The Performance and Locality Tradeoff in BitTorrent-like P2P File-Sharing Systems

Wei Huang, Chuan Wu, Francis C.M. Lau
Department of Computer Science, The University of Hong Kong
{whuang,cwu,fcmlau}@cs.hku.hk

***Abstract*—The recent surge of large-scale peer-to-peer (P2P) applications has brought huge amounts of P2P traffic, which significantly changes the Internet traffic pattern and increases the traffic-relay cost at the Internet Service Providers (ISPs). To alleviate the stress on networks, localized peer selection has been proposed that advocates neighbor selection within the same network (AS or ISP) to reduce the cross-ISP traffic. Nevertheless, localized peer selection may potentially lead to the downgrade of downloading speed at the peers, rendering a non-negligible tradeoff between the downloading performance and traffic localization in the P2P system. Aiming at effective peer selection strategies that achieve any desired Pareto optimum in face of the tradeoff, in this paper, we characterize the performance and locality tradeoff as a multi-objective b-matching optimization problem. In particular, we first present a generic maximum weight b-matching model that characterizes the tit-for-tat in BitTorrent-like peer selection. We then introduce multiple optimization objectives into the model, which effectively characterize the performance and locality tradeoff using simultaneous objectives to optimize. We also design fully distributed peer selection algorithms that can effectively achieve any desired Pareto optimum of the global multi-objective optimization, that represents a desired tradeoff point between performance and locality in the entire system. Our models and algorithms are supported by rigorous analysis and extensive simulations.**

I. Introduction

The large volumes of P2P traffic in today's Internet have significantly changed the Internet traffic pattern and dramatically increased the traffic-relay cost at the ISPs. Such a cost threat has led to ISPs' packet filtering and rate throttling towards P2P traffic [1], while on the other hand P2P application providers react by encrypting data and communicating with dynamic ports to prevent from being recognized [2]. There have recently emerged hot arguments that such a conflict cannot lead to desirable outcomes for both parties. Instead, traffic localization designs have been proposed that connect peers to nearby (local) neighbors in terms of delay, routing hop count, etc.,by approaches at either the P2P application side [3], [4] or ISP side [5], or based on collaborations between both parties [6].

While such local peer selection is effective in reducing P2P traffic across network boundaries, it may unfavorably degrade the downloading performance at peers in a BitTorrent-like file-sharing system [7], as local peers may not necessarily be ones that can supply large upload bandwidths. In another word, a non-negligible tradeoff may exist between the downloading performance and the traffic localization in the system.

Given such a realistic situation, a practical solution for the benefits of both the P2P application and the ISPs, is to achieve a desired tradeoff point between performance and locality in the P2P system, that is acceptable and possibly decided by both parties. Intriguing questions thus arise: How can one formally characterize such a tradeoff between performance and locality? How can one design effective and fully decentralized peer selection strategies, that achieve any desired tradeoff in a practical system?

To address these challenges, we novelly characterize the performance and locality tradeoff in peer selection as a multi-objective b-matching optimization problem, with the two objectives of downloading speed maximization and network cost minimization simultaneously; we design effective peer strategies that achieve any pre-set Pareto optimum (the tradeoff point) of the global multi-objective optimization in the fully distributed fashion. The original contributions of this paper include: First, we present a generic maximum weight b-matching model that characterizes tit-for-tat (TFT) peer selection in BitTorrent-like P2P systems. Second, we introduce multiple optimization objectives into the generic model, and the resulting multi-objective optimization problem effectively characterizes the performance and locality tradeoff in peer selection. Third, we design fully distributed peer selection algorithms that effectively achieve the desired Pareto optimum in the entire system, as supported by rigorous proof. The correctness and efficiency of our models and algorithms are also validated using extensive simulations.

In the remainder of the paper, we present a generic maximum weight b-matching peer selection model in Sec. II, and extend the model into a multi-objective optimization problem that characterizes the performance and locality tradeoff in Sec. III. The distributed algorithm to derive Pareto optimal peer selection is discussed in Sec. IV. We evaluate the algorithm using trace-driven simulations in Sec. V, discuss related work in Sec. VI, and conclude the paper in Sec. VII.

