

利用分工與協助來降低類 BitTorrent 系統中的跨 ISP 流量

Chun-Hsien Lu (呂俊賢) and Wei-Ting Lee (李威廷)
 Dept. of Computer Science and Information Engineering
 Fu Jen Catholic University
 輔仁大學資訊工程系
jonlu@csie.fju.edu.tw

Abstract In early stages of the Internet, most applications adopt the client-server model to obtain information. However, as the number of users increases, the Peer-to-Peer (P2P) architecture has replaced the traditional client-server model to become the mainstream for file sharing. P2P technology allows each peer to act as both a client and a server at the same time that shares files with other peers. Although P2P technology has the advantage of reducing the file download time and the server load, the mismatch between the overlay topology and the underlying network has led to degraded transmission performance and large costs to the Internet Service Providers (ISPs). We proposed a Reducing Cross-ISP Traffic via Coordination and Assistance (RCTCA) algorithm to increase traffic localization via collaboration between the ISPs and the P2P system. Specifically, peers in the same ISP will be organized as a group, and collaborate in a structured way. This way, a group can maximally utilize their local bandwidth, resulting in lower cross-ISP traffic without significantly increasing the average file download time. The simulation results show that, compared to the BitTorrent-like strategy, our RCTCA algorithm could reduce the cross-ISP traffic by 58% to 98% and the file download time by 34% to 75%.

Keywords: Peer-to-peer Network, Reducing Cross-ISP Traffic, File Download Time

摘要 一點對點系統由於其節點之間的虛擬網路拓墣與實際架構並不相稱，不僅會影響傳輸上的表現，更產生許多跨 ISP 的流量，造成 ISP 的成本上升。我們提出一個 Reducing Cross-ISP Traffic via Coordination and Assistance 的方法，透過 ISP 與點對點系統合作，提高系統中本地傳輸的流量比例。我們依據節點所在的 ISP 將節點分組，讓同組節點相互合作，每個節點先各自向組外抓取不同的片段，再互相傳輸，讓本地傳輸的流量比例最大化，以減少跨 ISP 網路流量卻不會大幅增加平均檔案下載完成時間。實驗結果顯示我們的方法與類 BitTorrent 方法相較，跨 ISP 傳輸流量可降低 58%，而平均檔案下載完成時間也能縮短 34%。

關鍵詞：點對點網路，降低跨 ISP 流量，檔案下載時間

1. 導論

近年來，隨著網路傳輸的發達，資訊傳播量大幅增加，分享檔案已成為現今網路上最常見的行為之一，且被分享的檔案數量仍迅速成長中[1]。傳統的主從式架構已經漸漸不敷使用，點對點(Peer-to-Peer, P2P)架構因應而生。這個技術把伺服器端的負載分散給底下所有的節點(Peer)，每個節點都可能扮演著伺服器和用戶端的角色，同時執行上傳與下載的功能，提供本身資源來服務其他節點的需求。P2P 架構能夠提供多種網路服務，如文件共享[2][3]，或是視訊串流和 IP 語音通話[4-8]等。比起傳統的主從式架構，P2P 架構充分利用每個節點的網路頻寬，因而擁有較大的網路負載能力、較小的伺服器負載及較快的檔案傳輸等優點。

BitTorrent (BT)[2]為一種非常熱門的 P2P 協定，它將檔案分割成許多較小的片段(piece)，作為分享的單位，當一個節點下載完一個片段後，即可開始分享給其他需要該片段的節點。這樣不僅可以讓節點儘早開始分享，同時也可以增加整個 P2P 網路中每個節點提供上傳頻寬的時間，改善節點下載完可能立即離線的問題。由於 BT 的鄰居選擇機制是透過隨機挑選，所以在網路拓墣中節點的連線容易出現跨越不同網際網路服務供應商(Internet Service Provider, ISP)的狀況，進行資料交換的時候容易因為實際有效傳輸頻寬較低，造成網路壅塞，導致下載時間增加。同時，近期研究顯示 P2P 架構產生的大量跨 ISP 流量已經造成 ISP 的營運成本顯著增加，其流量約當於整個網際網路跨 ISP 流量的 60% [9] [10]。

