

在 BT 同儕通訊上利用參與者訊息引導對外流量之區域化節約

白淳元、楊宏昌、曾黎明、游象甫

國立中央大學資訊工程研究所

{cvb, cyht, yu}@dslab.csie.ncu.edu.tw, tsenglm@csie.ncu.edu.tw

摘要

BitTorrent 是現在同儕網路中使用率較高的分散式檔案傳輸軟體，在 BitTorrent 中使用者程式無法得知所有的下載者，且彼此依對方上傳速度頻寬決定是否同意允許下載，導致使用者程式可能不知道區域網路中有其他同儕程式擁有相同的檔案，使得通過路由器之間的傳送造成了重複的流量，降低了頻寬的使用效率。本文提出一個機制與訊息引導算法，藉由通透的代理伺服器來攔截並修改使用者程式至網站伺服器上所下載 BitTorrent 的 metainfo 檔案，將使用者程式與原 Tracker 的連線導向至我們在區域網路中設置的內部 Tracker Agent，再由我們所設定的 BitTorrent 客戶端代理程式 (Peer agent) 引導不同的使用者程式抓取不同的檔案片斷，並且透過我們所設定的客戶端代理程式與訊息引導演算法來增加區域化的流量，以減少頻寬的浪費，使用者也能得到快速下載的服務品質。

關鍵字：同儕網路、BitTorrent、區域化

1. 緒論

現在利用同儕網路技術來進行檔案分享也已經受到廣泛的使用，而在此方面較為出名的軟體有 Napster [1]、Gnutella [2]。除了上述兩種，BitTorrent [3] 是現在使用率較高的軟體，BitTorrent 檔案傳輸系統是由 Bram Cohen 在 2001 年發展出來的，是屬於第三代的同儕網路技術。它採取其特殊的獎懲機制 (tit-for-tat) 來鼓勵使用者能夠貢獻上傳頻寬，而根據近期 CNN 的報導 [4]，BitTorrent 的流量佔了所有網路的流量中高達 20% 以上，其他報告也顯示出同儕網路的流量 (包含 BitTorrent) 在 ISP 中也佔了超過 60% [5]。

根據傳統 BitTorrent 的演算法，每個使用者都是從 Tracker 得到的使用者名單中選取一部份來做連結，而 Tracker 給予的名單是隨機選取，在加上同儕網路點對點直接連線的特性，每個使用者可能在下載同一個資料，但由於隨機選取的關係，使用者可能不知道區域網路中有使用者擁有相同的檔案，再加上內部使用者限速的關係，且受 tit-for-tat 演算法的影響，無法展現區域網路速度快的優勢，而導致許多跨網域的重複流量，不但造成使用者等待時間較長，供檔端 (server Peer) 負荷也較大。

而現今較常使用的方法，是利用防火牆來限制對外出去的頻寬，或利用 L7-content filter [6] 來過濾並阻擋 BitTorrent 的封包，但由於 BitTorrent 中

tit-for-tat 的機制可能會減少使用者間彼此的交流，因而增加檔案下載的時間，這是使用者所不樂見。

我們在本文提出一個機制，藉由通透的代理伺服器來攔截並修改使用者至網站伺服器上所下載 BitTorrent 的 metainfo 檔案，將使用者與原 Tracker 的連線導向至我們在區域網路中設置的內部 Tracker，再由我們所設定的 BitTorrent 客戶端引導不同的使用者抓取不同的檔案片斷，並且透過我們所設定的客戶端來增加區域化的流量，進而減少外部重複的流量。

本論文分為五個部份，在第二節列出相關研究。第三節提出系統架構。第四節為實驗，最後是結論及未來研究方向。

2. 相關研究

2.1 BitTorrent 原理

BitTorrent (BT) 是一套目前相當流行 P2P 的檔案分享系統，其開發者是 Bram Cohen。它讓每個使用者程式都貢獻自己的上傳頻寬，來讓所有的使用者程式來下載檔案，來使得下載更有效率。