II. Maximum Weight B-Matching Based Peer Selection: A Generic Model

A BitTorrent-like P2P file sharing network can be modeled as a directed graph $\mathcal{G} = (V, E)$ with vertices in V representing peers and edges in E connecting mutually selected peers. The tit-for-tat (TFT) mechanism, *i.e.*, peer i uploads to peer j if and only if peer j uploads to peer i, can be represented by a pair of directed edges established between the two nodes, which is referred to as a *matching* between two nodes in the graph.

We use binary variable x_{ji} to denote whether peer i wishes to download from peer j (1=yes, 0=no), *i.e.*, the data flows from peer j to peer i if the directed edge (j,i) is established. When $x_{ij} = x_{ji} = 1$, peer i and j both request to download from each other and there will be a matching between peer i and j in the P2P graph.

We use preference function $q_{ji}(x_{ji}) : \{0,1\} \to [0,+\infty)$ to represent peer i's preference in selecting peer j to download from. A higher preference value can reflect (a) a larger upload bandwidth from peer j to peer i, or (b) lower inter-ISP traffic relay cost (better traffic localization) from peer j to peer i. A concrete preference function will be discussed in Sec. III which characterizes the performance and locality tradeoff. For now, we only need to assume that q_{ji} is non-decreasing and quasi-linear; $q_{ji}(1)$ is peer i's preference in downloading from peer j and $q_{ji}(0) = 0$.

Let b be the maximum number of download connections each peer can establish. Let N_i denote the neighborhood of peer i containing known peers it learns from a tracking server in the BitTorrent-like system. Our peer selection problem at hand is to decide at each peer i the subset of neighbors in N_i to actually request to download from. Such a peer selection problem at peer i can be modeled into the following optimization problem, given the requests to download peer i itself has received from other peers (*i.e.*, $x_{ij}, \forall j \in N_i$):

$$\max \sum_{j \in N_i} q_{ji}(x_{ji}) \tag{1}$$

subject to:

$$\begin{aligned} &\sum_{j \in N_i} x_{ji} \le b, \\ &x_{ji} = x_{ij}, \forall j \in N_i, \\ &x_{ji} \in \{0,1\}, \forall j \in N_i. \end{aligned} \tag{2}$$

The constraints in (2) characterize the TFT mechanism in a BitTorrent-like system: Only when peer i uploads to peer j upon request ($x_{ij} = 1$), would peer j possibly upload to peer i ($x_{ji} = 1$). Given neighbors' current requests $x_{ij}, \forall j \in N_i$, the optimization in (1) derives the optimal values of x_{ji}'s, $\forall j \in N_i$, at peer i, *i.e.*, the best up-to-b neighbors that peer i will select to download from, in order to maximize its aggregate preference.

Putting all the local optimizations at peers together, we obtain the following global optimal peer selection problem in the entire P2P network:

$$\max \sum_{i \in V} \sum_{j \in N_i} q_{ji}(x_{ji}) \tag{3}$$

Subject to:

$$\begin{aligned} &\sum_{j \in N_i} x_{ji} \le b, \forall i \in V, \\ &x_{ji} = x_{ij}, \forall i \in V, j \in N_i, \\ &x_{ji} \in \{0,1\}, \forall i \in V, j \in N_i. \end{aligned}$$

Taking $q_{ji}(1)$ as the weight associated with the directed edge (j,i) in the P2P graph, the global optimization problem in (3) is essentially a *maximum weight b-matching problem* [8]. We propose a fully decentralized algorithm to solve the problem and achieve stable and optimal peer selection in the entire P2P network in Sec. IV.

III. Characterizing the Performance and Locality Tradeoff: The Multi-Objective Model

We now extend the generic matching-based model in the previous section to optimal peer selection that addresses the tradeoff between downloading performance and neighbor locality. At each peer, the downloading performance refers to its aggregate downloading rate from selected peers and the neighbor locality is reflected by the overall inter-ISP traffic relay cost (referred to as *network cost* hereinafter) incurred by downloading from the selected neighbors.