本論文提出一個方法，以減少這種代價高昂的跨 ISP 流量，卻不犧牲系統效能。本方法將節點依據其所處在的 ISP 分成數個群組，當節點在選擇欲下載的片段時，將挑選整個群組中尚未有任何節點擁有的片段來優先下載。而群組中已擁有的片段則由持有該片段的節點負責分享給整個群組，以避免重複下載而產生過多的跨 ISP 流量。本方法透過減少跨 ISP 流量，藉此降低延遲，預期將可大幅縮短檔案平均下載完成時間。

本論文其餘部分架構如下：第二節探討相關研究，第三節提出所設計之方法，第四節為效能評估，最後為結論。

2. 相關研究

在減少跨 ISP 流量的研究可以分為兩大類：第一類是透過鄰居的選擇，盡可能尋找同 ISP 的鄰居傳輸，以有效的減少跨 ISP 的流量。在[11]中，每個節點向 tracker 取得多數位於相同 ISP 的鄰居以及少量的相異 ISP 的鄰居，如此限制向外部傳輸的比例，並透過 Local Rarest First [2]策略提高了相同 ISP 傳輸的機會，但缺點是整體傳輸時間會顯著上升。[12][13]則是在不參考 ISP 資訊的情況下設法降低跨 ISP 流量。[12]使用分散式的方法分別與所有鄰居互相測量延遲(Round Trip Time, RTT)之後，傳輸時將優先選擇 RTT 較小的鄰居，可同時降低跨 ISP 流量與平均下載時間。[13]同樣以 RTT 再加上節點與節點之間的 hop 數作為鄰居選擇的參考。這些方法主要是採用分散式的處理，較不會造成單一節點太大的負擔。

第二類則是透過片段下載任務的安排，將節點依據所在 ISP 分為數個群組，並讓每個片段在每個同 ISP 群組中只會有一個節點負責對外下載，如此就可以降低跨 ISP 流量至最小。但此方法若是當負責下載某片段的節點存取失敗，就可能造成整個群組無法取得該片段，且其他節點在等待時，頻寬會處於閒置浪費狀態。[14]先讓所有節點建立鄰居關係並先允許隨意下載，當發現同屬一個 ISP 的全部節點的 buffer 集合起來擁有檔案中的所有片段時，這個群組的節點自此只允許向群組內部發出 request，不再產生額外跨 ISP 流量，整體傳輸時間也維持原先水準。[15]提出了利用一個門檻值，當整個群組擁有的檔案片段數量超過門檻值時，將限制群組僅留有一位超級節點可繼續向外部 ISP 發出 request，其他節點只能進行群組內下載，提前進入了群組內部互傳的階段，可以有效降低更多的跨 ISP 流量。[16]提出中央分配下載的方法，把 ISP 本身看作一個單位，此時單一片段進出各 ISP 的次數就是重要的考量。依據組內所有節點的下載能力，分派各個節點去抓取不同數量、不同部分的片段，再組合起來形成完整的檔案，減少頻寬的浪費及透過互相檢查降低失敗的風險。此方法讓單一片段只會進出 ISP 一次，大大降低重複跨 ISP 的次數。此類方法因為需要群組管理，大多採用中央式群組管理，缺點即為有中央節點癱瘓的

風險。但此類方法可以讓節點之間產生合作關係，可以考慮到同群組中節點的狀態，和分散式策略相比有著更容易控制跨 ISP 流量的優點。

3. 方法設計

3.1 系統概述

我們結合上述在選擇鄰居時盡可能尋找同 ISP 的鄰居傳輸，以及節點群組管理兩種機制，設計出 Reducing Cross-ISP Traffic via Coordination and Assistance (RCTCA)方法，主要的構想是讓存在於同一個 ISP 的節點可以互相合作，彼此先分工取得不同的片段，直到這些節點的 buffer 集合起來可以合成一個完整的檔案時，再互相傳輸完成檔案下載。圖 1 中顯示一完整檔案被分割為 12 個片段，而圓形代表某節點擁有該片段。圖中三個位於相同 ISP 的節點 1、2 及 3 在分別取得了不同的片段後，只要再透過互相傳輸，每個節點即可完成各自的檔案下載，而在彼此傳輸的階段將不會產生跨 ISP 流量。

