This paper was presented as part of the Workshop on Network Science for Communication Networks (NetSciCom)

Optimizing Topology in Bit Torrent Based Networks

Joydeep Chandra†, Sascha Delitzscher*, Niloy Ganguly† Ashish Jhunjhunwala†, Tyll Krueger*, Naveen Sharma†
*CITEC & Faculty of Physics, Bielefeld University, Germany.
†Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur, India.

***Abstract*—In this paper, we discuss the importance of the network connectivities of the peers in Bit Torrent based systems in determining the download performance of the peers. In this context, assuming that the fraction of the peers of each bandwidth are known, we derive optimal connectivities of the peers that help to improve the average latency of the peers. We represent the topology of a Bit Torrent based system as a weighted graph, where the average edge weight of the graph directly relates to the download latency of the peers. We formulate the average edge weight of the whole system as a linear function of the fraction of the edges that connect peers of different bandwidth and derive the topology that maximizes the average edge weight of the network. Simulation results based on the Bit Torrent protocol validates the fact that in the optimal topology, peers have 13% better download latency as compared to topologies formed in the normal Bit Torrent based systems. Further the obtained topology also improves the fairness of the system as compared to normal Bit Torrent significantly.**

***Index Terms*—Bit Torrent, Static Network, Active Network, Network Performance.**

I. Introduction

The popularity of Bit Torrent as a file sharing protocol has grown immensely in the last few years, thus gaining huge research interest in the scientific community. A major research objective in the case of Bit Torrent systems is to improve the download performance of the peers by addressing several key issues like incentive mechanisms, piece and peer selection mechanisms, and fairness issues.

Another important aspect that determines the download performance of the peers is their own bandwidth as well as the bandwidth of their neighbors. The neighbors of a peer are randomly selected by an entity called *tracker*, which the peers contact while joining the network. However at any particular time, links to only a subset of these neighbors remain active (these links are formed based on a set of rules elaborated in section II), as pieces are transfered through these links. We refer to these neighbors, the links to which are active at a particular time, as the *active neighbors* and the topology formed by these set of active neighbors as the *active topology*. To distinguish from the active neighbors, we refer to all the neighbors (inclusive of both active as well as non-active) as the *static neighbors* and the topology formed by these static neighbors as the *static topology* (ref. figure 1). Hence in this context, selecting suitable static neighbors based on the peers' bandwidths can be an effective technique to improve the download performance of the peers. However, researches have been mainly directed towards improving peer associations in the active topology through various means like developing better incentive mechanisms, piece and peer selection strategies etc.

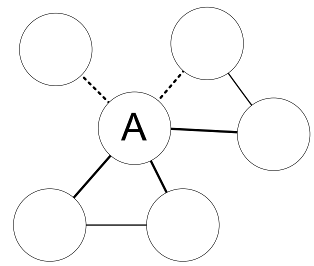


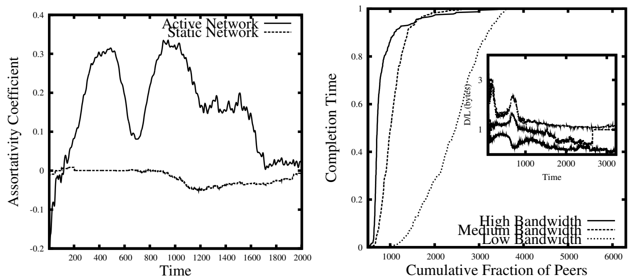
Fig. 1. The static and active neighbors of peer A in a Bit Torrent system. The set of active neighbors (indicated by the solid lines) is a subset of the set of static neighbors.

Improving the static topology has not been considered previously; the main reason being that previous experimental results [1] indicated that the download performance of the peers in a Bit Torrent system is dependent on the active topology which, researchers concluded, is independent of the underlying static topology that is actually formed. Our simulation of the assortativity coefficient of the peers, based on peer bandwidth, in the active and static topology (shown in figure 2(a)) also indicates that while the static topology is fairly random, the active topology is largely correlated and hence clusters of similar bandwidth peers will appear. However beyond this apparent independence if one looks in the entire spectrum of possible static networks, the observations are contrary. A major contribution of this paper is to report the observation that the nature of active topology ***do*** correlate with the static topology when its assortativity[1] is pushed (both in negative and positive direction) beyond a point. We experimentally validate this statement that we discuss next.