A. Multi-Objective Peer Selection

At each peer i, we use a non-negative constant r_{ji} to denote the maximum rate that peer i can download from peer j. Let c_{ji} be the non-negative network cost incurred by downloading from peer j to peer i. We assume the network cost between any pair of peers could be assigned based on the peering relationship of their corresponding ISPs, or using metrics such as the *p-distance* in P4P [6], that reflect the network policy and the current network status.

We use a vector-valued function [9] to represent the preference function $q_{ji}(x_{ji})$ in (1): $q_{ji}(x_{ji}) = \begin{pmatrix} r_{ji}x_{ji} \\ -c_{ji}x_{ji} \end{pmatrix}$. The new objective function in peer i's neighbor selection, which reflects the tradeoff between downloading rate maximization and network cost minimization, is as follows:

$$\max \sum_{j \in N_i} q_{ji}(x_{ji}) = \begin{cases} \max \sum_{j \in N_i} r_{ji}x_{ji} \\ \min \sum_{j \in N_i} c_{ji}x_{ji} \end{cases}$$

The global multi-objective optimal peer selection problem is (an extension from the global optimization problem in (3)):

$$\begin{cases} \max \sum_{i \in V} \sum_{j \in N_i} r_{ji}x_{ji} \\ \min \sum_{i \in V} \sum_{j \in N_i} c_{ji}x_{ji} \end{cases} \tag{4}$$

Subject to:

$$\begin{aligned} &\sum_{j \in N_i} x_{ji} \le b, \forall i \in V, \\ &x_{ji} = x_{ij}, \forall i \in V, j \in N_i, \\ &x_{ji} \in \{0,1\}, \forall i \in V, j \in N_i. \end{aligned}$$

This multi-objective optimization aims to derive the best peer selection strategies in the entire P2P network, that maximize the aggregate downloading rates and minimize the overall network costs incurred. Nevertheless, in multi-objective optimization, *optimal* solutions which achieve all objectives concurrently do not usually exist [9], *i.e.*, there commonly exists a tradeoff among the multiple objectives. In our optimal peer selection, there may not exist ideal optimal strategies and a tradeoff has to be compromised between both of our objectives. In what follows, we discuss how a *Pareto optimal* solution can be derived, that achieves any desired tradeoff of both objectives.

B. Pareto Optimal Solutions

A feasible solution to a multi-objective optimization problem is *Pareto optimal* if there is no other feasible solution which performs better than it, with respect to all objectives [9]. In our optimization in (4), feasible $\mathbf{x}^*$ is *Pareto optimal* if there does not exist feasible $\mathbf{x}$, such that $\sum_{i\in V}\sum_{j\in N_i} r_{ji}x_{ji} > \sum_{i\in V}\sum_{j\in N_i} r_{ji}x^*_{ji}$ and $\sum_{i\in V}\sum_{j\in N_i} c_{ji}x_{ji} < \sum_{i\in V}\sum_{j\in N_i} c_{ji}x^*_{ji}$.

A typical technique to find a *Pareto optimal* solution is *scalarization*, that converts the multi-objective problem into a regular optimization problem with a scalar objective function, that is the linear weighted combination of the original multiple objectives [9]. Introducing weights α and β ($\alpha+\beta=1, \alpha\geq 0, \beta\geq 0$) for the bandwidth maximization objective and the cost minimization objective, respectively, our multi-objective problem in (4) can be converted to the following:

$$\max \alpha \sum_{i\in V}\sum_{j\in N_i} r_{ji}x_{ji} - \beta \sum_{i\in V}\sum_{j\in N_i} c_{ji}x_{ji} \tag{5}$$

Subject to:
$$\sum_{j\in N_i} x_{ji} \leq b, \forall i\in V,$$
$$x_{ij} = x_{ji}, \forall i \in V, j\in N_i,$$
$$x_{ji} = \{0,1\} \forall i\in V, j\in N_i.$$

By solving the above linear program using different values of α and β, we can derive different *Pareto optimal* solutions to the multi-objective problem in (4). Therefore, given a weight pair which reflects the desired tradeoff between the two objectives (as can be decided by the P2P provider), the linear program derives the *Pareto optimal* peer selection strategy that achieves the desired tradeoff.