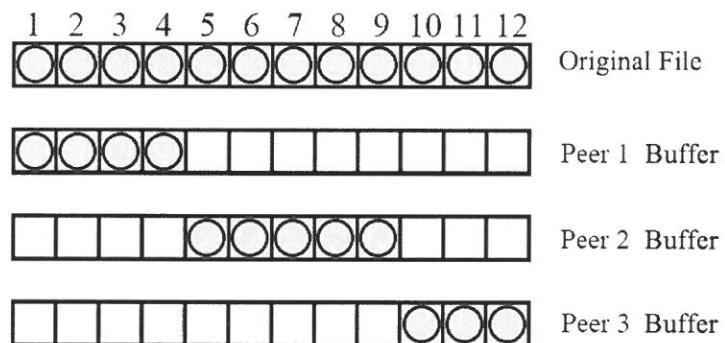


圖 1、同 ISP 節點的片段分配下載示意圖

3.2 系統機制

我們的方法主要包含四個部分：鄰居拓樸的建立、檔案的分配下載、節點彼此之間的協助機制、以及相同 ISP 下之傳輸策略，分別說明如下：

3.2.1 鄰居拓樸的建立

我們假設所有的節點皆與 seed 持續作為鄰居，並向 tracker 回報自己的 ISP 資訊。同一 ISP 的所有節點將形成一個群組，群組中的所有節點互為鄰居，每一個節點再由 tracker 分配給少量的相異 ISP 之節點作為鄰居，如圖 2 所示。因為每一個節點的下載任務各不相同，為了讓鄰居的可利用性增高，tracker 在配給相異 ISP 鄰居時，將優先提供有著相似下載任務的節點作為鄰居，如圖 2 中節點 p 與節點 m 彼此作為相異 ISP 鄰居，分配方法於下節中詳細說明。

3.2.2 檔案的分配下載

每個節點在依據 ISP 形成群組後，即被分配予不同的片段下載任務。整個檔案的片段由同組全部的節點平分下載。例如有 5 個節點處於同一個 ISP 中，想要下載一個 500 個片段的檔案，則每個節點依序分別負責 100 個不同的片段作為下載任務。只要每個節點完成下載任務，就可以確保該群組擁有該檔案全部的片段。且節點在選擇片段下載來源時，會優先選擇向鄰居下載。若沒有鄰居擁有該片段時才向 seed 下載，如此可以有效減少 seed 的負擔，善用 P2P 的優點。

節點的下載任務是在節點加入群組時，由 tracker 根據該 ISP 的節點數量及這些節點的報到順序來分配。Tracker 同時配給少量相異 ISP 鄰居給每個節點，這些鄰居將依據節點的下載任務來分配。以圖 2 作說明，圖中長方格代表節點的 buffer，圓圈代表其下載任務。每個節點最終都要取得一個分割為 12 個片段的檔案。ISP A 中 p 被分配負責下載片段 0~2、q 負責下載片段 3~5，依此類推，而 m、n、o 亦各自負責下載三分之一的片段。在鄰居可用性上，ISP A 的 q 負責片段 3~5，與 ISP B 的 n 負責片段 4~7 重複率較高，因此安排兩節點作為鄰居，較可以互相下載，免於過度依賴 seed。圖中僅繪出兩個 ISP 方便示意，實際上 ISP 數量沒有限制，每個節點會隨機分配到數個相異 ISP 的鄰居。