A. Validation of the Importance of Static Topology

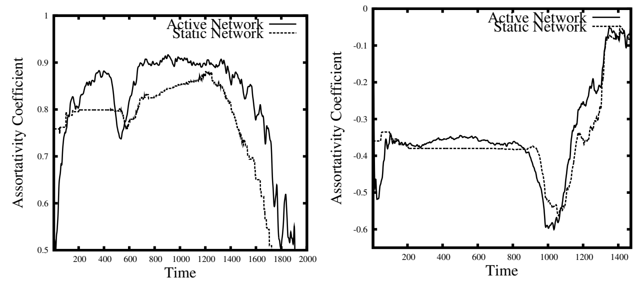
We simulated the assortativity coefficient r [2] of the nodes, based on their bandwidths, for the static and calculated the coefficient of the active network with time, the results of which are shown in figure 3. The figures indicate a huge dependence of the active network on the static network topology. When the static network is assortative, peers tend to exchange more number of pieces with its similar bandwidth neighbors. This indicates that the active network is also assortative. Similarly, when the static network is anti-assortative, peers exchange more number of pieces with dissimilar bandwidth neighbors. When the active network is assortative, the high bandwidth

[1]Assortativity of the peers is defined as the probability of mixing of peers of similar types, like here peers of similar bandwidth



(a) Bit Torrent Assortativity (b) Distribution Download Completion Time

Fig. 2. Figure 2(a) shows the assortativity of the active and static topologies in normal Bit Torrent protocol. Figure 2(b) shows the cumulative distribution of the download completion time of the high, medium and low bandwidth peers in normal Bit Torrent protocol. Figure in inset shows the ratio of bytes downloaded and uploaded over the time by the high, medium and low bandwidth peers.



(a) Assortative Static Network (b) Anti-Assortative Static Network

Fig. 3. The correlation of the assortativity of the active network and static network for assortative [figure 3(a)] and anti-assortative [figure 3(b)] static topology.

peers benefit more as compared to the low bandwidth peers in terms of download latency; the reverse situation occurs in case of anti-assortative networks, where the download latency of the low bandwidth peers improves much as compared to the high bandwidth ones. This dependence gives us the hope that optimizing the static graph will yield improvement in the average download performance of the peers. An optimized static/active graph essentially means that the flow of information, hence the download of pieces, is maximized. We next summarize our objectives.

B. Objectives

The static network of a Bit Torrent system is represented using a weighted graph, where the nodes of the graph represent the peers and the edges represent the links of the static network. The weights of these edges are determined by the bandwidth categories of the peers that are connected through these edges. Since the edge weights represent the volume of information flow possible through the links, the edge weight of a link is thus a representative of the download performance achieved by the corresponding peers of the link. The edge weight can be determined by the nature of the nodes (peers) joining it. For example, a high (low) bandwidth peer connected with a high (low) bandwidth peer results in a high (low) edge weight, while a high connected to a low bandwidth peer will yield an edge weight somewhere between high and low. Hence the first step of constructing optimum topology is to choose the categories of edges in such a fashion so that *the average edge weight, hence information flow* within the network is maximized. The next step would be actually to build up the static topology satisfying or largely conforming to the above constraints.

Rest of the paper is organized as follows. We next provide a brief overview of the Bit Torrent protocol performance. In section III we attempt to derive optimal topologies for a given set of bandwidth categories and fraction of nodes in each category. The simulation results are presented in section IV. Finally, we draw conclusions in section V.

II. Bit Torrent Overview

We provide here a brief description of the Bit Torrent protocol [3].

Peers willing to obtain a file (say F) initially download a torrent file containing the meta-info of file F. The torrent file contains the address of the tracker as well as the information about the file pieces. The torrent file is opened using a Bit Torrent client software, which connects to the tracker that sends a list of around 50 remote peers, selected randomly from the existing peer set. The peers connect to these remote peers thus forming the static topology, which is largely regular. However, peers can generally download simultaneously from a subset of around 40 peers but can simultaneously upload to a more smaller subset of peers ($\sim$ 5) in its neighborhood. This subset of active links forms the active topology and changes after every time slot of 10 seconds (ref. figure 1). In contrast to the static topology, the active topology changes much frequently over time [4].