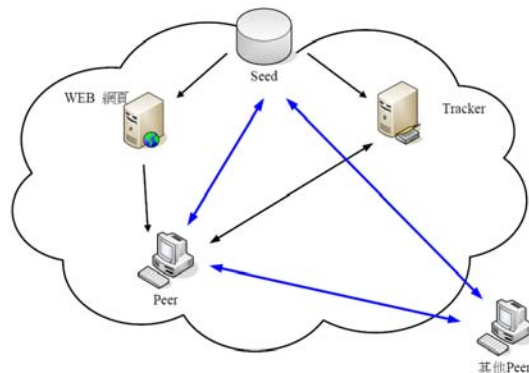


圖 1 BitTorrent 運作示意圖

如圖 1 所示，當使用者從網路上下載到所需的 .torrent，接著啟動 BitTorrent 客戶端來進行下載。用戶端軟體在讀取 .torrent 檔之後，首先確認該檔案之完整性 (integrity)，通過之後與指定的 Tracker 連接，Tracker 會告知下載者其他正在下載此檔案的 Peer 清單，但 Tracker 傳回的清單是從所有的清單中隨機選取的。

由於使用者程式是從主要其他的 Peer 來下載檔案的片斷。當你連接的 Peer 中有你需要的區塊

時，你會發出一個 interest 的訊息表示有意下載，此 Peer 會回使用者程式阻塞 (choking) 或疏通 (unchoking) 的訊息，若為疏通的訊息才可從 Peer 下載區塊。因此一個好的阻塞演算法要提高下載的效率。讓上傳速率高者得到較高的下載的速率，並適當的處罰只下載而不做上傳的 Peer，這也稱為 "tit-for-tat" 演算法。

每個 Peer 會跟固定數量的其他 Peer 保持疏通的狀態(預設為四個)，在 BitTorrent 中的作法是選出一個 Peer 提供給使用者程式很高的下載速率將優先疏通，在 BitTorrent 中是每10秒作一次計算。但此方法可能會將新加入的 Peer 排除在外(因為沒有擁有任何區塊)，為了解決此問題，BitTorrent 使用一種叫 "Optimistic Unchoking" 的方法，每個 Peer 都會保持一個 "Optimistic Unchoking" 的連線，不管對方上傳速度為何，每30秒輪流其他的 Peer 為 "Optimistic Unchoking"，藉由此方法可解決上述問題而且可以找到其他能力更好的 Peer。

當使用者程式對其中一個 Peer 的區塊有興趣，對方也發給疏通的訊息，使用者程式就開始向此 Peer 要求區塊，要選擇哪一個區塊比較有效率，這是由片斷選擇 (Piece selection) 演算法來決定的，其重要的原則如下：

- (1) 完整優先 (Strict Priority)：在 BitTorrent 設計中一旦有過大的請求會被拒絕，使用者程式則會請求了此片斷的子片斷 (Sub-Piece)，那麼該片斷剩下的子片斷優先被請求。盡可能快點獲得一個完整的片斷。
- (2) 稀有優先 (Rarest First)：對一個下載者來說，在選擇下一個被下載的片斷時，通常選擇的是它連接的 Peers 中所擁有的最少的那個片斷。但由於每個使用者程式連接的 Peers 都會有所不同，所以只是區域稀有優先 (Local Rarest First) 而不是全域稀有優先 (Global Rarest First)。
- (3) 隨機第一個片斷 (Random First Piece)：在下載剛開始的時候，下載者沒有任何片斷可供上傳，所以，需要儘快的獲取一個完整的片斷。第一個片斷是隨機選擇的，直到第一個片斷下載完成，才切換到“最少優先”的策略。
- (4) 最後階段模式 (Endgame Mode)：在最後階段模式，下載者會向其他的 Peer 發送所有未完成的子片斷請求，當有某個子片斷開始下載時，就會發送取消的訊息給其他 Peer，這樣檔案在最後階段可以比較快速完成

2.2 現有方法之研究

在同儕網路中減少跨網域流量最常見的方法就是快取 (caching)，Cachologic [5] 發表這樣的伺服器，它主要的方法是對同儕網路服務請求進行檢測，並識別其的關鍵字。然後對比伺服器資料庫，檢測該請求下載的檔是否已經

存在於伺服器中。如果該檔已經存在於伺服器，則立即從伺服器提供下載；如果該檔在伺服器中沒有找到，伺服器並不中斷服務，同時保存一個備份在伺服器以備同樣的服務請求使用。

除了快取的方法，T. Karagiannis [8] 提出了一種 "Gateway Peers" 的方式，所謂的 Gateway Peer 是一個在區域網路中的 Peer，但是唯一可對外溝通的 Peer，這樣每個區域網路中只要保留一份檔案的備份，然後在分享給區域網路中的其他同儕程式

在 Transparent and Gated cache [9] 中就提出一個類似 Gateway Peer 的架構，主要是針對 BitTorrent 而提出的方法，其架構如圖 2 所示