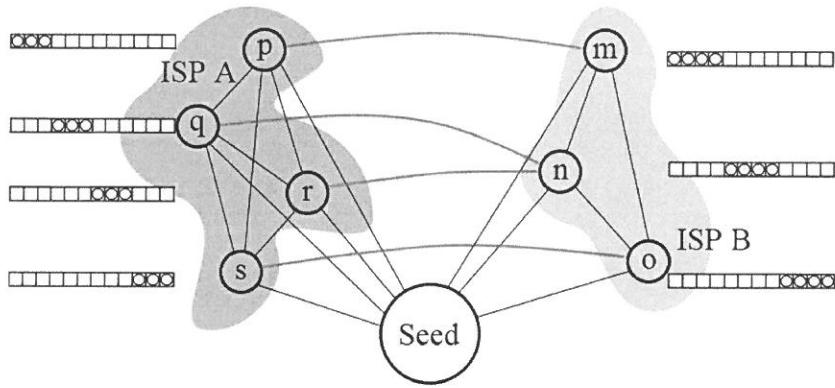


圖 2、節點鄰居關係與片段任務分配示意圖。

3.2.3 節點彼此之間的協助機制

因為每個節點的能力、鄰居情況大不相同，就算是分配相同的下載數量，也不見得能在同一時間點完成，若節點完成下載任務後只能等待其他節點完成，則會造成該節點的頻寬閒置。因此我們設計了一個協助機制，只要有節點完成了下載任務時，每個節點將依據其報到順序，往自己的下一位開始依序徵詢，嘗試去協助尚未完成任務的節點。我們假設群組中的節點依順序形成迴圈（最後一位節點的下一位即為第一節點），先行完成的節點會依序幫助群組中的下一個節點。如圖 3 所示，假設 q 已經完成自己的下載任務， q 即向 tracker 查詢在群組序列中下一位的節點 r ，接著向 r 詢問在其所負責下載的片段中，是否有尚未下載且沒有其他節點已參與協助下載的片段。如果有， r 便將這些剩餘任務分一半給 q 協助下載。圖中 r 已經沒有任務需要幫忙回絕了 q ，此時 q 再嘗試詢問序列中的下一位節點 s ，而節點 s 將下載片段 12 的任務分給 q 協助下載，以利加快相同 ISP 群組節點完成全部片段的抓取。

另外新加入的節點會向 tracker 登記後加入群組，並將自己視為已經完成下載任務，開始嘗試協助 tracker 所隨機指定的一個同群組中節點，若該節點不需要協助則再徵詢下一位。另外在協助過程中，若有節點發現有其他節點離線，該節點會馬上通知 tracker，而 tracker 將把該段任務重新分配給該發現的節點。

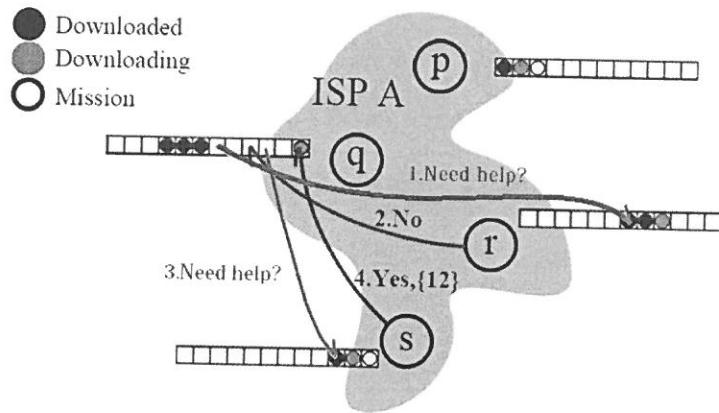


圖 3、協助下載示意圖

3.2.4 限制相同 ISP 下傳輸策略

當已經確定群組內對於檔案中的每個片段都已經有人擁有或者負責下載後，該群組將不再對外發出新的 request，並且以 Rarest First 的方法開始群組內的互相傳輸，但仍然會回應來自其他群組的節點所發出的 request。若真的因為節點離線造成某片段在群組中無任何節點擁有，才再由其中一節點向其他群組抓取。開始互相傳輸後，頻寬較大的節點(假設為 q)很有可能已經下載完大部份的片段，而僅缺少的最後幾個片段卻被其他頻寬較小的節點(假設為 s)標記為正在下載。原本 q 是要等待 s 下載完片段才能向其複製，但如此會因為等待過久而造成節點頻寬閒置，此時將允許此陷入瓶頸的節點 q 可以自由向其他群組或者向 seed 下載。

我們的方法可以盡量讓每一個節點的頻寬在檔案下載完成之前都能被充分地利用，且迅速地在自己的群組中完成檔案的抓取。每一個片段進入 ISP 的次數將趨近於 1，僅在最後階段時可能會有少量重複的跨 ISP 傳輸，如此能有效地降低跨 ISP 流量。過程中也因為各節點彼此協助完成下載整個檔案，間接達到能者多勞的概念，即高頻寬的節點負責下載較多的部分。