The file F is broken into smaller pieces which peers exchange among themselves. Peer selection for uploading pieces is done using a CHOKING/UNCHOKING mechanism [5]. Every peer sends an INTERESTED message to its neighbors if it has some missing piece to offer. After an interval of every 10 seconds, every peer selects four neighbor peers preferentially from whom it has recently obtained pieces with the highest bandwidth rate (the tit-for-tat principle). The selected neighbors are then said to be unchoked by the peer, which means that the peer will upload requested pieces to them if they are interested. Rest of the neighbors are said to be choked. After every 30 seconds the peer selects a random neighbor (which is not already unchoked) that has sent an INTERESTED message and unchoke it. This process, called optimistic unchoking is primarily aimed towards helping newly arriving peers that does not have any piece to exchange.

Bit Torrent Simulator: To simulate the Bit Torrent protocol and to test the proposed optimizations, we have developed a discrete event simulator that follows the actual Bit Torrent official protocol [3], including the newly introduced modified seeder choking algorithm [1]. The bandwidth categories of the peers in the simulator can be tuned to any values; however in this paper we use 2 (high and low) or 3 (high, medium and low) bandwidth categories for the simulations. Further the arrival and departure of the peers, that generates the churn in the network has been modeled according to the recent empirical studies made in [4].

Simulation results of the normal Bit Torrent protocol with equal proportions of high, medium and low bandwidth peers indicate that the download performance is heavily biased towards the high bandwidth peers. The cumulative distribution of the total download time of the peers for low and high bandwidth in a normal Bit Torrent system (figure 2(b)), indicates a huge difference in the download latency of the high and low bandwidth peers (nearly 6 times), although the bandwidth of the high bandwidth peers is nearly 3 times higher than the low bandwidth ones and the ratio of the number of bytes downloaded and uploaded is comparable for the high and low bandwidth peers (inset figure 2(b)). Thus the Bit Torrent system is not fair with respect to the low bandwidth peers. This observation also adds to an extra motivation, whereby one of the objectives which can be set is to ensure fairness. Although we are driving for better performance, fairness of the performance also needs to be tested.

We next formalize the topology optimization problem and discuss solutions for the same.

III. Topology Optimization

We next formalize the topology optimization problem and derive optimal topologies for certain special cases.

A. Formalizing the Problem

We model the static network of a Bit Torrent system as a *regular graph* $G(V, E)$, where the set of vertices V represent the peers in the network and E, the set of edges connect them. We assume that the bandwidth of each peer belongs to any one of the categories $x_1, x_2, \ldots, x_n$, where $x_1 < x_2 < \ldots < x_n$. Further we assume that the fraction of peers of each bandwidth category is known and is given as $\rho_1, \rho_2, \ldots, \rho_n$. Let q_{ij} denote the fraction of edges that connects nodes of bandwidth category x_i with that of x_j.

We assume that for an edge, when the two peers corresponding to the edge belong to the same bandwidth category, say x_i, the weight of the edge is assumed to be x_i. However, if one of the nodes is of bandwidth x_i and the other x_j, where $x_i < x_j$, then the weight of the edge is assumed to be x_j with probability $p_{ij}^{(x_j)}$ and x_i with probability $p_{ij}^{(x_i)} = 1 - p_{ij}^{(x_j)}$ respectively, depending upon whether the peer with bandwidth x_j transfers piece to peer with bandwidth x_i or vice versa. We later derive expressions for $p_{ij}^{(x_j)}$.

If $E(\gamma)$ represents the mean edge weight of the edges of the graph, then $E(\gamma)$, which is our objective function, can be represented as

$$E(\gamma) = \sum x_i q_{ii} + \sum_i \sum_{j:j>i} \left[x_j p_{ij}^{(x_j)} q_{ij} + x_i (1 - p_{ij}^{(x_j)}) q_{ij} \right] \quad (1)$$

We can eliminate the term $\sum x_i q_{ii}$ from the above expression of $E(\gamma)$ by establishing a relation between the fraction of the nodes in a category and its corresponding link fraction, which we state next using a theorem, the proof of which is avoided due to want of space.

