# Distributed Proximity-Aware Peer Clustering in BitTorrent-Like Peer-to-Peer Networks

Bin Xiao[1], Jiadi Yu[1,2], Zili Shao[1], and Minglu Li[2]

[1] Department of Computing
The Hong Kong Polytechnic University, Hong Kong
{csbxiao, csjdyu, cszlshao}@comp.polyu.edu.hk
[2] Department of Computer Science and Engineering
Shanghai Jiao Tong University, China
{jdyu, li-ml}@cs.sjtu.edu.cn

**Abstract.** In this paper, we propose a hierarchical architecture for grouping peers into clusters in a large-scale BitTorrent-like underlying overlay network in such a way that clusters are evenly distributed and that the peers within are relatively close together. We achieve this by constructing the CBT (Clustered BitTorrent) system with two novel algorithms: a peer joining algorithm and a super-peer selection algorithm. Proximity and distribution are determined by the measurement of distances among peers. Performance evaluations demonstrate that the new architecture achieves better results than a randomly organized BitTorrent network, improving the system scalability and efficiency while retaining the robustness and incentives of original BitTorrent paradigm.

**Keywords:** Proximity-aware, Clustered BitTorrent (CBT), peer-to-peer networks, super-peer, scalability.

## 1 Introduction

Peer-to-Peer (P2P) applications have become popular in contexts such as Internet file sharing. P2P technology has a number of advantages over the traditional server-client content distribution model. The hardware is inexpensive, it is scalable so as to accommodate a large number of users and large amounts of content, it is fault tolerant when content is being shared by multiple sources, and has faster download time.

One of the most popular P2P file-sharing applications is BitTorrent [1], [2]. BitTorrent is not, however, unproblematic. In a BitTorrent network [3], each client peer joins a separate *torrent* to share the downloading of a specific file. A torrent contains a *tracker* and peers that are currently online, all of which are sustained by a track server. This torrent tracker collects the state of every peer in the torrent and returns a random list of peers in response to the joining request of a peer in BitTorrent. The torrent tracker and all of the peers together in a torrent form a mesh overlay network. This mesh overlay network is flat and works well when the network is small. Two problems arise, however, when the network becomes larger to the point where there are tens of thousands of

E. Sha et al. (Eds.): EUC 2006, LNCS 4096, pp. 375–384, 2006.

participating peers, which is possible in a real-world BitTorrent overlay network. The first problem is that random connections among peers in the mesh overlay network make it possible for geographically distant peers to connect, increasing file downloading time. The second problem is that a large number of peers can cause bottlenecks in the torrent tracker.

It is possible to obviate these problems by using a hierarchical architecture to construct large-scale BitTorrent overlay networks. In such an approach, the overlay network is composed of several (or tens of) clusters of peers. Each cluster behaves as a BitTorrent community and keeps the robustness and incentives scheme of file sharing of the BitTorrent system. It selects a peer to be a super-peer in a cluster and this super-peer becomes a *local tracker*. The torrent tracker supervises all of the super-peers. This hierarchical architecture prevents the torrent tracker from being overloaded. If we can group peers according to their real-world proximity, we can further reduce average latencies for packet delivery between peers and increase the network bandwidth usage.

In this paper, we present a hierarchical architecture CBT (Clustered BitTorrent) for grouping proximate peers, as may be found in BitTorrent-like P2P applications. We build the scalable hierarchy overlay network using two novel algorithms, the peer joining algorithm and the super-peer selection algorithm. These allow a peer to join the cluster in which it is closest to its super-peer among all super-peers. As a result, all peers in a cluster are proximate in the underlying network topology. The proposed peer joining algorithm is a distributed algorithm in the sense that the peer itself makes the decision as to which cluster to join when it first contacts the torrent tracker. The proposed super-peer selection algorithm identifies a super-peer by selection from regular peers as that peer which is maximally distant from all existing super-peers. This distributes peers evenly among clusters and produces a small cluster diameter. Performance evaluations show that the new CBT can achieve good performance and scalability, markedly improving file download speed and reducing the torrent tracker load.

This paper is organized as follows. In the next section, we introduce related work. Section 3 describes the proposed hierarchical CBT (Clustered BitTorrent) system to be applied in BitTorrent-like P2P networks. Section 4 presents the construction of CBT applying the peer joining algorithm and super-peer selection algorithm. Section 5 shows the simulation results, and Section 6 offers our conclusions.

