

第三章、系統介紹

本章節介紹 Massive Deployment System（MDS）[1] [2]系統之架構，本論文研發之系統是由 MDS 系統修改而來，在 3.1 節中，將介紹各個 MDS 系統之元件。

3.1 MDS 系統架構

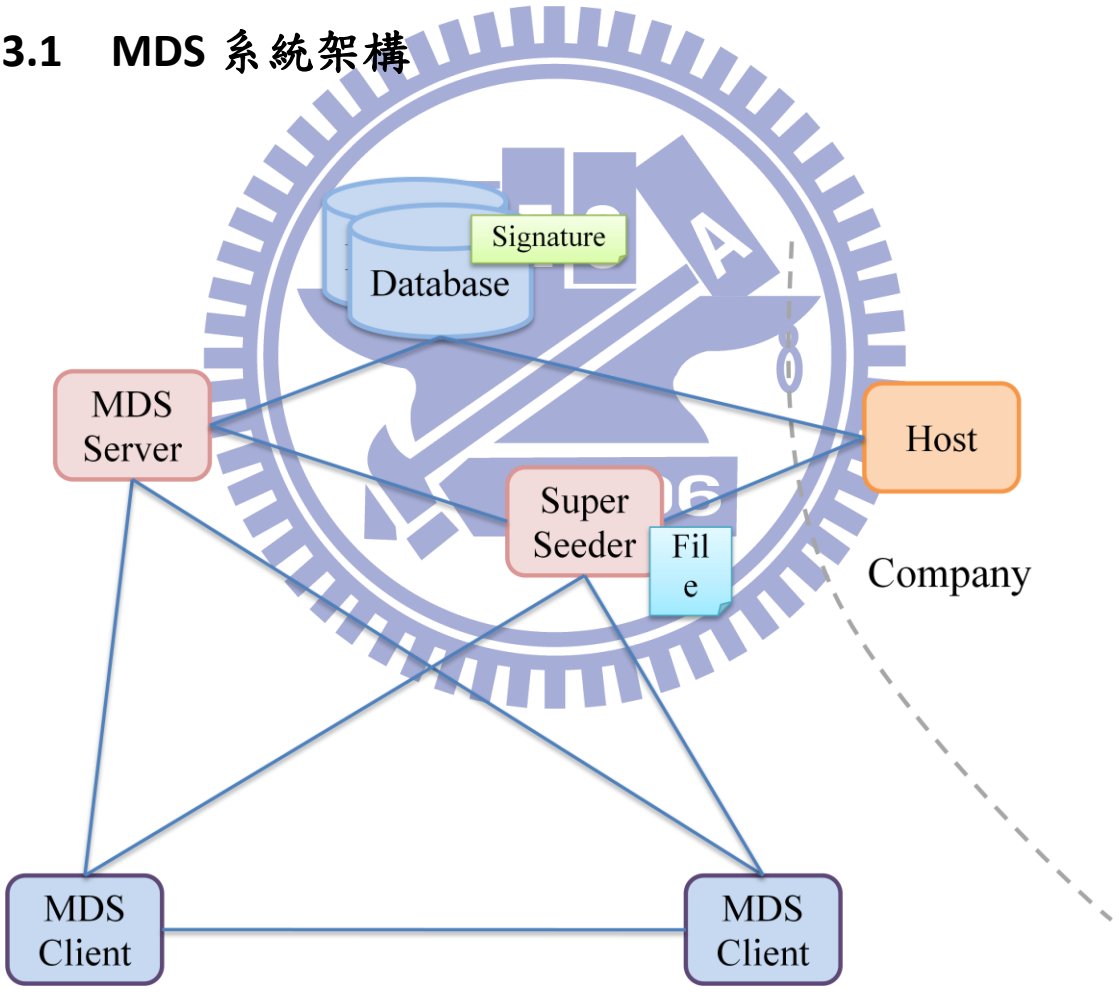

圖 15：MDS 系統架構

架構圖中虛線右邊為公司管理元件 Host，左邊則為 MDS 核心元件，或公司外部伺服主機。各元件說明如下：

- **File：**

  企業欲發佈之資源檔案，如同 BitTorrent Protocol 規定，可為單一檔案或資料夾結構。

- **Signature：**

  為 Host 針對 File 所製作之簽章，格式與 BitTorrent 之 Metainfo File 相同，內含資源檔案目錄結構、片段 Hash 值與其他相關參數。

- **Host：**

  由公司內部產品管理部門所操作。其功能如下：

  - 製作 Signature。
  - 傳送 Signature 至 Database。
  - 傳送 File 至 Super Seeder。

- **MDS Server：**

  為 MDS 架構中的核心元件，其功能為：

  - 協調 Super Seeder 與 MDS Client。
  - 傳送 Signature 給 Super Seeder 與 MDS Client。
  - 傳送 MDS Client 連線清單至 MDS Client。
  - 所有連線對象之狀態監控。

- **Super Seeder：**

  MDS 架構中，最先擁有完整發佈檔案之元件，其功能為：

  - 擁有完整的 File。
  - 接受 MDS Client 的 File 片段請求，傳送 File 片段至 MDS Client。

- **MDS Client：**

  擁有檔案片段交換能力之用戶端，其中通訊息定仿造 BitTorrent Protocol，其功能為：

  - 向 Super Seeder 要求 File 片段。
  - 與其他 Client 交換所擁有的 File 片段。

- **Database：**

  在 MDS 中扮演資料儲存的角色，主要功能為：

  - 儲存 Signature 檔案。
  - MDS Server、Super Seeder 與 MDS Client 連線資訊。
  - 系統記錄與各個元件狀態報告。

第四章、系統實作

本論文之系統是由 Massive Deployment System（MDS）[1] [2]系統修改而來，主要修改了 Host 與 MDS Client 兩個元件，使其支援檔案分群與優先權機制。