Theorem 1: Suppose in a network of peers of equal degrees (say d), where each peer has bandwidth $x_i \in \{x_1, x_2, \ldots, x_n\}$ and the fraction of edges connecting peers with bandwidth x_i and x_j denoted as q_{ij} $(1 \leq i, j \leq n)$ is known, then the fraction of peers of bandwidth i can be represented as,

$$\rho_i = q_{ii} + \frac{1}{2} \sum_{j<i} q_{ij} + \frac{1}{2} \sum_{j>i} q_{ij}. \quad (2)$$

From equation 2 we find that

$$\sum_{i=1}^{n} x_i \rho_i = \sum_{i=1}^{n} x_i q_{ii} + \frac{1}{2} \sum_{i=1}^{n} x_i \sum_{j=1}^{i-1} q_{ij} + \frac{1}{2} \sum_{i=1}^{n} x_i \sum_{j=i+1}^{n} q_{ij}$$

$$\Rightarrow \sum_{i=1}^{n} x_i q_{ii} = E(x) - \frac{1}{2} \sum_i \sum_{j:j>i} x_j q_{ij} - \frac{1}{2} \sum_i \sum_{j:j>i} x_i q_{ij}, \quad (3)$$

where $E(x) = \sum_{i=1}^{n} x_i \rho_i$ is the average bandwidth of the peers in the network. From equations 1 and 3, we derive the objective function as

$$E(\gamma) = E(x) + \sum_i \sum_{j:j>i} \left[\Delta_{ji} \left(p_{ij}^{(x_j)} - \frac{1}{2} \right) q_{ij} \right], \quad (4)$$

where $\Delta_{ji} = x_j - x_i$ for $i < j$. Thus our objective is to find optimum values of q_{ij} that maximizes $E(\gamma)$ and follows the constraint in equation 2. We next attempt to derive expressions for p_{ij}.

B. Deriving $p_{ij}^{(x_j)}$ and $p_{ij}^{(x_i)}$

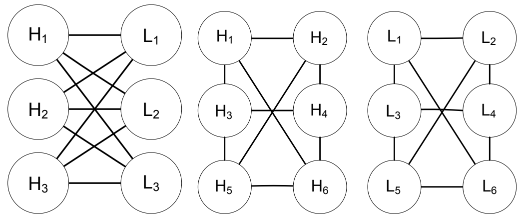
To determine the value of $p_{ij}^{(x_j)}$, we assume that the link connecting peers with bandwidth x_i and x_j respectively is active under exactly one of the following conditions

1) A piece is being transferred in the direction $x_j \to x_i$ via regular unchoke mechanism, the probability of which is assumed to be ϕ_{ji}.
2) A piece is being transferred in the direction $x_i \to x_j$ via regular unchoke mechanism. Since the peers follow a *tit-for-tat* mechanism, then according to the previous case, if the probability that a piece is transferred in direction $x_j \to x_i$ is ϕ_{ji}, the probability that a piece will be transferred in the opposite direction, i.e. $x_i \to x_j$ is $\phi_{ij} = \frac{x_j}{x_i} \phi_{ji}$.
3) A piece is being transferred in the direction $x_i \to x_j$ via optimistic unchoke mechanism, the probability of which is suppose ξ_{ij}.
4) A piece is being transferred in the direction $x_j \to x_i$ via optimistic unchoke mechanism, the probability of which is ξ_{ji}. Since the peers for optimistic unchoking are selected randomly, we can assume that $\xi_{ij} = \xi_{ji} = \xi$.

Thus assuming that exactly one of the above four events must occur when the link is active, we have

$$\phi_{ji} + \xi + \frac{x_j}{x_i} \phi_{ji} + \xi = 1$$

$$\Rightarrow \phi_{ji} = \frac{(1 - 2\xi) x_i}{x_i + x_j} \quad (5)$$



(a) Case: $p^{(x_h)} > \frac{1}{2}$ (b) Case: $p^{(x_h)} < \frac{1}{2}$

Fig. 4. The connectivities of high and low bandwidth nodes, that maximizes the edge weight in a regular graphs, for two cases $p^{(x_h)} > \frac{1}{2}$ (figure 4(a)) and $p^{(x_h)} < \frac{1}{2}$ (figure 4(b)). For $p^{(x_h)} > \frac{1}{2}$, the graph forms a bipartite graph, where no peers of high or low bandwidth are directly connected to each other, on the other hand when $p^{(x_h)} < \frac{1}{2}$, the high and low bandwidth peers gets isolated from each other to maximize the edge weight.