## 2 Related Work

The research on Peer-to-Peer (P2P) file sharing [4],[5],[6],[7] has attracted much attention. The most representative and perhaps currently most popular file sharing system is BitTorrent. Previous work, such as [2], [8], [9], mainly focus on the understanding of the BitTorrent mechanisms. Not much work proposes approaches to handling a large-scale P2P overlay network to leverage the load on the tracker, and improving file sharing efficiency. The approaches could lie in the construction of hierarchy architecture in BitTorrent and the proximity-aware joining of peers into a group. This kind of hierarchical architecture-based node clustering has been applied in decentralized P2P networks, such as FastTrack [10], SODON [11], and ECSP [12].

Research has been conducted separately as well to study the grouping of nearby peers. Beaconing scheme [13] proposes a way to find nearby peers to a given peer through the contact of several beacons, and investigates the case when the number of possible peers is large. The network-aware method [14] uses the prefixes and network information in BGP (Border Gateway Protocol) routing table snapshots to identify clusters that group nearby peers. The property has been explored in [15] that proximate peers exchanging pieces of a file can improve the file sharing efficiency in the BitTorrent network. The paper proposed a scheme to build a proximity-based overlay network, which aims at returning a list containing nearby peers to a downloader. Thus, the scheme in [15] can achieve part of robustness compared with the original BitTorrent system since some peers could be hot-spots.

## 3 CBT System

In this section, we will present the CBT (Clustered BitTorrent) system, its architecture, the file download process of a peer and functions of system elements, such as the torrent tracker and super-peer. In a BitTorrent-like P2P network that shares content among all participants, each participating peer can upload pieces of a shared file that it holds for other peers to access while simultaneously downloading other pieces. In such a scenario there could be tens of thousands of participants or even more, creating the need for a scalable system that fully utilizes network resources, such as a hierarchically clustered system.

### 3.1 System Architecture

Figure 1 shows the CBT system architecture. Peers which are close by each other are grouped in individual clusters. An unspecified number of these clusters are then grouped as a Clustered BitTorrent (CBT). When a small number of peers join a torrent, the CBT may only contain a single cluster. This cluster is called as the fundamental cluster, in which the torrent tracker is the cluster head. The CBT may have several to tens of clusters to accommodate a large number of peers. This hierarchical system is composed of a torrent tracker and three types of peers: *seed*, *downloader*, and *super-peer*. The torrent tracker is responsible for collecting the state of every peer in the fundamental cluster and for returning a random list of peers in response to the request of a new arrival to be allowed to join the fundamental cluster. Of the three types of peers, the seed is a peer that shares all the pieces of a file. The downloader is a peer that simultaneously downloads and uploads pieces of a file with others. The task of the super peer is to act like a local tracker, tracking all elements in its cluster. All super-peers are connected to the torrent tracker to form a backbone overlay network.

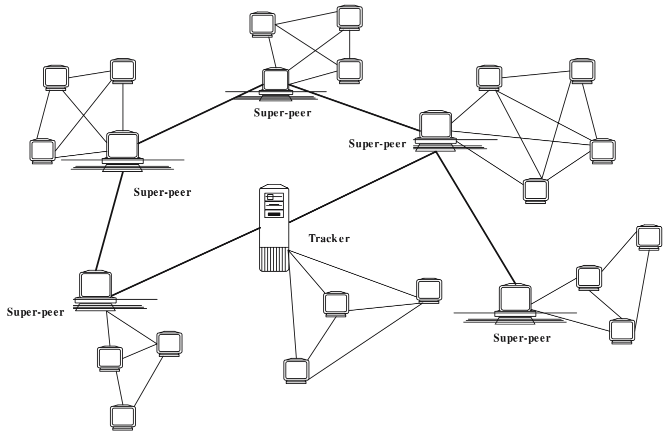



**Fig. 1.** A hierarchical architecture of CBT in a proximately grouped P2P overlay network

A cluster is a basic component in CBT. To improve the CBT system performance, we construct a cluster such that the included super-peer, seeds and downloaders are relatively close for file sharing. The downloader thus can get a shared file from nearby peers with a high speed and decrease the network traffic. In order to enhance the reliability and scalability of the system, each cluster selects a backup peer, which copies the entire cluster state information periodically from the super-peer. When a super-peer offlines or leaves the network, a selected backup peer takes over as the super-peer.