當第一位下載文件的使用者程式產生的系統運作。區域網路內的使用者在網頁瀏覽器上點擊有興趣的物件，伺服器會將物件的 metainfo 檔 (.torrent) 回傳給使用者，但使用者所取得之 metainfo 檔實為修改過之版本。在 Tracker 代理者部分當 Torrent 代理者在收到有使用者下載新的 .torrent 檔的 HTTP 請求時便已通知 Tracker 代理者，Tracker Agent 會取得原始 metainfo 所記載之 Trackers，並於自身 Tracker 發佈該文件，並向原 Tracker 詢問下載者清單。在當第一位使用者程式取得修改過之 metainfo 檔案，便會與 Tracker 代理者 (TA) 連線，而 Tracker 代理者已經發佈了該 metainfo 檔案。便會回覆使用者程式一張目前的下載者清單，此清單是目前 Tracker 代理者所登記過的區域網路使用者。

外部客戶端代理程式在被 Torrent Agent 驅動之後，持有原來的 metainfo 檔案，索取相同的來源檔案。並將資料寫入至一與區域網路中的 TA Client 共享的磁碟區中，由內部的 TA Client 讀取分享的檔案並發佈於內部 Tracker Agent，此時區域網路中只有 TA Client 有文件本身資料，內部的使用者程式便會嘗試連接上 TA Client，並在連接之後發出索取文件片斷 (piece) 的請求。

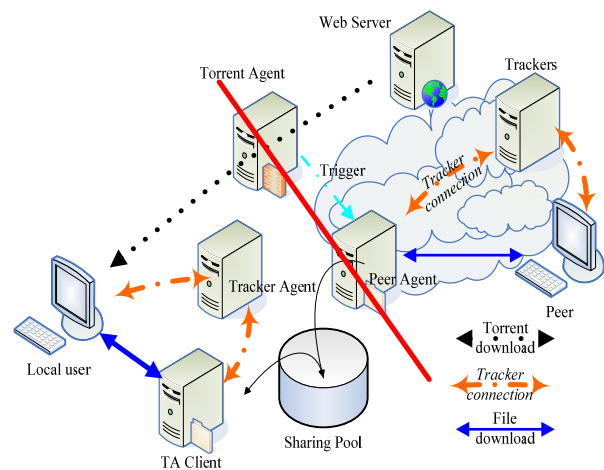


圖 2 Transparent and Gated cache 系統架構圖 [9]

3. 系統設計

在本節的部分主要分析我們系統的目的，及需求概述。

3.1 方法分析

在很多相關研究報告中都有分析 BitTorrent 在網路上的流量 [10][11][12]，這些研究將 BitTorrent 網路的流量主要分為三個階段：flash crowd，steady state 和 winding down [10]，且 Peer 的加入會隨著時間的增加而呈指數的減少 [11]。flash crowd 主要是發生在檔案第一次散佈的時候，會隨著檔案大小的不同而持續不同的時間，在這三個階段中 flash crowd 會產生最多重複的跨網域流量，因此我們主要就是減少在這階段中所產生重複的流量。

重複的流量所指的是在區域網路中不同使用者，向區域網路外的 Peer 要求同一個片斷，我們的目的就是在 flash crowd 的階段中讓區域網路中不同使用者分別去要求不同的片斷，接下來由於每個使用者擁有的片斷不同，就可以彼此分享自己所擁有的資源，而達到流量區域化的目的。

在 BitTorrent 中片斷選擇並不是胡亂選取的，有一個非常重要的特性就是稀有優先，第二章有介紹過所謂稀有優先是選擇的是它連接的 Peers 中所擁有的最少的那個片斷，根據這個特性我們設計出一個架構。

我們所採用的方法是利用 Peer 之間會互相交換訊息，來得知彼此所擁有的片斷，我們可以偽造互通的訊息，讓 Peer 誤認為某部分片斷的擁有者很多，根據 BitTorrent 中片斷選擇演算法，Peer 會優先要求偽造訊息中所沒有的片斷。當 BitTorrent 客戶端彼此要互相連接時，先有一段互相握手的協議，接著告知對方自己所擁有的片斷，主要靠著互相交換 bitfield 的消息，bitfield 是將所擁有的部份，用類似點陣圖的方式來表示，若以擁有的話為 1，否之為 0，BitTorrent 客戶端在收到這由 1、0 所組成的串列，便可得知對方擁有一些檔案片斷。