Thus $p_{ij}^{(x_j)} = \phi_{ji} + \xi = \frac{(1-2\xi)x_i}{x_i+x_j} + \xi$.

Finding the values of q_{ij} from the expression in equation 4, for any values of n, is difficult. However for most practical cases, the bandwidth categories of the peers are restricted to 2 or 3; we next attempt to find solutions for networks with two bandwidth categories, high and low.

C. The Case of 2 Bandwidth Levels

For the case $n = 2$, we consider two bandwidths, high and low, denoted as x_h and x_l respectively and let ρ_h and ρ_l denote the fraction of high and low bandwidth peers. Similar to $p_{ij}^{(x_j)}$ in the n bandwidth category case, for $n = 2$, we denote the probability of a transfer from high to low bandwidth peer as $p^{(x_h)}$, i.e. a link connecting high and low bandwidth peer has weight x_h with probability p and x_l with probability $1 - p^{(x_h)}$. Further, let q_{hh}, q_{ll} and q_{hl} denote the fraction of edges connecting high-high, low-low and high-low bandwidth peers respectively. Then the average edge weight (similar to equation 4) can be derived as

$$E(\gamma) = E(x) + (x_h - x_l)\left(p^{(x_h)} - \frac{1}{2}\right) q_{hl} \tag{6}$$

subject to conditions

$$\rho_h = q_{hh} + \frac{q_{hl}}{2} \text{ and } \rho_l = q_{ll} + \frac{q_{hl}}{2} \tag{7}$$

Our objective is to maximize $E(\gamma)$. We find from equation 6 that since the term $E(x)$ is a constant and $x_h - x_l > 0$, hence for $p^{(x_h)} > \frac{1}{2}$, the term $(x_h - x_l)\left(p^{(x_h)} - \frac{1}{2}\right) q_{hl}$ becomes greater than zero and thus needs to be maximized for maximizing $E(\gamma)$. Similarly, when $p^{(x_h)} < \frac{1}{2}$, the corresponding term becomes negative and hence needs to be minimized. As can be observed, when $p^{(x_h)} > \frac{1}{2}$, $E(\gamma)$ is maximum when $q_{hl} = 1$ and when $p^{(x_h)} < \frac{1}{2}$, the corresponding value of $q_{hl} = 0$. These values imply that when $p^{(x_h)} < \frac{1}{2}$, to maximize the edge weight the high and low bandwidth peers should have no connections among themselves, thus indicating a total clustering of the peers based on their bandwidth. Correspondingly, for $p^{(x_h)} > \frac{1}{2}$, the edge weight is maximized when $q_{hl} = 1$, i.e. there is a maximum mixing between the peers of different bandwidth. Thus we find that there exists a critical value of $p^{(x_h)}$ for which the nature of the topology changes completely when we intend to maximize the average edge weight (ref. figure 4). However, for practical purpose, to maintain the scalability of the network the network needs to be connected. Hence to maintain a given connectivity of the network and yet maximize the edge weight, we need to do the following.

When $p^{(x_h)} < \frac{1}{2}$, we need to minimize the connections between the different bandwidth nodes. Thus the solution is to partition the graph following a min-cut algorithm, where the partition sizes are known apriori. However the graph partitioning problem where the fraction of nodes in each partition is given is a known NP-Complete problem [6], and efficient heuristics for the solution of the same exists [7].

Similarly when $p^{(x_h)} > \frac{1}{2}$, the problem of partitioning the graph to maximize the average edge weight requires to partition the graph into high and low bandwidth peer sets such that the total edge weight between the two sets is maximum. Thus this problem evolves as a max-cut problem which is also a known NP-complete problem [6] and similar heuristics for approximating the partition exists. For our simulations, we use the Kernighan-Lin heuristic [7] to generate the min-cut partition of the network.