4.1 傳統 BitTorrent 檔案結構

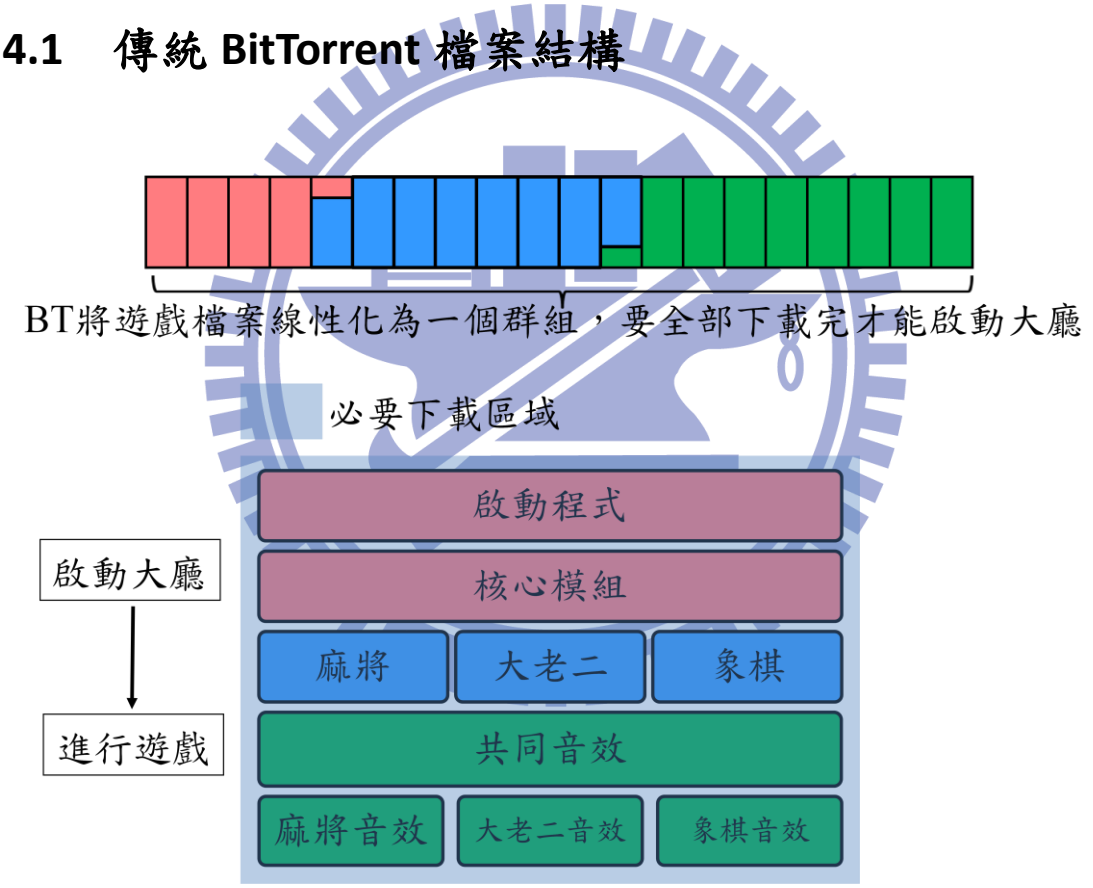

圖 16：傳統 BitTorrent 檔案結構

傳統的 BitTorrent 檔案結構，在製作 Metainfo File 時，會將所有檔案線性化成一個群組，意思是每個檔案的後段不滿一個片段的地方，

會與下一個檔案的前段相接，如圖 16 所示，以合作企業，群想網路科技股份有限公司為例，利用傳統 BitTorrent 製作的 Metainfo File，導致各個模組檔案被分割成片段時，彼此頭尾相接，當麻將玩家欲下載麻將遊戲，在這種情況，必須將所有檔案，包含非麻將所需部分皆下載，才可進行遊戲，萬一總體遊戲檔案體積龐大，下載將曠日廢時，這是採用過去 MDS 系統所遭遇的問題。

4.2 系統檔案結構

如同上節所述，我們希望在 MDS 系統所產生的 Signature 檔案中使檔案分群。當遊戲公司將各個模組之間的關係定訂出來後，本論文所改良之 Host 即可以依照其規定之關係，將檔案分群，這邊採用的作法是，群組後端不足一片 Piece 的部分，利用一個以「0」值填滿之無意義檔案補足，並賦予其一不與其他群組重複之檔名。如此，原本的群組檔案加上新增的無意義檔案，必可以被所定義片段大小整除，不會與下一個群組的檔案一同分割在同一個片段中。

下圖即是系統檔案分群之圖例說明：

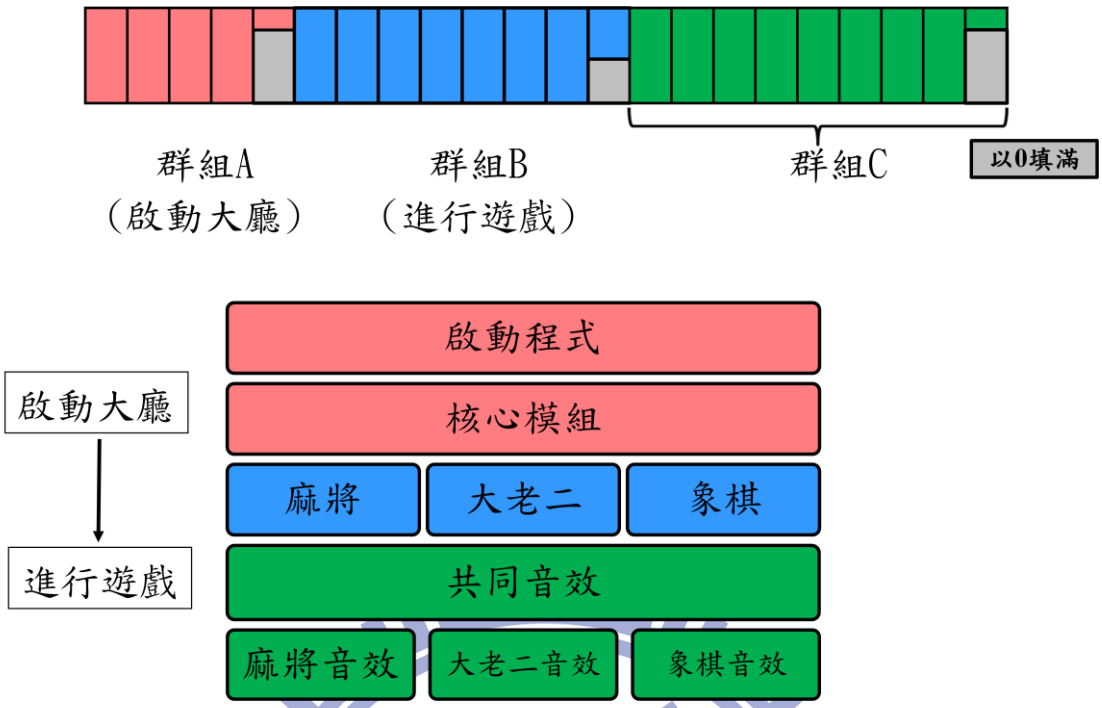

圖 17：檔案分群

圖 17 中所描述內容為，同顏色之模組即同一群組，可見三群，分別為群組 A、群組 B 和群組 C，每個群組不足一個片段部分，由無意義檔案補足（灰色區塊）。

4.3 檔案群組與優先權

因為遊戲檔案的載入，通常會有模組區分，且有其順序性，若是先載入必要模組，玩家即可進入遊戲。但是採用 MDS 系統來進行檔案發佈時，除了所有檔案線性化為一個群組之外，片段的下載基本上是採取隨機挑選下載，玩家無法預期何時可以進入遊戲。因此，在將