### 3.2 File Download Process

A new peer to CBT who aims at downloading a shared file needs to take two steps, the step to join a cluster and the step to exchange data with other peers in the same cluster. In the first step, a peer locally decides which cluster to join in the hierarchical P2P network. Following the same procedure as in the BitTorrent system when a user clicking on the torrent of the shared file, a peer links to the torrent tracker. In CBT, if there is only one cluster (the fundamental cluster), the peer directly joins the cluster and receives a list that contains random peers to connect from the torrent tracker. If there are more than one cluster, the peer receives a list of all super-peers (including the torrent tracker itself) from the torrent tracker. From these super-peers, the new arriving peer selects the nearest super-peer and joins its cluster.

In the second step, the downloader contacts the selected super-peer to connect with random peers in the same cluster to begin the pieces exchange process. The downloader receives the list of peers from the super-peer in the joining cluster rather than from the torrent tracker. Inside a cluster, the file downloading mechanism is exactly the same as in a BitTorrent community [3]. Each peer can apply the *choking* algorithm to provide a cooperative way of file sharing. The pieces exchanged can comply with the *rarest piece first* algorithm to assure more seeds available in the system. The random list of peers to connect with for a downloader maintains the robustness to avoid hot-spots of popular peers that may receive downloading requests from many others, and to make each peer involve in the file uploading and downloading process.

### 3.3 Functions of the Torrent Tracker and Super-Peers

The torrent tracker in the new proposed CBT system can perform not only the same functions as a tracker in the traditional BitTorrent system i.e., it collects registration information about each peer in the fundamental cluster. Also, the torrent tracker needs to monitor the super-peer backbone network. To get the information of super-peers, the torrent tracker must communicate with them periodically. Because the torrent tracker has much less job than in a traditional BitTorrent, supervising a small part of peers and the backbone super-peers, its overhead is reduced and a tracker server can handle more torrents.

A super-peer acts as a local cluster tracker in the CBT and provides file sharing as well. In a cluster as the cluster head, a super-peer collects and maintains state information of all peers in its cluster to provide information for their random connections. Meanwhile, super-peers exchange their states with the torrent tracker periodically for the cluster management. Super-peers are designed to reduce load on the torrent tracker and they are chosen among peers who are willing to serve as a super-peer. A super-peer can be selected from seeds that have more network resources to be a cluster tracker and are willing to provide the super-peer service. The selected super-peers should be evenly distributed in a global P2P overlay network such that the size of each cluster is similar with a super-peer at its center. Note that although a seed can be selected as a super-peer, it is not true that all seeds will become super-peers.

## 4 CBT Construction

When a new arrival user clicks on a torrent link to start to download a shared file from CBT, the user as a peer has to decide which cluster to join. In each cluster, a super-peer functioning as a cluster head must be decided too. In this section, we describe the peer joining algorithm for a peer to join a closer cluster, and the super-peer selection algorithm to achieve an even distribution of clusters.

### 4.1 Peer Joining Algorithm

In the hierarchical CBT system, a new arrival wishes to join a cluster such that the randomly connected peers are in its vicinity. If all peers are within a small distance from the cluster super-peer, which stands at the cluster center, the arriving peer may attain this goal by finding the nearby super-peer and joining its supervised cluster. We present the peer joining algorithm for a peer purposely joining a closer cluster. This is a distributed algorithm and can be applied locally in each client peer.

Before the presentation of the peer joining algorithm, we deliver notations to be used in the paper. Let $H\{p_0, p_1, ..., p_{k-1}\}$ be a set of peers in a network, where $k$ is the number of total peers. Let $P\{sp_0, sp_1, ..., sp_{m-1}\}$ be a set of super-peers that behave as trackers in separate clusters, where $m$ is the number of super-peers and $P \subseteq H$. Let $SP_i\{sp_i, b_0, b_1, ..., b_{n-1}\}$ ($i = 0, 1, ..., m-1$ and $n$ is the number of peers in the cluster $SP_i$) be a set of peers in the $i$th cluster. Thus, we have $SP_i \subseteq H$ and $H = SP_0 \cup SP_1 \cup ... \cup SP_{m-1}$. Let $u$ be an arriving peer that needs to find a closer super-peer. The peer joining algorithm returns a super-peer $sp_j$, such that $\forall q \in P$, $dist(u, sp_j) \leq dist(u, q)$, i.e., $sp_j$ is the closest super-peer to the new peer $u$ and $u$ should join the cluster $SP_j$. Note that $dist(u, v)$ shows the distance between two peers $u$ and $v$ and it can represent either the RTT (Round-Trip-Time) value, TTL (Time-To- Live) value or the combination of them.