3.2 系統環境

系統環境如圖 3，我們主要設置 Torrent Agent 於網頁伺服器與使用者瀏覽器之間，作為攔截區域網路使用者下載 metainfo 檔案。當使用者啟動其 BitTorrent 用戶端軟體之後，Tracker Agent 則負責區域網路中使用者的 Tracker 通訊，並負責給予客戶端代理程式的位址及部分的外部使用者的清單。客戶端代理程式負責給予不同 Peer 不同的偽造的檔案片斷擁有訊息，並且負責處理 Peers 對它所發出的請求。

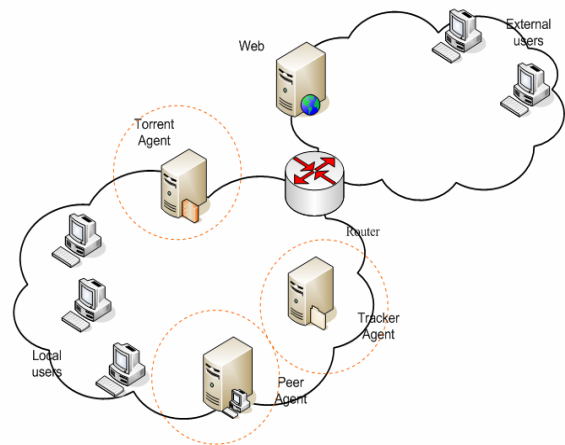


圖 3 系統環境之配置圖

3.3 系統說明

系統運作流程的第一階段如圖 4。區域網路內的使用者首先在網頁瀏覽器上點擊有興趣的物件，伺服器會將物件的 metainfo 檔 (.torrent) 回傳給使用者，但使用者所取得之 metainfo 檔實為修改過之版本。在 Tracker 代理者部分當 Torrent 代理者在收到有使用者下載新的 .torrent 檔的 HTTP 請求時便已通知 Tracker 代理者，Tracker Agent 會取得原始 metainfo 所記載之 Trackers，並於自身 Tracker 發佈該文件，並向原登記 Tracker 詢問下載者清單。

在當第一位使用者取得修改過之 metainfo 檔案，便會與 Tracker 代理者(TA)連線，而 Tracker 代理者已經發佈了該 metainfo 檔案。便會回覆使用者一張目前的下載者清單，此清單是一部分我們所設定的外部網路的 Peers 清單以及客戶端代理程式的位址，使用者便會根據此清單去跟客戶端代理程式做連接溝通。

如圖 5 所示，當第一個使用者程式跟客戶端代理程式做連接溝通時，客戶端代理程式會告知使用者程式有一群 Peers 擁有某一部份的片斷偽造消息，接著第二個使用者程式來跟客戶端代理程式做連接溝通，客戶端代理程式也會告知使用者程式有另一群 Peers 擁有另一部份的片斷偽造消息，接著 Peer 會根據從 Tracker agent 中所得到的外部清單，來跟外部的 Peers 做溝通，接下來會開始要求檔案片斷。

如圖 6 所示，由於客戶端代理程式會告知使用者程式有一些 Peers 擁有某一部份的片斷偽造消息，根據 BitTorrent 片斷選擇演算法，BitTorrent 客戶端會經過計算，判斷出客戶端代理程式所沒擁有的部份，是在使用者程式連接中的 Peers 中屬於比較稀少的部份，接著 BitTorrent 客戶端便會跟 Tracker agent 所給予的外部網路的 Peers 清單優先請求客戶端代理程式偽造消息中所沒有的檔案片斷，由於客戶端代理程式給予每個使用者程式偽造的片斷擁有消息都有所不同，每個使用者程式一

開始會優先要求不同的檔案片斷,但合起來是一份完整的檔案。

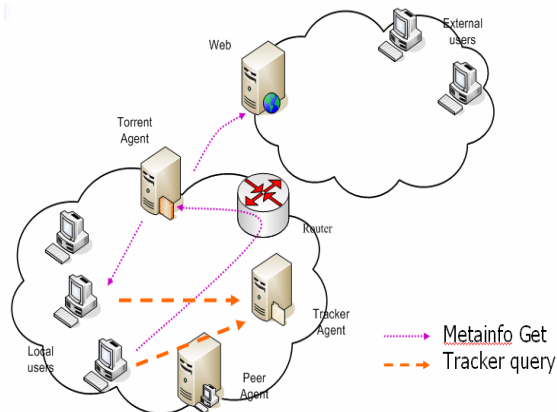


圖 4 系統初始時之示意圖

有擁有這部份的檔案片斷,因此客戶端代理程式會向內部的其他已經抓取這部分檔案片斷的同儕程式來發出請求,等到客戶端代理程式從內部其他同儕程式收到這部份的檔案片斷,客戶端代理程式會將檔案片斷存在自己的緩衝存儲器(Buffer)中,並且以很快的速度來回應使用者程式的請求,來達到內部互相交流的目的。