檔案分群之後，系統會針對各個檔案增加優先權設定，如下圖所示。

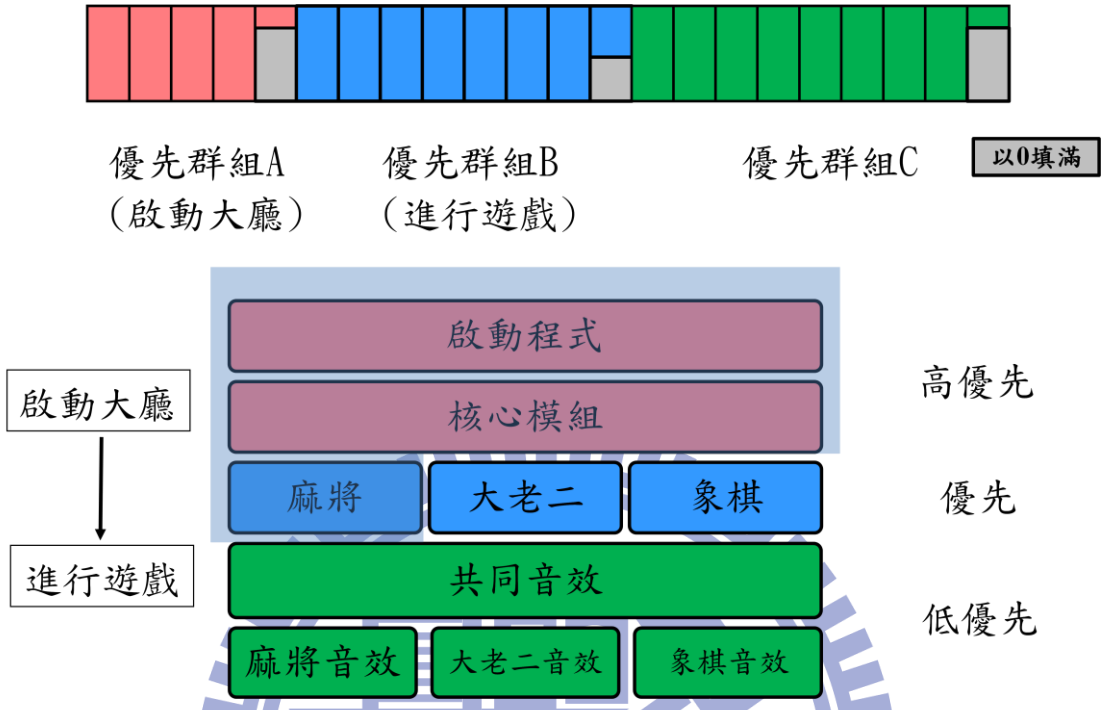

圖 18：檔案群組與優先權

如圖 18 例子，以群想麻將玩家為例，遊戲分為兩階段，先進入大廳，再進行遊戲，以啟動大廳來說，必要模組有「啟動程式」、「核心模組」，待大廳啟動，麻將玩家只需要再下載「麻將」模組，即可進入遊戲，因此對該玩家來說，不管總體發佈檔案體積多大，只要先行下載「啟動程式」、「核心模組」、「麻將」這三個模組，就可進行遊戲。

4.4 優先群組與遊戲更新檔

遊戲公司經常會針對遊戲 bug 進行修正，以期提高玩家進行遊戲的品質，若是使用 MDS 系統發佈檔案，每一次發佈更新檔後所製作之 Signature，對於 MDS Client 來說，都是一份全新的下載。導致遊戲下載變的非常沒有效率，如果只需下載更新檔，對於玩家來說，遊戲體驗將不會受到太多影響。如下圖所示。

圖 19：遊戲群組與更新檔

以「麻將」模組更新為例，玩家只需下載麻將模組所在之片段即可，其他群組則不受到影響，因為系統已將檔案分群，就算所在片段

位置與前次 Signature 不一樣，但是新版 Signature 中所記載之 Hash 值是相同的，不會對於檔案的驗證造成影響。

下表是系統所產生之 Signature 欄位狀態。

announce	http://cycgame.com
comment	啟動模組.exe:90:核心模組.module:80
created by	Enhanced-CTorrent/dnh3.3.2
creation date	1253096861
info_hash	f075fca13c5533e1948a78067cc393eee8492179
info	files: 0 {length: 344163, path: 0: 啟動模組.exe}; 1 {length: 180125, path: 0: 啟動模組.exe_dummy_}; 2 {length: 320024, path: 0: 核心模組.module}; 3 {length: 204264, path: 0: 核心模組.module_dummy_} name: 遊戲產品 piece length: 262144 pieces: 0 ���C��JK��m�`[!5 1 ��□□□□□□□:g����G5 2 3 ?? ???Q!?????

表 2：Signature

4.5 Signature 發佈流程

本節將介紹系統中 Signature 發佈過程。Signature 是由 Host 製作，發佈流程基本上與原本之 MDS Host 沒有太大差別，只是在本系統中，Signature 包含了群組優先權之資訊。