**The Peer Joining Algorithm**

**Instance:** The super-peer set $P$ and $\mid P \mid = m$, clusters $SP_i$ ($i = 0, 1, ..., m-1$), the peer $u$.

**step 1.** The new peer $u$ measures the distance from itself to every super-peer in $P$, and get distance information $dist(u, sp_i)$ in terms of the RTT value ($dist_{RTT}(u, sp_i)$) and the TTL value ($dist_{TTL}(u, sp_i)$) ($i = 0, 1, ..., m-1$).

**step 2.** Find a super-peer $sp_j$ from $P$, such that $sp_j \in P$ and $\forall q \in P$, $dist(u,sp_j) \leq dist(u, q)$, i.e., the super-peer $sp_j$ is the closest super-peer to the peer $u$ in the set $P$.

**step 3.** The peer $u$ joins the cluster that contains the super-peer $sp_j$ and $SP_j = SP_j \cup \{u\}$.

The time complexity for this algorithm is $O(m)$.

## 4.2 Super-Peer Selection Algorithm

A super-peer performs the local tracker functions in a cluster. Once a new cluster has been created, a super-peer must be selected and the candidate peer should show a high tendency to stay online and be a seed. We present a super-peer selection algorithm to solve which seeds will be selected as super-peers.

The super-peer selection algorithm will choose $t$ ($t \geq 1$) seeds to be new super-peers and each new super-peer stands for a new cluster. The $t$ newly constructed clusters together with previous clusters should have a global distribution to contain all peers in a hierarchy P2P system such that the average distance from a peer to its cluster super-peer is short. Let $S\{s_0,s_1,...,s_{r-1}\}$ be a set of current seeds where $r$ is the number of seeds in $H$ and $S \subseteq H$. Given a peer $s$ and the super-peer set $P$, we define the distance between $s$ and $P$ as $Dist(s, P) = min\{dist(s, sp_0), dist(s, sp_1), ..., dist(s, sp_{m-1})\}$ where $dist(s, sp_i)$ denotes the distance between two peers $s$ and $sp_i$. The super-peer selection algorithm is represented as follows.

**The Super-peer Selection Algorithm**

**Instance:** The super-peer set $P$ and $\mid P \mid= m$, the seed set $S$ and $\mid S \mid= l$, let $t$ be the number of new super-peers to be selected and $t \leq l$.

**step 1.** Find a seed $s_i$ from $S$ such that $\forall q \in S$, $Dist(s_i, P) \geq Dist(q, P)$, i.e., the seed $s_i$ is the farthest seed in $S$ to the set $P$ of super-peers.

**step 2.** Select $s_i$ as a new super-peer and it is added to the set $P$, i.e., $P = P \cup \{s_i\}$ and $\mid P \mid=\mid P \mid +1$, $S = S - \{s_i\}$ and $\mid S \mid=\mid S \mid -1$.

**step 3.** Repeat step 1 to step 2 until the number of super-peers in the set $P$ increases to $m + t$, i.e., $\mid P \mid= m + t$.

The super-peer selection algorithm takes $O(m + t)$ time per iteration to generate a super-peer. Totally we have $t$ iterations and the time complexity for this algorithm is $O(t \cdot (m + t))$. The described super-peer selection algorithm always chooses a new seed that has the farthest distance to all existing super-peers. This algorithm leads to an even distribution of peers among clusters and achieve a small diameter for each cluster.

## 5 Performance Evaluation

In this section, we present simulation results to evaluate the original BitTorrent system and our proposed hierarchy system CBT. We designed a simulator to precisely quantify system overhead and effectiveness in terms of *download completion time* in different system sizes. The download completion time is the time required for a peer to complete downloading all pieces of a file starting from the peer joining a P2P network. We highlight that our proposed algorithms, the peer joining algorithms and the super-peer selection algorithms, show their impacts on decreasing the download completion time of peers.

In our simulations, an 8MB file, which is divided into 16 pieces and 512K per piece, was shared from seeds by variable numbers of peers concurrently. The available bandwidth between two peers for data delivery is relevant to their packet overlay distance. A wide bandwidth corresponds to a short distance and a narrow bandwidth corresponds to a long distance. Every peer can connect at most 8 other peers. The system contains 8 seeds initially and we can construct 8 clusters with those 8 seeds as super-peers.