4. 效能評估

4.1 模擬環境

我們使用 Java 撰寫一個模擬程式，來評估我們所提出的 RCTCA 群組管理機制在不同環境下的效能。我們假設一個 seed 或節點可以同時服務並傳輸資料給最多三個請求。所有節點假設於模擬開始時已全部加入系統，每個節點當下載完成後會繼續留在線上服務其餘未完成的節點。實驗中，seed 假設擁有所有的檔案片段，單獨處於一個 ISP 中，其上傳頻寬為 6Mbps。其餘節點隨機分配於各個 ISP 中，頻寬分為高、中及低三種，其下載/上傳頻寬分別為 6Mbps/2Mbps, 3Mbps/1Mbps, 1.2Mbps/0.4Mbps。然而為了表現當節點進行跨 ISP 資料傳輸時，網路壅塞造成實際有效傳輸頻寬較低之情形，我們假設當兩節點進行跨 ISP 傳輸時，資料傳輸頻寬將減少一半。例如，節點 A 向節點 B 下載片段時，若兩節點位於相同 ISP，有效傳輸頻寬為 A 的下載頻寬與 B 的上傳頻寬兩者之較小值；若兩節點位於相異 ISP，則有效傳輸頻寬為較小值之一半。檔案大小設定為 600MB，共分割為 100 個片段，每個片段的大小為 6MB。服務策略為 First Come First Serve，詳細模擬參數如表 I 所示。

表 I、模擬環境參數設定

Seed	1 個，上傳頻寬為 6Mbps
ISP 數量	2~50 個
節點數量	10~500 個
高、中、低節點 頻寬	上傳/下載：6Mbps/2Mbps、3Mbps/1Mbps、 1.2Mbps/0.4Mbps
節點最大可同時 服務數量	3 個 request
檔案及片段大小	600MB，分為 100 個 6MB 之片段

4.2 模擬結果

在本實驗中，我們假設 ISP 的數量為 5 個，此時節點的數量增加代表著同樣的檔案被複製的次數增加，即所有片段被複製的次數增加。圖 4 及圖 5 顯示片段的來源經由 seed 以及跨 ISP 的流量和節點數量之間的關係。我們的方法基本上讓單一片段只會進出 ISP 一次，而下載此切割為 100 個片段的檔案時，每一個 ISP 都僅有 100 次的片段進出。在總共 5 個 ISP 的環境下，可以得知下載來自處在相異 ISP 的節點或者 seed 的片段數量總合應為 500。當節點數量夠多時，每一個 ISP 群組中的節點數量隨之增加，造成系統初期只要每個節點向 seed 發出請求，就已經使 ISP 群組取得檔案大部分的片段，而不需要跨 ISP 再向其它節點發出請求。如圖 4 中，當系統有 250 個節點時，每個 ISP 平均擁有 50 個節點，而系統開始時僅 seed 擁有片段，所有的節點均向 seed 發出第一次的請求，產生 250 次向 seed 下載。要發出第二次請求時，自己的鄰居擁有的片段也很少，造成再一次向 seed 發出請求的機會很高，而節點平均發出兩次請求就可以滿足群組內擁有全部 100 個不同的片段。而在類 BT 的方法中，因為採取 Local Rarest First 策略，系統初期幾乎所有節點都向 seed 發出請求，所以向 seed 下載的次數較我們的方法為多。而在選擇向哪一個鄰居發出請求是採取隨機的方式，隨著節點數量上升，總下載次數增加，向 seed 下載以及跨 ISP 下載的次數也會增加。