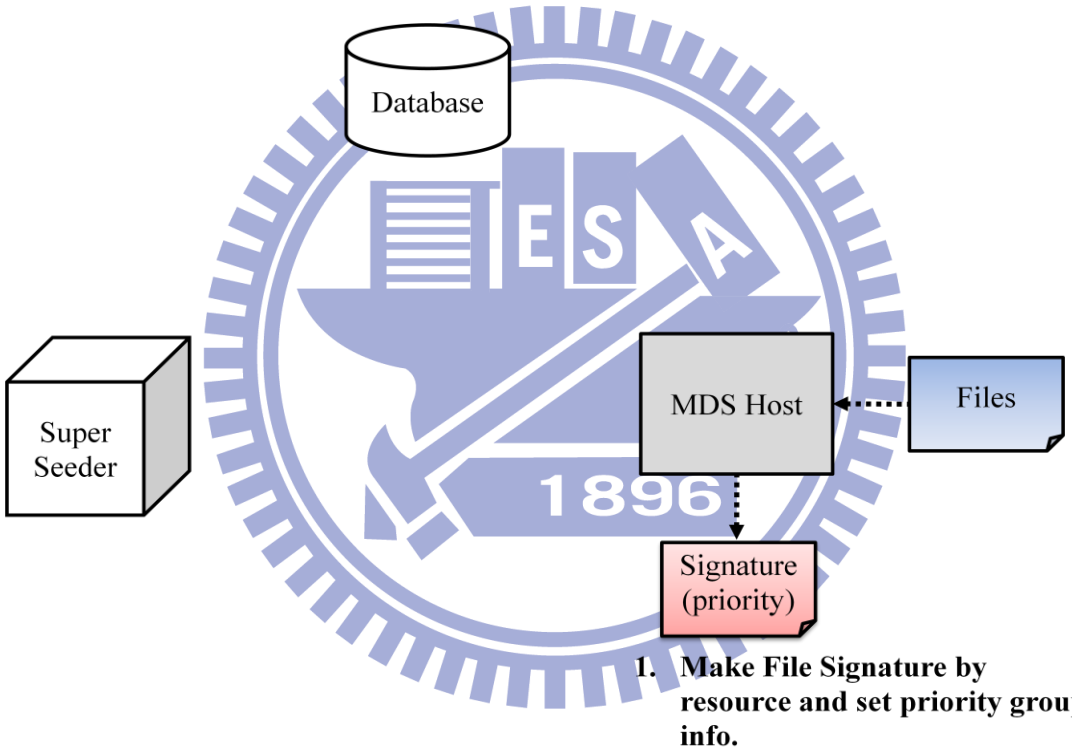

圖 20：Signature 發佈流程步驟一

圖 20 描述 Host 讀入所欲發佈檔案，除了依照 BitTorrent Protocol 製作 Signature 之外，其中也包含群組優先權之資訊。

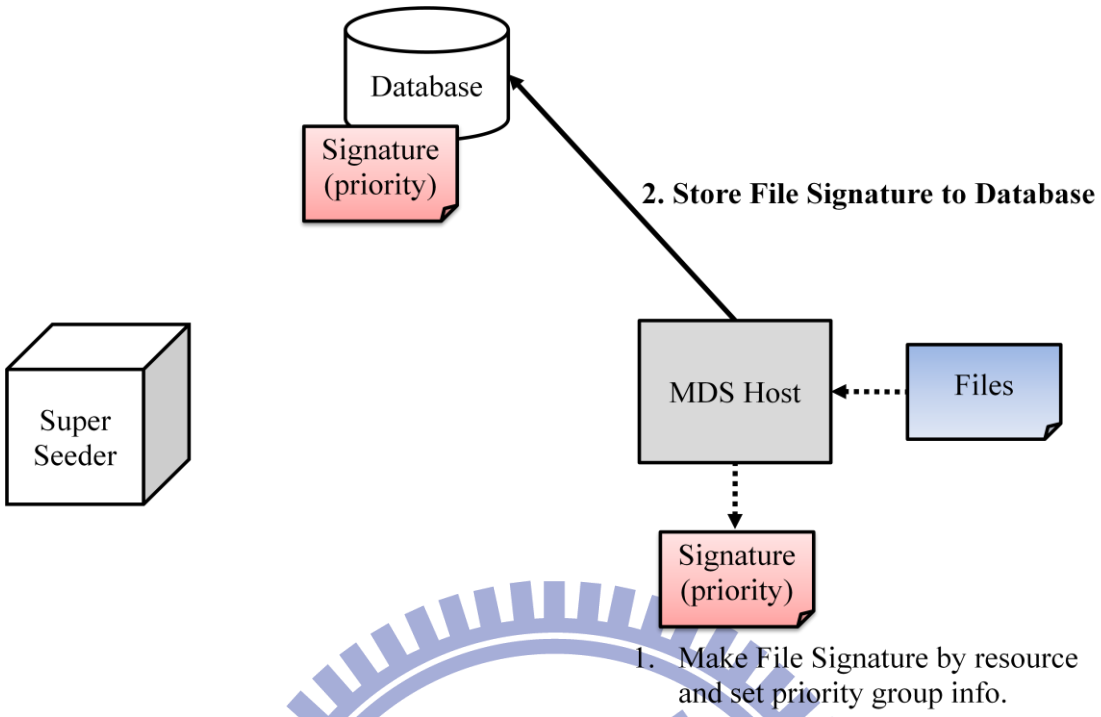

圖 21：Signature 發佈流程步驟二

圖 21 描述 Host 製作 Signature 完畢之後，將之送至 Database 中儲存，如此 Database 中即會保有一份此 Signature 之相關資訊。