如圖 7,當使用者程式完成客戶端代理程式所設定抓取的檔案片斷,客戶端代理程式不會就此消失,因為客戶端代理程式以很高下載速度來回應使用者程式之前所要求的檔案片斷,經過 BitTorrent 的壅塞演算法計算會優先將 Peer agents 設為疏通,在其他的使用者也是同樣情況,讓客戶端代理程式可以較高的優先權限來互相抓取檔案片斷來回應給同儕程式的請求,使用者程式也會因為客戶端代理程式而增加了在內部互相交流檔案的機會,自然降低了向外抓取檔案片斷的機會,而使得流量區域化。

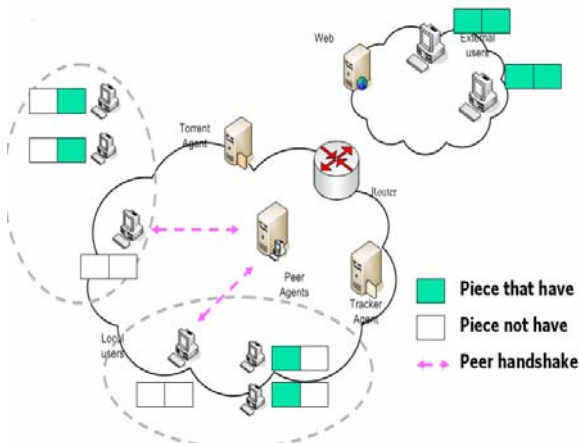


圖 5 使用者程式向客戶端代理程式溝通

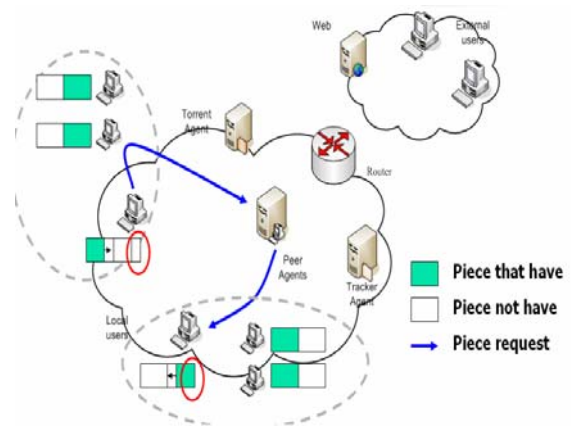


圖 7 客戶端代理程式處理使用者程式要求

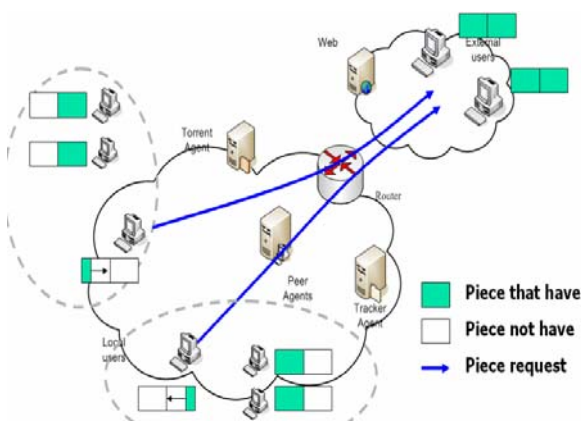


圖 6 使用者程式 向外部同儕程式要求檔案

當使用者程式向外部網路的 Peers 請求檔案片斷,並已經抓取設定某部份的檔案片斷,同時 BitTorrent 客戶端也會向客戶端代理程式發出偽造消息中的檔案片斷的請求,但客戶端代理程式並沒

3.4 訊息引導

由於客戶端代理程式會根據不同的使用者程式給予不同偽造的消息,來引導使用者程式優先抓取某部份的檔案片斷,因此訊息引導是系統相當關鍵的部分,我們所設定訊息引導演算法,首先讓第一個使用者程式引導抓整個檔案的二分之一的片斷,接著陸續加入的使用者程式則是引導抓取尚未引導部分的前二分之一,主要讓越早加入的使用者設定抓取較多的檔案片斷,越晚加入的則設定較少,可以使每個使用者完成系統所設定的片斷時間差距縮小,下列為訊息引導的演算法。