We further note that if there does exist an optimal solution to the multi-objective problem which optimizes both objectives concurrently, it can be derived by solving the linear program using any non-negative weights satisfying $\alpha+\beta=1$ [9].

IV. Distributed Multi-Objective Peer Selection

We now design a fully decentralized algorithm that achieves stable and optimal peer selection with respect to the generic optimization model in Sec. II, and apply it in multi-objective peer selection to achieve any desired tradeoff between performance and locality as discussed in Sec. III.

A. Generic Preference Based Peer Selection Algorithm

We solve the global optimization problem in (3) with the generic preference objectives using a distributed algorithm, in which each peer i iteratively carries out its optimal neighbor selection based on its local optimization in (1). The algorithm is given in Algorithm 1.

In the distributed algorithm, peer i ranks all known neighbors according to the preferences $q_{ji}(1), \forall j\in N_i$, into its preference list $W(i)$. It sends requests to download to peers which rank highest in the preference list and adds those requested peers into its proposal list $P(i)$; at each neighbor j which receives this request, it adds i into its receiving list $R(j)$ (lines $4-7$ in the main procedure). Peer i then waits for requests from others, and places those peers to which it has requested a download connection and from which it has received a request too (the matched peers) into its matching list M(i) (lines $14-28$). Matchings are also dynamically adjusted in order to achieve the best peer selection that maximizes aggregate preferences at each peer: if peer i's matching list is full with b peers when it receives a request from a peer it prefers more than a matched peer, it will replace the least preferred peer in M(i) with the new requesting peer (lines $19-22$); When peer i's matching list is full, it will

Algorithm 1 Peer Selection Algorithm at Peer i

Notation

W(i): Preference list at peer i, containing all known node(s) in N_i, ordered by their preference values $q_{ji}(1), \forall j\in N_i$.
P(i): Proposal list at peer i, containing the node(s) to which it has requested to download from.
R(i): Receiving list at peer i, containing the node(s) which has (have) requested to download from peer i.
M(i): Matching list at peer i, containing the node(s) with which it has established a matching.
rank(j,i): Rank of peer j at i, according to its preference $q_{ji}(1)$.

```
1: procedure REMOVE(node i, list X, node k)
2:   Remove i from node k's list X(k)
3: end procedure

1: procedure ADDMATCHING(node i, node j)
2:   M(i) := M(i) + {j}, Remove(j, R, i), Remove(j, P, i)
3:   M(j) := M(j) + {i}, Remove(i, P, j), Remove(i, R, j)
4: end procedure

1: procedure REPLACE(node i, node j, node k)
2:   Remove(k, M, i), Remove(i, M, k), AddMatching(i, j)
3: end procedure

1: procedure MAIN PROCEDURE
2: Get W(i) from the tracking server
3: Repeat 4-28 until M(i) does not change any more
4:   if the number of peers in matching list |M(i)| < b then
5:       pick peer j in W(i) with the highest rank, Remove(j, W, i)
6:       send j a request to download, P(i):=P(i)+{j}
7:       inform j to do R(j):=R(j)+{i}
8:   else /*|M(i)| = b*/
9:      for each j in P(i) such that rank(j,i) < rank(k,i), where
        k is a peer with the lowest rank in M(i) do
10:             Remove(j, P, i), Add j to W(i)
11:             inform j to do Remove(i, R, j)
12:     end for
13:  endif
14:  if Receiving list R(i) is not empty then
15:     pick peer j from R(i) with the highest rank
16:     if j∈P(i), i.e., i has sent a request to j
17:        if the number of peers in matching list |M(i)| < b then
18:          AddMatching(j, i)
19:        else if ∃k ∈ M(i), rank(k,i) < rank(j,i) then
20:              Replace(i, j, k)
21:              add k to W(i)
22:            endif
23:        endif
24:     else if |M(i)| < b or ∃k ∈ M(i), rank(k,i) < rank(j,i)
25:              send j a request to download, P(i):=P(i)+{j}
26:           endif
27:     endif
28:  endif
29: end procedure
```

also withdraw its previous requests sent to neighbors whose preferences are lower than those of its matched peers (lines $9-12$). The algorithm repeats at each peer until no more changes occur to its matching list.