### 5.1 Impact of Clustering Peers

We compared our hierarchical cluster system CBT, which applies the peer joining algorithm, with a BitTorrent-like system in the simulation environments with different number of peers involved. The average download completion time for each peer and how many peers finish the download job in an interval are two important factors to demonstrate the efficiency of a P2P system. The download performance is evaluated in terms of the percentage of peers to be seeds

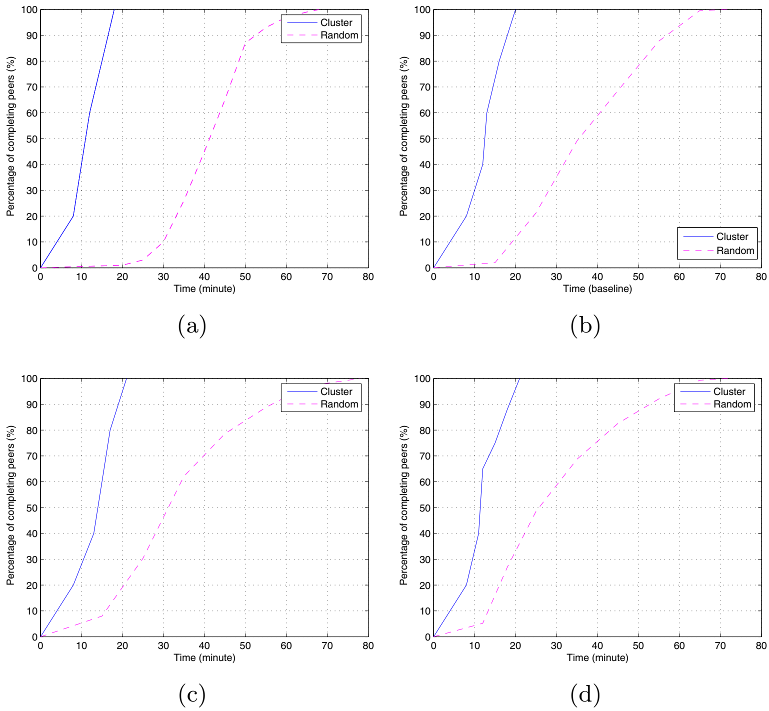


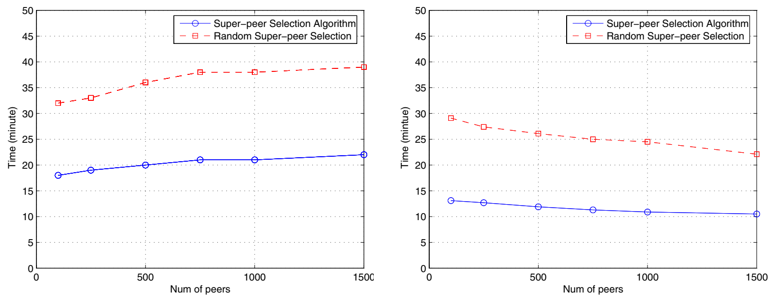

**Fig. 2.** Percentage of peers to be seeds in different scales in a network (a) 100 peers; (b) 500 peers; (c) 1000 peers; (d) 1500 peers

(Figure 2). The simulation results show that the cluster system (CBT) can achieve a faster download speed and higher file availability than the random-connection BitTorrent network does. In the Figure, the *Cluster* and *Random* curves represent the simulation results of the CBT system and random-connection BitTorrent network respectively.

Figure 2(a)-(d) display the percentage of peers who complete all pieces of a shared file to become seeds in networks of different scales. We chose respectively 100, 500, 1000 and 1500 concurrently running peers to evaluate two compared systems, CBT and BitTorrent. In both systems, the percentage of peers to be seeds increases gradually as the simulation time moves forward. However, the percentage of peers to be seeds in CBT is always higher than that of the random network as in BitTorrent. As denoted in Figure 2(c), the time for the last peer to complete the job is around 20 minutes in the cluster system (CBT) while it is around 80 minutes in the random network, which is 4 times slower. This means that the total download time for all peers completing their jobs is remarkably reduced in our system. In other words, the CBT system obtains a faster speed to render every peer becoming a seed than a random-connection BitTorrent network does.

### 5.2 Impact of Super-Peer Selection

The super-peer selection shows its impact on the file sharing efficiency in hierarchical P2P networks, like CBT. A well designed super-peer selection algorithm can evenly distribute all peers into distinct clusters. In this subsection, we show that by applying the proposed super-peer selection algorithm, all downloaders can quickly get all pieces of a file compared to the system with random super-peer selection mechanism. We show their comparison results in terms of the maximum (Figure 3(a)) and average (Figure 3(b)) download completion time.