但檔案會有熱門程度的區別,當檔案越熱門內部的使用者會有較多人參與客戶端代理程式所回應的次數也會越多,因此訊息引導的演算法也要做些調整,如可將未引導檔案的二分之一調整成三分之一或四分之一,但是要如何判斷檔案的熱門程度,我們可由以下方法來做判斷:

```

N_file: number of piece to guide
File.numberpieces: number of total piece
File.guided[]: piece that have guided
File.have[]: piece information send Peer to guide
if got newtorrent
    for i=1 to File.numberpieces
        File.guided[i] = 0
        N_file=ceil(numberpiece / 2)
else if got handshake of R
    count = 0
    for i=1 to File.numberpieces
        File.have[i] = 1
    for i=1 to File.numberpieces
        { if File.guided[i] == 0
            { if count == N_file
                { if N_file > 1
                    { N_file=ceil(N_file / 2) }
                    break
                }
            }
            File.guided[i] = 1
            File.have[i] = 0
            count += 1
        }
    }
return bitfield(File.have)

```

```

//variable declaration

// initial

// No piece guided
// set number of piece to guide
//got handshake from Peer

//pseudo Peers have all piece ,initial

// decide which Peer to get which piece

// set number of piece to guide

//piece have guided
//piece to guide set pseudo Peer not have

```

1. 由客戶端代理程式的溝通次數來做判斷，但這種方法可能沒辦法做到及時的判斷來更新，而會有所誤差。
2. 當 Tracker agent 取得外部清單的時候，便可知外部使用者的個數，再以此為依據判斷檔案熱門的程度，再決定該分給多少人抓。若內部使用者超過這個數，就不給予外部的清單。

大。

4. 實驗

本章節針對我們設計的方式，展示實驗時運作情形及其數據，並做分析討論。
 (內部使用者下載的總頻寬÷檔案的總大小) - 100% = 重複頻寬比例

4.2 實驗一

實驗一的主要目的，首先是測試每個使用者是否會根據客戶端代理程式偽造消息，來優先抓取不同的區塊的。

如圖 8 共有三位使用者其中一個為擁有完整檔案的 Seed 而客戶端代理程式 E 及客戶端代理程式 D 則分別設定為擁有前半段及擁有後半段的訊息，Peer C 只知道 Seed A 及客戶端代理程式 E 的訊息，Peer B 只知道 Seed A 及客戶端代理程式 D 的訊息，檔案的總大小為 24330372 Kbytes。

由圖 9, 10 可知使用者程式確實會被客戶端代理程式所引導，但不是客戶端代理程式所設定要抓取的片斷的比例會有所不同，這是因為 BitTorrent 一開始的片斷選擇是隨機存取的，但這比例不會太

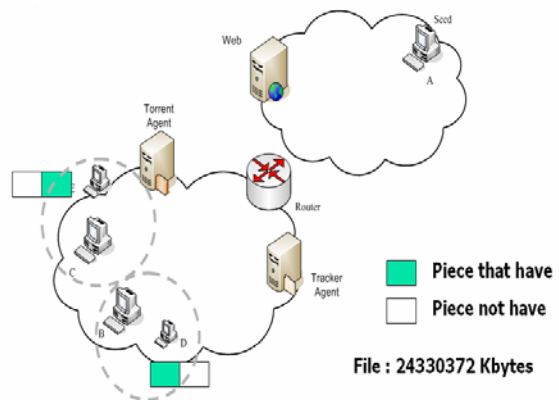


圖 8 實驗一環境示意圖

| 名稱 | 大小 | 進度 | 下載 | 上傳 | 需時 | 種子用戶 |
|------|----------|-------|--------|-------|----------|------|
| BTEX | 23.20 MB | 52.1% | 51kB/s | 0kB/s | 00:03:44 | 12 |

| 內容簡介 | 完成度 | 檔案名 | 大小 | 開啟方式 |
|------|-----|-----------------|----------|------|
| 任務傳真 | 一般 | 97.3% 1.rar.bcl | 11.60 MB | 讀寫 |
| 任務傳真 | 一般 | 0.6% 2.rar.bcl | 11.60 MB | 讀寫 |