B. Analysis of Algorithm Optimality

We next show that such a distributed iterative algorithm converges to a stable maximum weight b-matching among peers in the entire network, which represents the optimal solution to the global optimization problem in (3), under a mild assumption.

Definition 1. *In an undirected graph $\mathcal{G} = (V, E)$, a stable maximum weight b-matching is a subgraph $\mathcal{M}$ of $\mathcal{G}$ which satisfies: (1) it contains all nodes in V and each node is incident with at most b edges in E; (2) no edges in $\mathcal{M}$ changes any more; (3) the sum of weights (preferences) on all incident edges at all nodes in $\mathcal{M}$ is no smaller than that in any other b-matching $\mathcal{M}'$ in $\mathcal{G}$.*

Assumption 1. *Given the preference lists at the peers, there does not exist a preference cycle in the network,* i.e.*, there is no such a sequence of peers, $p_0, p_1, \cdots, p_{m-1} (m \geq 3)$, such that p_i prefers p_j (where $j = (i+1)$ MOD m) than any other peers in the sequence, $i = 0, 1, \cdots, m-1$.*

Such an assumption largely holds when a peer's preferences towards different candidate peers are different in the P2P network. The variant preference values can be achieved by introducing a small random error into the preferences derived, *e.g.*, based on downloading rate and network cost as discussed in Sec. III-A.

Theorem 1. *Under assumption 1, the distributed algorithm in Algorithm 1 converges to a stable maximum weight b-matching in the network, that represents the optimal peer selection defined by the global optimization problem in (3).*

Proof Sketch: We prove that the algorithm converges to a stable b-matching, by showing that (under assumption 1) the same b-matching in the network won't appear more than once throughout the execution of the iterative algorithm and the total number of possible b-matchings in the network is finite.

We prove the maximum weightedness of the achieved stable b-matching, by showing that the matching won't stabilize if there exists a better b-matching resuming larger overall weights (preferences) along its edges than the current one. □

Due to space limit, interested readers are referred to our technical report [10] for the detailed proof.

C. Applying to Multi-Objective Peer Selection

Algorithm 1 can be directly applied to derive the Pareto optimal peer selection striking any desired performance and locality tradeoff, by using the following combined multi-objective preference function at each peer i:

$$q_{ji}(x_{ji}) = \alpha r_{ji} x_{ji} - \beta c_{ji} x_{ji}, \forall j \in N_i. \tag{6}$$

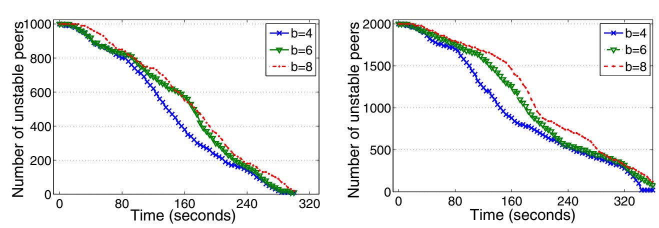
In particular, each peer i ranks its known neighbors using the above preference function (6) into its preference list $W(i)$, and carries out Algorithm 1 in an iterative fashion. Algorithm 1 at peer i derives its local Pareto optimal peer selection based on the local optimization in (1); the iterations of the algorithm at all peers converge to the global Pareto optimal peer selection as is the solution to (5).