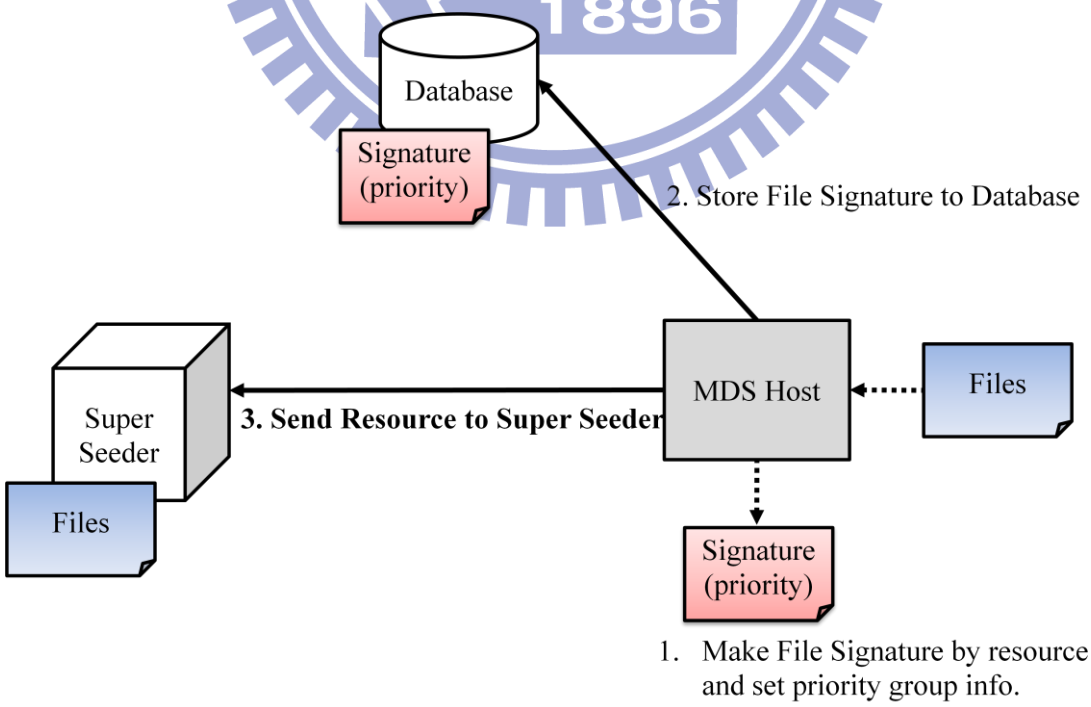

圖 22：Signature 發佈流程步驟三

圖 22 是描述 Signature 發佈的最後一個步驟，Host 會將所欲發佈檔案（File）傳送至 Super Seeder 處。如此 Super Seeder 才可發送檔案片段給用戶端。

4.6 Client

本系統之 Client 基本上是由 MDS Client 修改而來，目的是能夠支援讀取 Signature 所加入之群組優先權資訊，並且支援優先權下載。並且能夠在所規定之優先權群組下載完畢後，進行其他動作或中斷下載，如此才可使玩家執行所優先下載之遊戲程式。

Client 優先權下載之規則，大致分為下列三點：

- 依據優先權設定依序下載檔案群組：

  當各個群組之優先權設定皆無重複時適用。

- 相同優先權之群組，則視為同一群組：

  即使在製作 Signature 中，將兩相同優先權之群組分群，為使下載行為單純，在 Client 中，相同優先權之群組，視為同一群組，片段的選擇，並無先後之分，但已盡量完成同一群組為優先考量。

- 無優先權之群組，待優先權下載完畢，再隨機選擇下載：

  若是有無指定優先權之群組，待所有具有優先權之群組下載完畢後，再隨機選擇片段下載。

第五章、實驗結果

本章節將會進行本系統之實驗，並針對實驗結果進行分析[14]。

5.1 實驗環境設定

本論文的實驗環境，將以群想網路科技公司所提供之樣本[15]為基礎，群想網路提供環境參數如下表：

環境參數	設定值
玩家同時連線人數	1000 人
伺服器上傳頻寬	10 Mbps
總體遊戲檔案大小	160 MB

表 3：群想網路科技提供環境參數

為順利進行實驗，我們將上表參數約略縮小 10 倍，並增加一些其他必要環境參數，比如玩家 ADSL 速率，玩家 IP 位址形態。並為了實驗本研究所提出之檔案分群與優先權機制，總體檔案大小部份，會隨著實驗不同，而有不同的參數設定。

環境參數	設定值
模擬玩家同時連線人數	100 人
模擬伺服器上傳頻寬	1024 KB
模擬總體遊戲檔案大小	16 MB
模擬玩家網路頻寬環境	下載 256 KB/Sec, 上傳 64KB/Sec
模擬玩家 IP 位址形態	皆設定為 Public IP

表 4：模擬實驗環境參數

5.2 實驗設計

本論文實驗主要是探討檔案分群與優先權機制，在網路遊戲中，是否具有優勢，因此會藉由模擬遊戲公司提供遊戲檔案給玩家下載之行為，待玩家下載完畢，記載下載完畢之時間點，所有實驗中，玩家人數皆為 100 人，伺服器上傳頻寬皆為 1024 KB，如表 5 所示，以方便比較。

環境參數	設定值
模擬玩家同時連線人數	100 人
模擬伺服器上傳頻寬	1024 KB
模擬玩家網路頻寬環境	下載 256 KB/Sec, 上傳 64KB/Sec
模擬玩家 IP 位址形態	皆設定為 Public IP

表 5：實驗共通設定

以下是各實驗相關參數：

- 實驗一：遊戲檔案分群，玩家等待時間

本實驗把遊戲檔案進行分群，且將啟動遊戲必要檔案切割出來，並且將之設置為高優先權。為方便比較，群組分為兩群，其中第一群假設為遊戲核心模組，優先權高；第二群假設為遊戲多媒體模組，優先權低。假設玩家下載完優先權高之遊戲核心模組後，即可進行遊戲。核心模組大小不變，多媒體模組則漸次加大，以模擬日漸龐大的多媒體檔案。實驗將分別量測記錄下載完核心模組與完成多媒體模組的時間。

其中遊戲檔案大小分為下列三種情況：

1. 共 16 MB，分兩群（8 MB + 8 MB）。
2. 共 58 MB，分兩群（8 MB + 50 MB）。
3. 共 108 MB，分兩群（8 MB + 100 MB）。

- 實驗二：檔案分群下載與傳統不分群下載

本實驗目的是比較檔案分群下載，與傳統不分群檔案下載，透過本實驗，將檔案分群後，並未增加過多玩家等待時間。本實驗總體模擬遊戲檔案大小固定，分為兩優先群組，分別為 8 MB 的麻將檔案群組與 8 MB 大老二檔案群組。並將玩家分為兩群，各 50 人，分別針對麻將群組與大老二群組進行下載。

- 實驗三：遊戲共用模組下載順序探討

本實驗透過群組優先順序掉換，分別量測核心模組下載時間，與全體下載完成時間。模擬遊戲全部檔案大小為 24 MB，等分為三個群組。分別為「核心共用模組(1)」、「麻將模組(2)」、「大老二模組(3)」。玩家分為兩群，各 50 人，分別下載麻將、大老二遊戲。其中模組下載優先順序分為下列三種模式：