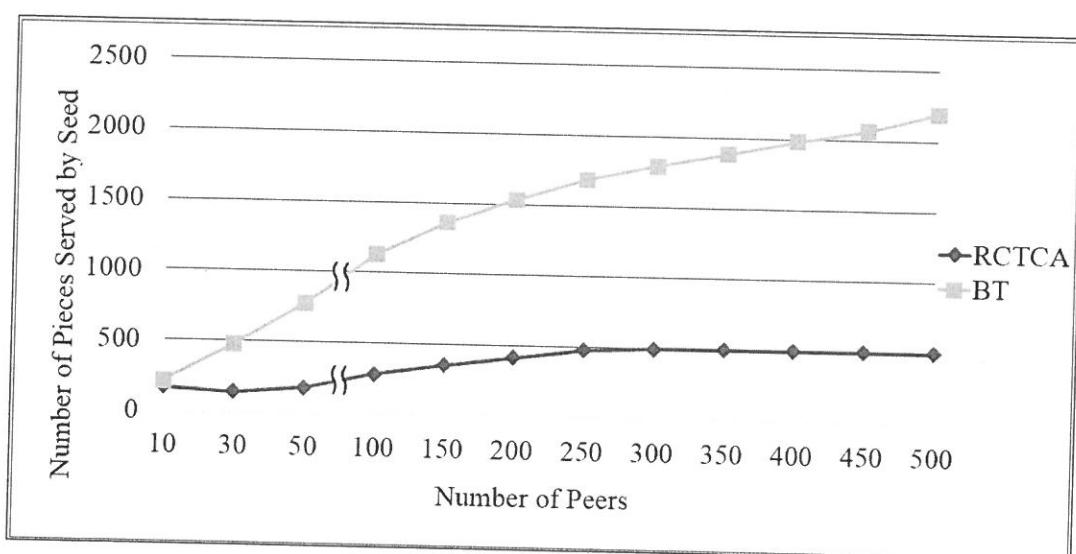


圖 4、節點數量變化時，透過 seed 下載的片段數量

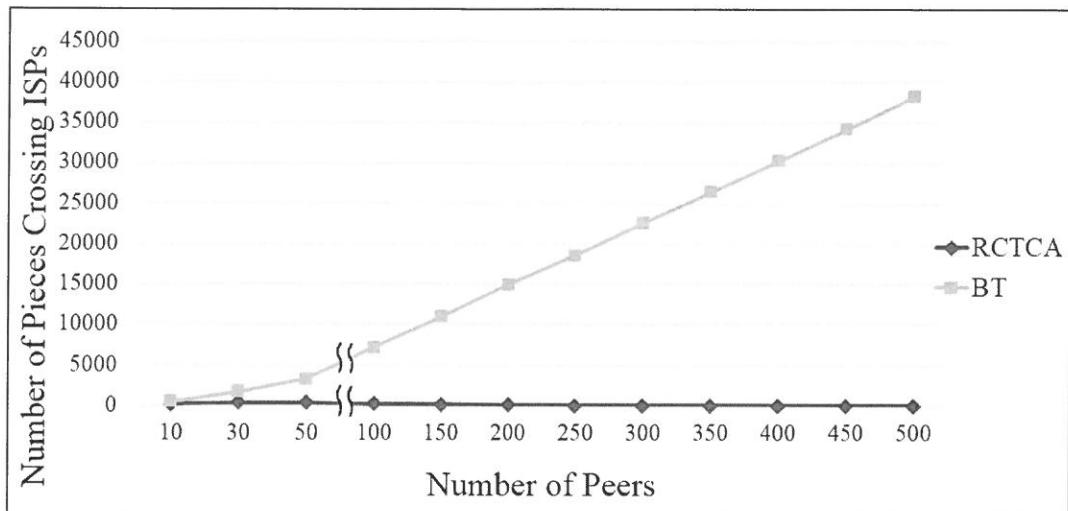


圖 5、節點數量變化時，跨 ISP 傳輸的片段數量

因為假設節點在作跨 ISP 傳輸時僅擁有平時一半的頻寬，同 ISP 中傳輸速度較快，我們提出的方法較 BT 擁有更多的同 ISP 傳輸，下載速度明顯較快，如圖 6 所示。