V. Performance Evaluation

We implement Algorithm 1 in our BitTorrent simulator implemented using C++, and simulate a P2P swarm with up to 2000 peers, that download a file of 128 MB. Parameter settings of the experiments are based on practical data/distributions derived from real-world traces from field tests of P4P [6] using Pando clients in 2008, as provided by authors of P4P: Peers' upload capacities follow a heavy-tailed Pareto distribution in the major range of [256 Kbps, 10 Mbps] with the shape parameter of $k = 3$, corresponding to a mean upload capacity of 384 Kbps. The maximum rate that a peer i can download from a peer j with upload capacity u_j, *i.e.*, r_{ji}, is decided by the upload bandwidth share u_j/b that j can provide. Each peer is assigned 15 existing peers in the swarm upon joining.

The peers in our swarm are uniformly randomly assigned to 10 ISPs. We assign a cost value to each pair of ISPs to redirect their peering relationship, which are different numbers chosen from the range of [0, 700]; a larger number represents a higher traffic relay cost from one ISP to another, and the cost is 0 within the same ISP. The network cost incurred by downloading from peer j to peer i, c_{ij} in our algorithm, is set to be the cost value between their corresponding ISPs.

We first investigate the convergence of our distributed iterative algorithm in P2P swarms of different sizes with different numbers of download connections allowed at each peer (*i.e.*, b). In our experiments, when a peer finishes its own file downloading, it remains and continues uploading to its matched peers until all finish downloading. Fig. 1 shows the evolution of the number of unstable peers in the network (*i.e.*, those who are still changing their peer selection), in P2P swarms with 1000 peers and 2000 peers, respectively. In all cases, the number of unstable peers decreases quickly, *i.e.*, peer selection in the network converges quickly to the stable b-matching. Considering that a peer needs $40-50$ minutes on average to download the entire file of 128 MB, such a convergence time of $5-6$ minutes is minor, and the peers are already downloading using the current peer matchings while adjusting to the best peer selection.

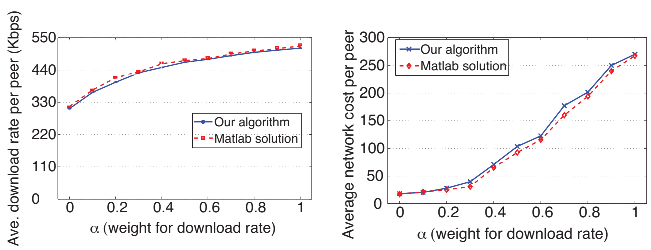


(a) 1000-peer swarm (b) 2000-peer swarm

Fig. 1. Convergence of P2P matching in swarms of different sizes.

We then investigate the optimality of the stable b-matching (peer selection in the network), by comparing the downloading rates and network costs in the converged network with the optimal solutions of the global optimization problem in (4) derived using Matlab. We experiment in P2P swarms with 2000 peers and $b = 6$, under different settings of the weight for download rate, α, and the weight for network cost, β ($\beta = 1 - \alpha$). Comparing Fig. 2(a) and 2(b), we clearly observe the tradeoff between performance and locality in P2P swarms under realistic settings: the larger α is, *i.e.*, the more weight a peer puts on download rate maximization in its peer selection, the higher the aggregate download rate per peer is in the resulting b-matching, at the cost of increased aggregate network cost at each peer simultaneously.

In both figures, the curve derived by our algorithm and the curve showing Matlab solution largely overlap with each other, validating the global optimality of the resulting peer selection by our algorithm. The small gaps between the curves can be explained that the download rate (network cost) by our algorithm is computed as the average per-peer download rate over its file downloading process, including the rates (costs) it obtains when the peer selection has not stabilized; on the other hand, the Matlab solutions shown represent the download rates (network costs) in stabilized matchings. The fact that the gaps are minor has further validated the insignificant influence of algorithm convergence time, as shown in Fig 1, in the overall downloading process at the peers.



(a) Download rate (b) Network cost

Fig. 2. Performance and locality comparisons between optimal peer selections derived by our algorithm and Matlab.