模式 A：

麻將玩家：

「核心共用模組(1)」與「麻將模組(2)」優先下載，再下載「大老二模組(3)」。

大老二玩家：

「核心共用模組(1)」與「大老二模組(3)」優先下載，再下載「麻將模組(2)」。

模式 B：

麻將玩家：

依序下載「麻將模組(2)」、「核心共用模組(1)」與「大老二模組(3)」。

大老二玩家：

依序下載「大老二模組(3)」、「核心共用模組(1)」與「麻將模組(2)」。

模式 C：

麻將玩家：

依序下載「核心共用模組(1)」、「麻將模組(2)」與「大老二模組(3)」。

大老二玩家：

依序下載「核心共用模組(1)」、「大老二模組(3)」與「麻將模組(2)」。

5.3 實驗結果與分析

本節將展示所有實驗統計出來之數據，並分析其結果。

5.3.1 實驗一數據

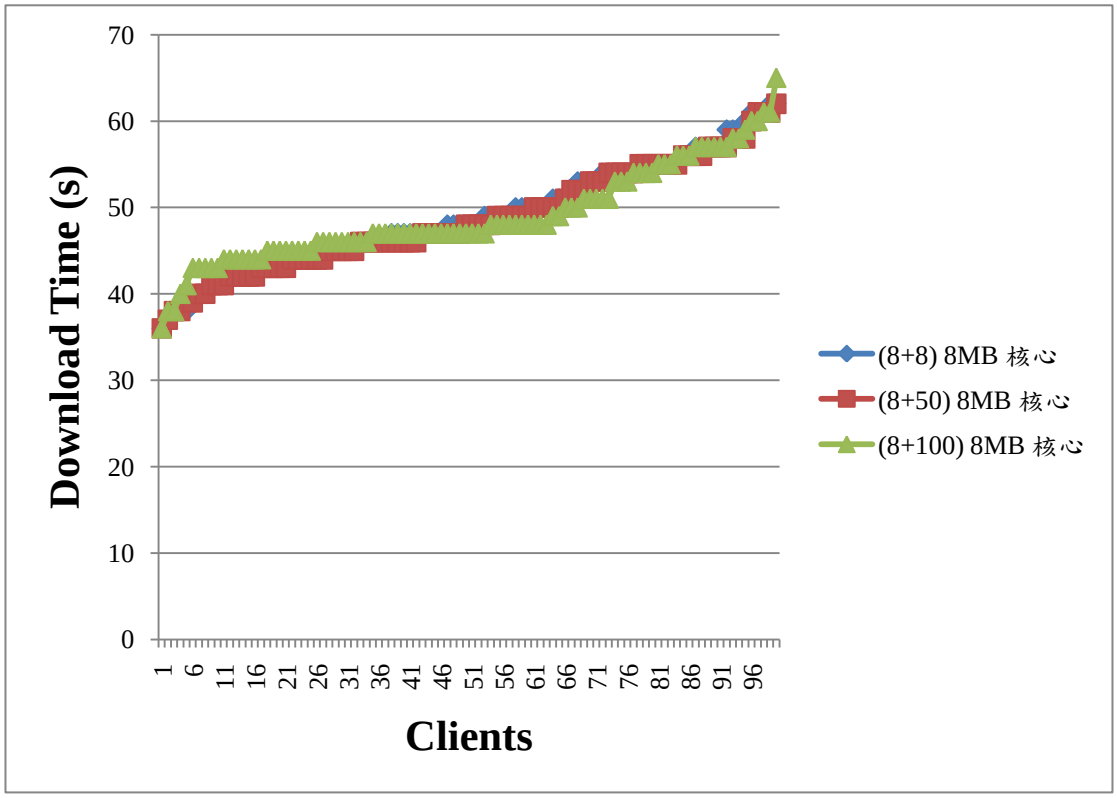

表 6：核心模組下載完成時間

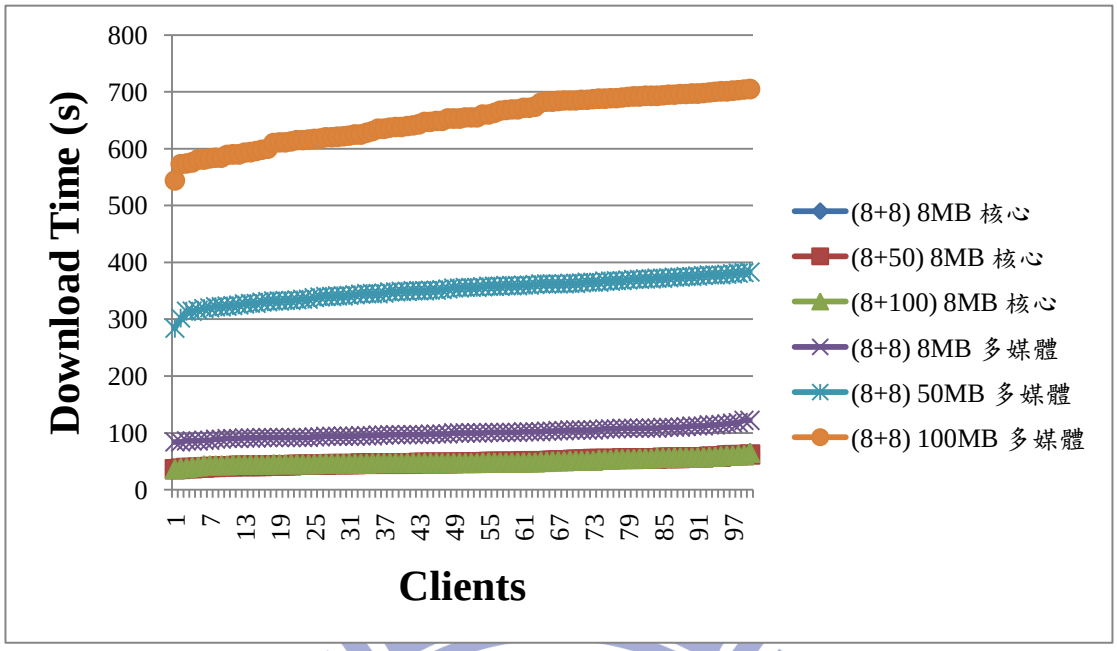

表 7：核心模組下載完成時間與整體下載完成時間

由以上兩表可看出，群組若是切割得當，不管餘下之群組大小有多龐大，皆可透過優先權設定，使遊戲核心模組先行下載完畢，玩家得以快速進入遊戲，減少等待時間。

5.3.2 實驗二數據

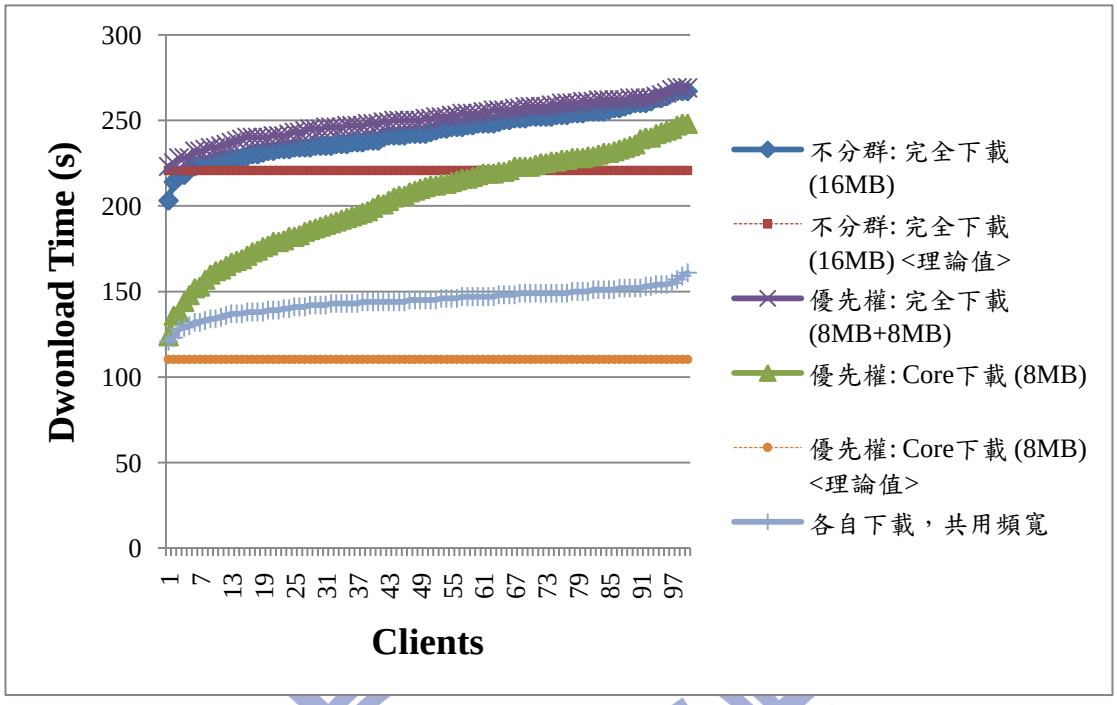

表 8：分群下載時間與不分群下載時間

由上表可以發現，分群後完整下載的時間，與只分一群完整下載的時間差不多，大致上會稍慢幾秒，但優先群組下載中核心部份花費的時間，尤其是在後段部份，效能不佳。有部份玩家核心下載完成時間，與總體檔案下載完成時間接近。原因是因為玩家分為二群，率先完成第一優先權檔案的玩家，會繼續向 Super Seeder 索取第二優先權的檔案，此時 Super Seeder 的頻寬就會分享部份頻寬給希望抓第二優先權的用戶端，因為如此，導致部分用戶，核心遊戲模組下載變長。