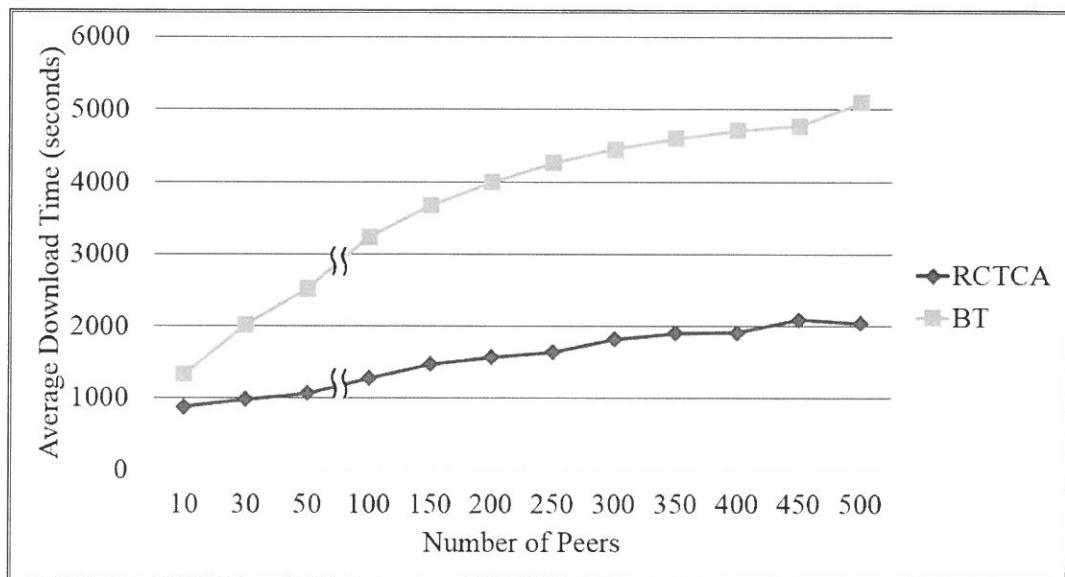


圖 6、節點數量變化時，平均下載完成時間的變化

圖 7 繪出了我們的方法中節點之間互相幫助下載的次數，該數值隨著節點平均負責的片段數量而變動。當節點數量增加時，每個 ISP 群組的節點數隨之增加，每個節點負責的片段任務因而減少，於 250 個節點時達到幫助次數的高峰。因為平均每個節點負責要下載兩個片段時，下載完的節點可以幫助其他仍在下載第一個片段的節點。而節點數量超過 250 之後，開始會有節點僅負責一個片段要下載，造成節點過多時會完全不需要協助。