VI. Related work

A number of proposals have emerged in recent years on P2P traffic localization designs, in order to reduce the ever-increasing P2P inter-ISP traffic. From the P2P application perspective, biased peer selection towards local peers within the same AS or ISP is discussed [3], [4], [11]. From the perspective of ISPs, Saleh *et al.* [5] explore the potential of deploying proxy caches in different ASs to alleviate the load on the Internet backbone. Promoting collaborations between both parties, P4P [6] presents a novel architecture by which ISPs provide P2P applications necessary information for them to make peer selection decisions, which honor network policy and the current network status.

Le Blond *et al.* [7] have recently evaluated the impact of P2P locality on inter-ISP traffic volume and peer download completion time, using extensive experiments on a controlled environment with $10,000$ BitTorrent peers. Large inter-ISP traffic reduction is observed, while a certain level of locality's negative impact on peer download time is also revealed.

Different from all these work, our aims in the paper are to mathematically characterize the tradeoff between download performance and traffic localization (that has been largely ignored in existing studies) using matching-based optimization models, and to design fully distributed, effective peer selection algorithms to actually achieve any desired tradeoff point.

With respect to matching-based P2P modeling, Mathieu *et al.* [12], [13] have studied a b-matching model for preference-based collaborator selection in BitTorrent. They focus on analysis of the convergence speed of the matching and properties of the stabilized system. Differently, we not only model optimal peer selection into a b-matching-like optimization problem, but also design an algorithm to derive the optimal solution in a fully decentralized fashion.

VII. Concluding Remarks

This paper targets formal characterization of the tradeoff between performance and locality in a BitTorrent-like P2P system, and effective design of optimal peer selection strategies to achieve any desired tradeoff. Using multi-objective matching-based optimization, we effectively characterize the tradeoff, as well as design fully distributed optimization algorithms to carry out the peer selection. Both analytical proof and simulation results verify that our algorithm achieves global Pareto optimal peer selection, as represents a desired tradeoff between performance and locality in the network. In addition, we are fully aware of the simplifications that the current model represents of a practical BitTorrent-like protocol, and we are extending the model and algorithm to address more practical scenarios, such as seeding, optimistic unchoking, as well as peer dynamics, in our ongoing work.

References

[1] *Comcast is Using Sandvine to Manage P2P Connections*, http://www.dslreports.com/forum/r18323368-Comcast-is-using-Sandvine-to-manage-P2P-Connections.

[2] *BitTorrent Developers Introduce Comcast Busting Encryption*, http://torrentfreak.com/bittorrent-devs-introduce-comcast-busting-encryption-080215/.

[3] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet Service Providers Fear Peer-assisted Content Distribution?" in *Proc. of IMC'05*, October 2005.

[4] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," in *Proc. of ICDCS'06*, July 2006.

[5] O. Saleh and M. Hefeeda, "Modeling and Caching of Peer-to-Peer Traffic," in *Proc. of ICNP'06*, November 2006.

[6] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," in *Proc. of ACM SIGCOMM*, August 2008.

[7] S. Le Blond, A. Legout, W. Dabbous, and I. Mediterranee, "Pushing BitTorrent Locality to the Limit," *Arxiv preprint arXiv:0812.0581v2*, 2009.

[8] K. Cechlárová and T. Fleiner, "On a Generalization of the Stable Roommates Problem," *ACM Trans. Algorithms*, vol. 1, no. 1, pp. 143–156, 2005.

[9] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*. Springer, 2004.

[10] W. Huang, C. Wu, and F. C. Lau, "The Performance and Locality Tradeoff in BitTorrent-like P2P File-Sharing Systems," http://i.cs.hku.hk/∼cwu/papers/tradeoff.pdf, The University of Hong Kong, Tech. Rep., September 2009.

[11] D. R. Choffnes and F. E. Bustamante, "Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems," in *Proc. of ACM SIGCOMM*, August 2008.

[12] F. Mathieu, "Self-stabilization in Preference-based Systems," *Springer Journal of P2P Networking and Applications*, vol. 1, pp. 104–121, 2008.

[13] F. Mathieu, G. Postelnicu, and J. Reynier, "The Stable Configuration in Acyclic Preference-based Systems," in *Proc. of IEEE INFOCOM 2009*, May 2009.