圖 9 Peer B 抓取檔案的情形

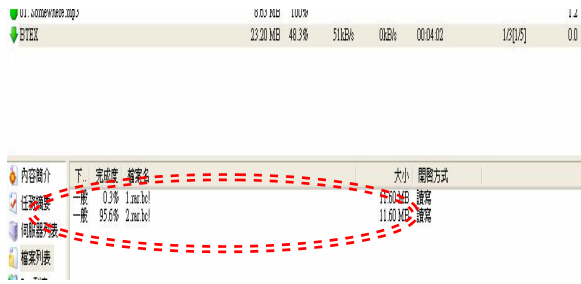


圖 10 Peer C 抓取檔案的情形

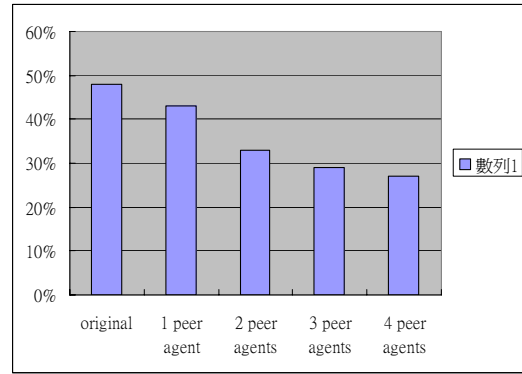


圖 12 實驗 2 的數據

4.2 實驗二

實驗二的目的，是評估在同樣情況下，內部的使用者程式是否會根據客戶端代理程式的個數，而產生變化。

實驗的拓撲圖如圖 11 所示，共有五個參加者，A 是一個擁有完整檔案的 Seed，B.C 是加入一陣子抓取某部分片斷的 Peer，D.E 則是屬於內部剛加入的使用者，而檔案的總大小為 24330372 Kbytes。

首先按照 BitTorrent 原本系統來測試重複的頻寬佔多少比例。再按照本論文提出的架構，其中的偽造訊息則分別設定前半段及擁有後半段的訊息，然後加入客戶端代理程式測試重複頻寬的比例。再逐一增加客戶端代理程式的個數。

圖 12 是實驗 2 所得到的數據，由圖可知隨著客戶端代理程式的增加，重複頻寬的比例的確有所下降，但下降的比例也漸漸減少，因為內部使用者程式上傳的頻寬是固定的，而客戶端代理程式只能夠幫助使用者程式的頻寬盡量傳給其他內部的同儕程式。

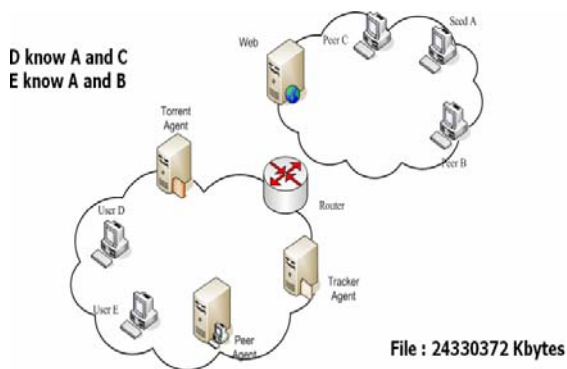


圖 11 實驗二的環境示意圖

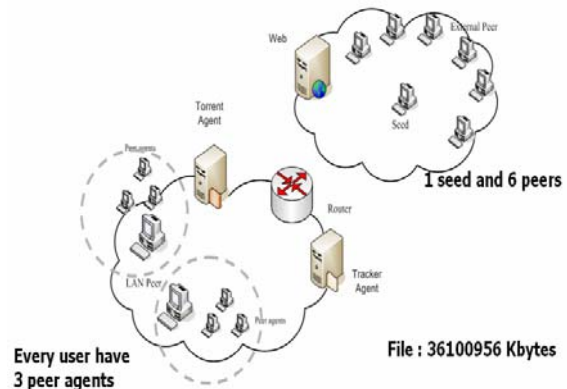


圖 13 實驗三之環境示意圖

4.3 實驗三

實驗三主要目的是測試在每個使用者程式連接相同的客戶端代理程式的情況下，內部使用者程式的增加所產生的變化。

實驗三的示意如圖 13 所示，外部網路共有七個參與者，其中有一個擁有完整檔案的 Seed，其餘的都是普通的 Peers，而區域網路的使用者會隨著實驗而變化，Tracker 所給予外部使用者的數量也會有所不同，但擁有客戶端代理程式的數目是固定的。檔案的總大小為 36100956 Kbytes。