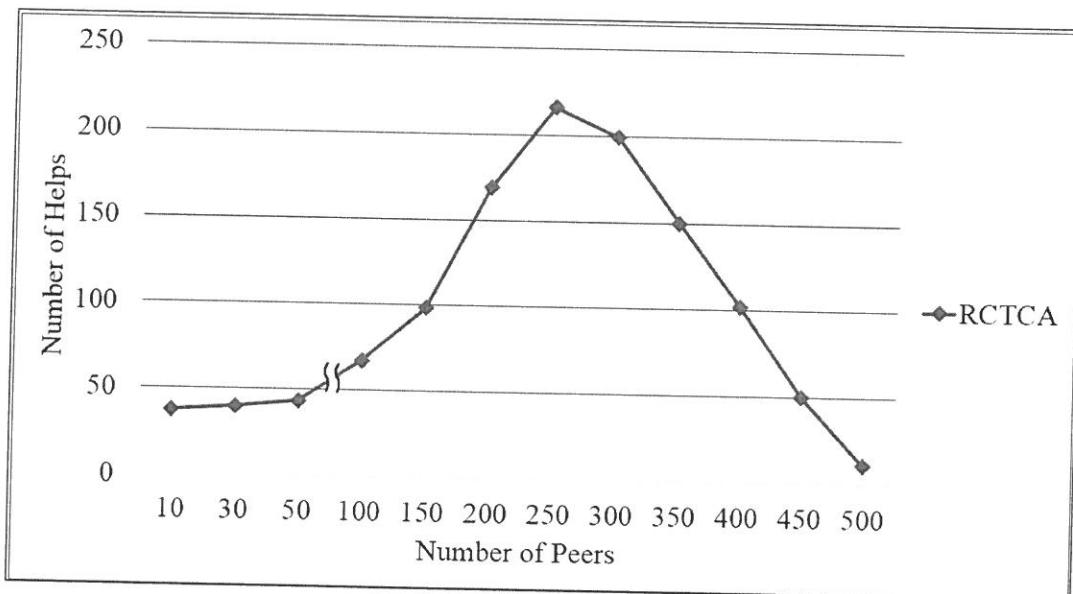


圖 7、節點數量變化時，節點幫助次數的變化

5. 結論

現今大多數的點對點傳輸技術皆著重於如何在最短的時間內讓所有節點下載完成，在鄰居分配上忽略了考量節點之間可能分佈在不同的 ISP 網路，因而產生大量跨 ISP 的流量，導致下載時間增長。為了改善此問題，我們提出了一個透過 ISP 群組管理的 RCTCA 機制。節點依據 ISP 分組，讓存在於同一個 ISP 的節點可以互相合作，彼此先分工取得不同的片段，直到這些片段集合起來可以合成一個完整的檔案時，再互相傳輸完成檔案下載。每一個片段進入 ISP 的次數將趨近於 1，僅在最後階段時可能會有少量重複的跨 ISP 傳輸，如此能有效地降低跨 ISP 流量。

過程中也因為各節點彼此協助完成下載整個檔案，間接達到能者多勞的概念，即高頻寬的節點負責下載較多的部分。在模擬實驗中，我們的方法與類 BT 方法比較的結果，隨著節點的數量增加、ISP 的種類越多，跨 ISP 傳輸流量降低 58%至 98%，而檔案下載完成時間也隨之縮短 34%至 75%。

參考文獻

- [1] F. Oberholzer-Gee and K. Strumpf, "The Effect of File Sharing on Record Sales. An Empirical Analysis," *Journal of Political Economy*, Vol. 115, February 2007, pp. 1-42.
- [2] BitTorrent, <http://www.bittorrent.com/>
- [3] S. Fanning, Napster, <http://www.napster.com>
- [4] PPStream, <http://www.pps.tv/>
- [5] X. Zhang, J. Liu, B. Li, and T. Yum, "CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-peer Live Media Streaming," *IEEE INFOCOM*, March 2005, pp. 2102-2111.
- [6] PPTV, <http://www.pptv.com/>
- [7] D. A. Tran, K. A. Hua, and T. Do, "ZIGZAG: An Efficient Peer-to-peer Scheme for Media Streaming," *IEEE INFOCOM*, 2003, pp. 1283-1292.
- [8] J. Xuxian, D. Yu, X. Dongyan, and B. Bhargava, "GnuStream: a P2P media streaming system prototype," *2003 International Conference on Multimedia and Expo*, July 2003, pp. II-325-II-328.
- [9] C. Wang, H. Wang, Y. Lin, and S. Chen, "A currency-based P2P incentive mechanism friendly with ISP," *2010 International Conference on Computer Design and Applications*, June 2010, pp.V5-403-V5-407.
- [10] M. Lin, J.C.S. Lui, and D.-M. Chiu, "An ISP-Friendly File Distribution Protocol: Analysis, Design, and Implementation" *IEEE Trans. on Parallel and Distributed Systems*, Vol. 21, Issue 9, September 2009, pp. 1317-1329.
- [11] R. Bindal et.al., "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," *IEEE ICDCS*, 2006.
- [12] L. Sheng and H. Wen, "Reducing Cross-Network Traffic in P2P Systems via Localized Neighbor Selection," *Fourth International Conference on Communications and Networking in China (ChinaCOM)*, 2009, pp. 1-5.
- [13] W. Li, S. Chen, and T. Yu, "UTAPS: An Underlying Topology-aware Peer Selection Algorithm in

- BitTorrent,” *IEEE Advanced Information Networking and Applications (AINA)*, 2008, pp.539-545.
- [14]O.-Y. Rong, C. Hui, “A Novel Peer Selection Algorithm to Reduce BitTorrent-like P2P Traffic Between Networks,”, *International Conference on Information Technology and Computer Science (ITCS)*, 2009, pp. 397-401.
- [15]侯雲笙,「在類BitTorrent系統中設定門檻以降低跨越ISP之流量」, 輔仁大學資訊工程研究所碩士學位論文, 2014, pp.1-26.
- [16]S. Singh, P. Gambhir, “Distributed Concurrent Downloading of Common Content in a BitTorrent Peer Group,” *IEEE International Conference on Signal Processing and Communications*, 2010, pp. 1-5.