透過修改 Super Seeder，使其發現使用者要求的檔案並非高優先權之檔案，即暫時不發送片段給他，以節省 Super Seeder 頻寬。並

縮短其他用戶端下載第一優先權檔案之時間。

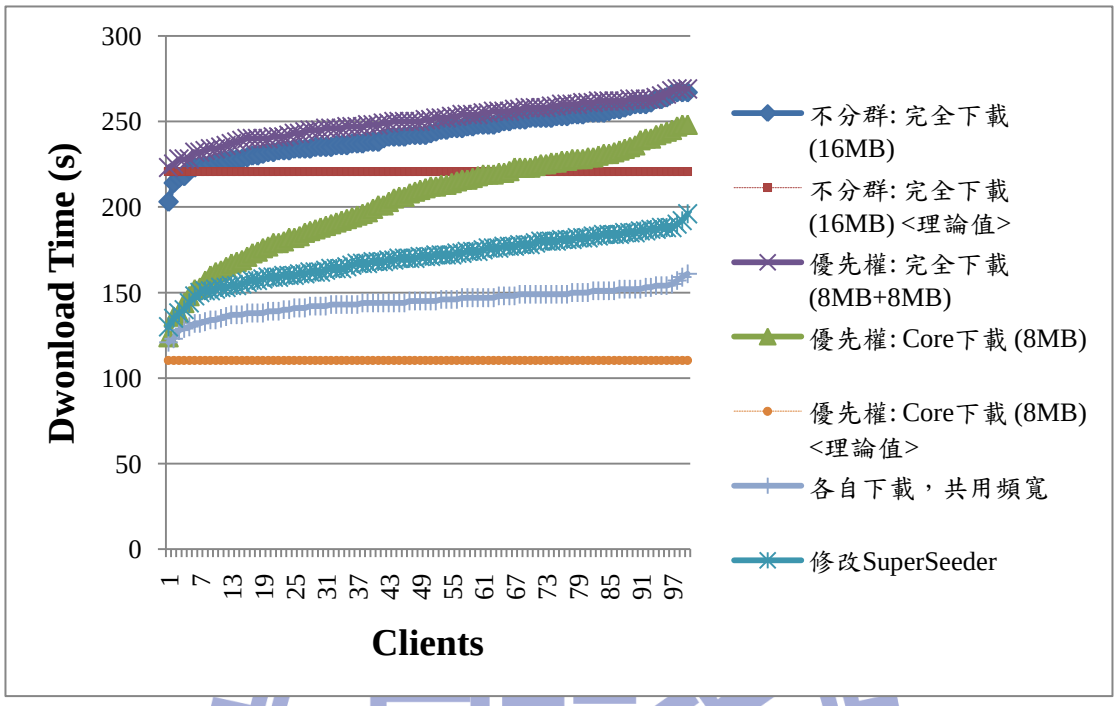

表 9：分群下載、不分群下載時間與修改 Super Seeder 上傳行為

5.3.3 實驗三數據

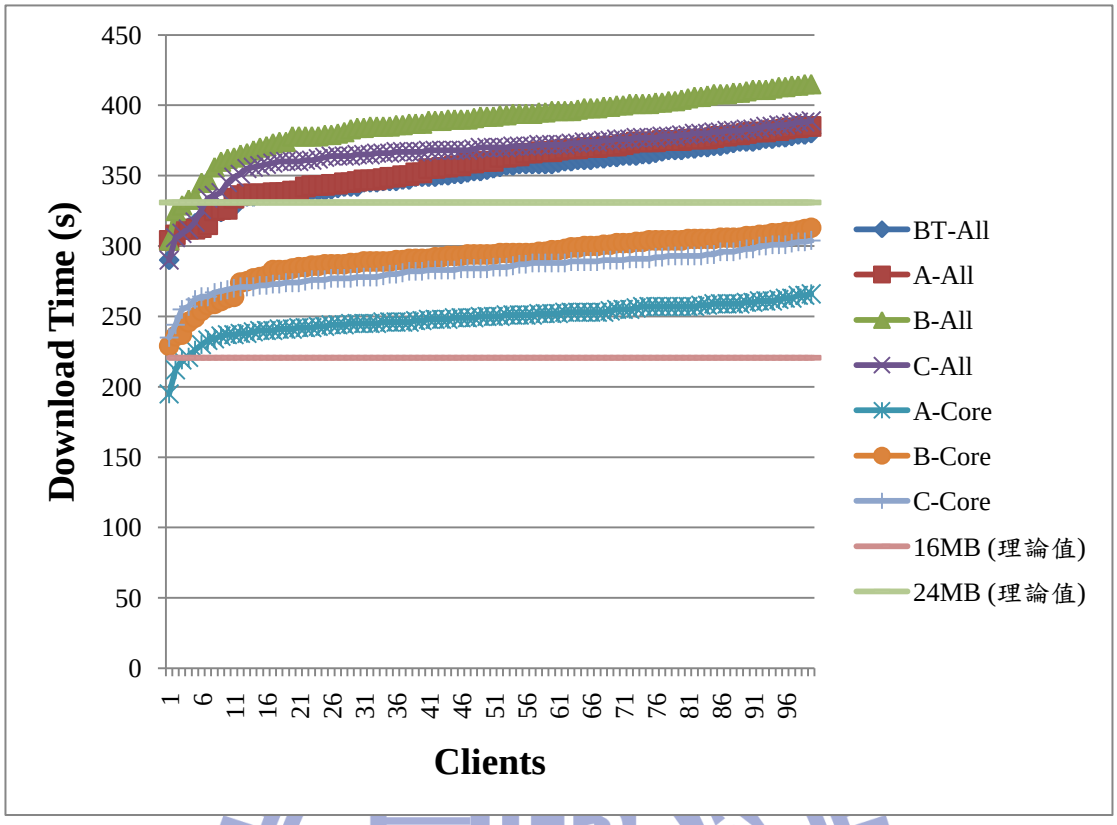

表 10：共享模組下載數據

由實驗三數據中可發現，分群下載中模式 A 的下載效率最好，接著是模式 C，最後是模式 B。

模式 A 將共用模組與高優先模組設定為等同優先權，如此的下載效率明顯高於其他兩種模式。模式 C 採用先下載共用模組的方式，效率略高於模式 B，模式 B 則優先下載自身所需要之模組，其次才下載共用模組，相較之下，模式 B 的下載效率最差。

第六章、結論與未來展望

6.1 結論

本論文提出了一套具有檔案分群與優先權機制的 P2P 檔案下載系統，尤其是針對線上遊戲公司，因為遊戲產品的模組化，使得優先分群下載可以發揮功效。本研究達成了以下目標：

- 透過優先分群，玩家可優先下載遊戲必要模組，加速進入遊戲時間。
- 透過分群，遊戲更新檔案發佈，也可不用如同傳統 BitTorrent 系統或 MDS 系統，需要再重新下載整包遊戲，而可以只下載更新檔案，達成快速進入遊戲之目的。
- 優化 Super Seeder 上傳對象選擇，優先選擇高優先權片段需求之客戶端傳遞，使得高優先群組的整體平均下載速度加快。

6.2 未來展望

對於我們實驗室先前所開發之 MDS 系統，與本論文所提出之檔

案發佈系統，還有許多部份是可以加強的：

- **開發真正跨平台之 Client：**

  目前本系統 Client 是採用 Cygwin 模擬 Posix API 的方式，移植到 Microsoft Windows 平台上使用，若是可以採用跨平台函式庫，或 Windows 原生 API，Client 的下載效能應可更加提高。

- **開發模擬環境，與效能分析工具：**

  開發過程中，經常需要針對所改善的程式進行效率量測，目前作法皆採用人工手動啟用大量用戶端，並分析其 Log，每次實驗皆花費不少寶貴時間，且人工手續複雜，一個步驟出錯，當次實驗的準確性即受到質疑，因此，有一個良好的模擬分析系統，會對開發 P2P 程式有很大的助益。

- **系統管理介面：**

  本系統希望是可實際應用於真實世界的企業之中，比如線上遊戲公司，隨著各個元件與參數設定越來越多，有一個集中管理介面，可以使得採用此套系統的企業，更佳容易上手。

參考資料

[1] Che-Yi Lin, “The Study of Massive Deployment System Based on P2P Technology - The Servers and System”, Master Thesis, NCTU, Taiwan, September 2008.

[2] Chou, Ting-Liang, “The Study of Massive Deployment System Based on P2P Technology - Clients and Performance Analysis”, Master Thesis, NCTU, Taiwan, September 2008.