首先在區域網路內加入兩個使用者，按照 BitTorrent 原本系統來測試重複的頻寬佔多少比例。按照本論文所提出架構，並設定使用者連接分別連接 3 個客戶端代理程式，再測試重複頻寬的比例以及檔案完成的時間。逐一增加內部使用者的個數，重複上述步驟

由圖 14,15 可得知隨著內部使用者的增加，雖然在我們的架構下重複的頻寬雖然還是存在，但已明顯減少許多，而且隨著使用者的增加減少的比例越多，這是因為原本內部使用者程式對外部的上傳頻寬，都被客戶端代理程式導向內部的同儕程式了，檔案完成的時間也相對減少。

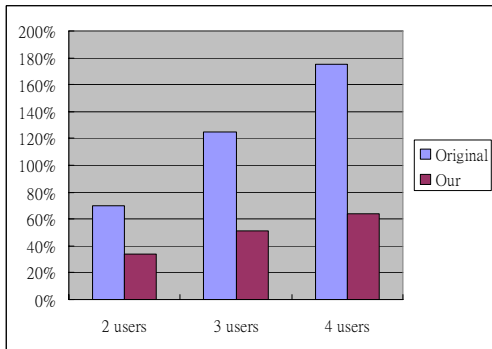


圖 14 實驗三之數據圖之一

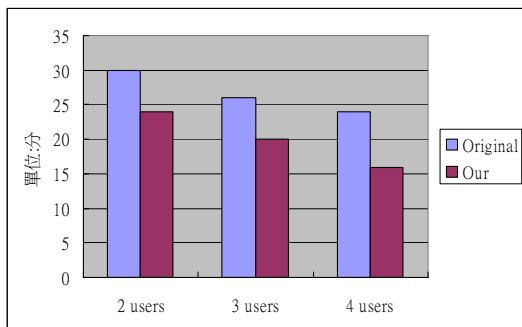


圖 15 實驗三之數據圖之二

5. 結論

本文針對 BitTorrent 提出一個機制與訊息引導演算法，藉由通透的代理伺服器來攔截並修改使用者至網站伺服器上所下載 BitTorrent 的 metainfo 檔案，將使用者程式與原 Tracker 的連線導向至我們在區域網路中設置的內部 Tracker，再由我們所設定的 BitTorrent 客戶端代理程式來引導與影響不同的使用者程式抓取不同的檔案片斷，並且透過我們所設定的客戶端來增加區域化的流量。

而我們所提出作法擁有的優點主要是：

- 現存使用者不必有明顯的特別設定
- 有效增加內部互傳的流量
- 降低路由器對外傳輸之負擔
- 不需額外龐大的儲存空間

未來可以將本系統放在比較大型的網路環境做測試，並深入探討訊息導引演算法之分配模式，量測其效能，並且探討如何配置本架構以達到最好的效能。

致謝

The authors would like to thank the National Science Council of the Republic of China for fanatically supporting this research under Contract No. NSC-94-2213-E-008-026.

參考文獻

- [1] Napster, <http://www.napster.com/>
- [2] Gnutella 0.6 protocol draft, <http://rfc-gnutella.sourceforge.net/>
- [3] B. Cohen, "Incentives build robustness in bittorrent," 2003. [Online]. Available: <http://citeseer.nj.nec.com/cohen03incentives.htm>
- [4] EContentMag.com, "Chasing the user: The revenue streams of 2006," 2005. [Online]. Available: <http://www.econtentmag.com/Articles/ArticleReader.aspx?ArticleID=14532n&ContextSubtypeID=8>
- [5] CacheLogic, "Cachelogic - advanced solutions for Peer-to-Peer networks." [Online]. Available: <http://www.cachelogic.com>
- [6] L7-filter, <http://l7-filter.sourceforge.net/protocols>
- [7] S. Patro and Y. C. Hu, "Transparent Query Caching in Peer-to-Peer Overlay Networks," in *Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS), 2003*
- [8] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should internet service providers fear Peer-assisted content distribution," in *Proceedings of the Internet Measurement Conference 2005, 2005*.
- [9] 陳宜凡, 白淳元, 游象甫, 曾黎明 "A Transparent and Gated cache for BT-P2P communication," TANET 2005
- [10] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Hamra, and L. Garcés-Erice, "Dissecting bittorrent: Five months in a torrent's lifetime," 2004
- [11] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis and modeling of bittorrent-like systems," in *Proceedings of the Internet Measurement Conference, 2005*
- [12] Dongyu Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks," in *Proceedings of SIGCOMM, 2004*.