(a) (b)

**Fig. 3.** Download completion time (a) Maximum download completion time; (b) Average download completion time

Figure 3(a) displays the maximum download completion time in all downloaders given varied number of peers in networks with and without the super-peer selection algorithm. The maximum download completion time denotes the required time for the last peer to hold the entire file. The simulation result depicts that the system employing the super-peer selection algorithm can obtain almost half the time as in the system without such algorithm. In the testing system with 1500 peers, the maximum download completion time is less than 40 minutes in a hierarchical system, which is still less than the time required in a flat random connection network (e.g., BitTorrent) as shown in Figure 2(d).

In Figure 3(b), we plot the average download completion time for varied number of peers in networks with and without the super-peer selection algorithm. The simulation result demonstrates that, with the super-peer selection algorithm, the average download completion time is shorter than that without it. For example, a system applying the proposed super-peer selection algorithm can decrease the average download completion time up to about 50% in the network having 500 peers, and up to about 60% in the network having 1000 peers.

## 6 Conclusion

To build a scalable large-scale hierarchical overlay network, we propose a novel architecture in the CBT system that employs the peer joining algorithm and super-peer selection algorithm. The first algorithm addresses how to form clusters and the second algorithm determines the local tracker in each cluster. To precisely group a peer into a cluster, we can use the proximity measurements of both the RTT value and TTL value between a pair of peer and super-peer. Such measurements are practical in a real P2P overlay network environment. This paper provides simulation results to demonstrate that the distributed proximity-aware peer clustering method is able to achieve good performance and scalability, and can be used to build a large-scale BitTorrent-like P2P overlay network.

## References


1. T. Karagiannis, A. Broido, N. Brownlee, kc claffy, and M. Faloutsos, "Is p2p dying or just hiding?," in *Proceedings of Globecom,* Dallas, TX, USA, November, 2004.
2. J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips, "The BitTorrent P2P file-sharing system: Measurements and analysis," in *Proceedings of IPTPS*, 2005.
3. B. Cohen, "Incentives Build Robustness in BitTorrent," in *Proceedings of P2P Economics Workshop*, 2003.
4. V. N. Padmanabhan and K. Sripanidkulchai, "The case for cooperative networking," in *Proceedings of IPTPS*, 2002, pp. 178-190.
5. Boon Thau Loo, Joseph M. Hellerstein, Ryan Huebsch, Scott Shenker, Ion Stoica, "Enhancing P2P File-Sharing with an Internet-Scale Query Processor," in *Proceedings of VLDB 2004*, pp.432-443.
6. Christoph Lindemann,Oliver P. Waldhorst, "A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications," In *Proceedings of. International Conference on Peer-to-Peer Computing (P2P)*, September 2002, pp. 73-80.
7. K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," In *Proceedings of the. nineteenth ACM symposium on Operating systems*, ACM Press, 2003, pp. 314-329.
8. L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, Analysis, and Modeling of BitTorrent-like Systems," in *Proceedings of Internet Measurement Conference 2005 (IMC 2005)*, New Orleans, LA, USA, October, 2005.
9. D. Qiu, and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proceedings of SIGCOMM04*, Aug. 2004, pp. 367-378.
10. FastTrack website, http://www.fasttrack.nu/
11. P. Zheng and C. Wang, "SODON: A High Availability Multi-Source Content Distribution Overlay," in *Proceedings of ICCCN*, Chicago, USA, October, 2004, pp. 87-92.
12. J. Li, and S.T. Vuong, "An Efficient Clustered Architecture for P2P Networks," in *Proceedings of the 18th International Conference. on Advanced Information Networking and Applications*, 2004, pp. 278-283.
13. C. Kommareddy, N. Shankar and B. Bhattacharjee, "Finding close friends on the Internet," in *Proceedings of 9th ICNP*, 2001, pp. 301-309.
14. B. Krishnamurthy and J. Wang, "On Network-Aware Clustering of Web Clients," in *Proceedings of ACM Sigcomm*, Sweden, 2000, pp. 97-110.
15. A. Qureshi, "Exploring Proximity Based Peer Selection in BitTorrent-like Protocol," *MIT 6.824 student project*, 2004, Available at: http://pdos.csail.mit.edu/6.824-2004/reports/asfandyar.pdf.