[3] Bram Cohen, “Incentives Build Robustness in BitTorrent”, May 2003.

[4] Rakesh Kumar, Keith Ross, “Optimal Peer-Assisted File Distribution:Single and Multi-Class Problems”, Hot Topics in Web Systems and Technologies, 2006.

[5] Arnaud Legout, I.N.R.I.A., Guillaume Urvoy-Keller and Pietro Michiardi, Institut Eurecom Sophia Antipolis, France, Technical Report, “Understanding BitTorrent : An Experimental Perspective”, November 2005.

[6] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Fellber, A. Al Hanmra, and L. Garces-Erice, “Dessecting BitTorrent: Five Months in a Torrent’s Lifetime”, 2004.

[7] Microsoft Research, “Analyzing and Improving BitTorrent Performance”, 2005.

[8] Justin Frankel, Tom Pepper, Gnutella, Available: http://en.wikipedia.org/wiki/Gnutella , 2000 - 2009.

[9] eMule-Team, eMule, Available: http://www.emule-project.net , 2002 - 2009.

[10] Shawn Fanning, Napster, Inc., Available: http://www.napster.com , 1999 - 2001, 2004 - 2008.

[11] Sharman Networks, KaZaA, Available: http://en.wikipedia.org/wiki/Kazaa, 2001-2009.

[12] Skype Limited, Skype, Available: http://en.wikipedia.org/wiki/Skype, 2003-2009.

[13] BitTorrent Specification, Available: http://wiki.theory.org/BitTorrentSpecification, February 2008.

[14] Yi-Chan Shan, “Performance Analysis of P2P File Distribution”, private communication, 2008.

[15] 群想科技, 客戶端 NAT 與頻寬分析, 內部文件, 2008 年 8 月.