D. The Case of n Bandwidth Levels

On attempting to derive optimal values of q_{ij} for any generic value of n, we find from equation 4 that the average edge weight, $E(\gamma)$ depends on all the values of $p_{ij}^{(x_j)}$ for all possible values of i, j.

From the expression of $p_{ij}^{(x_j)}$ obtained from equation 5, we find that for all $x_i < x_j$, we have $p_{ij}^{(x_j)} > \frac{1}{2}$ if

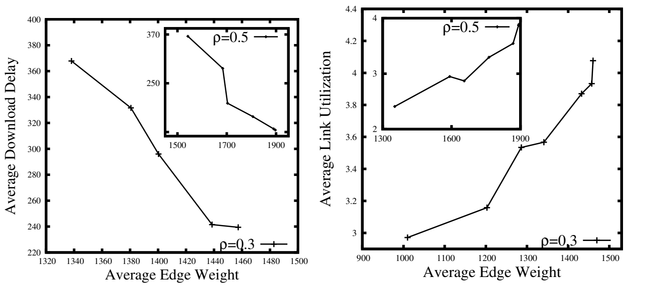
$$\frac{(1-2\xi)x_i}{x_i + x_j} + \xi > \frac{1}{2}$$
$$\Rightarrow \xi > \frac{1}{2}. \tag{8}$$

This essentially means that optimistic unchoking is performed always. This is an impossible situation, hence $p_{ij}^{(x_j)}$ is less than $\frac{1}{2}$ for any practical situation. Thus in this case, we always need to minimize $E(\gamma)$ (refer equation 4, where $\left(p_{ij}^{x_j} - \frac{1}{2}\right)$ becomes negative for all values of i, j) and hence the optimal topology can be obtained by iteratively applying the min-cut algorithm for every pair of bandwidth categories.

In the next section, we show with the help of simulations that using our model and partitioning the network accordingly yields substantial improvement in the link utilization and download latency as compared to the random network generated using the tracker.

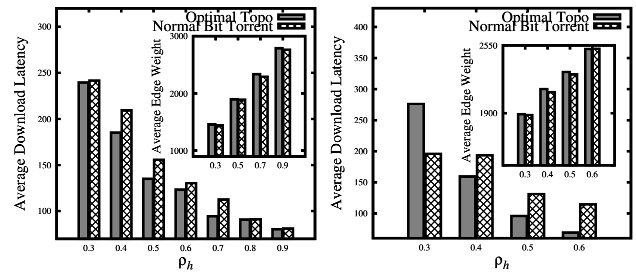
IV. Simulation Results

We discuss the simulation results obtained to validate the proposed models. The Bit Torrent simulator developed by us was briefly described in section II. In the simulator the static graph gets evolved over time. But the min-cut algorithm is essentially producing a complete static graph. Developing an evolving algorithm which encompasses all the dynamics of



(a) Download Delay vs. Edge Weight (b) Link Utilization vs. Edge Weight

Fig. 5. Figure 5(a) shows the relation of the average edge weight and the average download latency of the peers for $\rho = 0.3$ and the figure in inset shows the same for $\rho = 0.5$. Figure 5(b) shows the correlation between average edge weight and link utilization of the peers.



(a) Average Download Latency Comparison with ρ_h (2 Bandwidth Levels) (b) Average Download Latency Comparison with ρ_h (3 Bandwidth Levels)

Fig. 6. Figure 6(a) shows the variation of average download time for the optimal topology and the normal Bit Torrent protocol with ρ_h and the inset figure shows the corresponding average weight of the peers. Figure 6(b) shows the same for 3 bandwidth levels.

node churn is a non-trivial problem and not discussed here. However, for simulation purpose, we assume that this static graph is a representation of the stable underlay arising out of evolution and churn.

We simulated various network parameters like the average edge weight of the network and the average download latency of the peers in the system for various values of q_{ij}, where $i \neq j$, and also for various values of ρ_i, for both 2 and 3 bandwidth categories. The downloading file was broken into 300 pieces; we measured the average edge weight and the average download latency of a peer after it has downloaded 30 pieces. This artificially ensures the stability of the underlay which is required to fully understand the impact of static network. We initially show the correlation between the average download latency of the system and average edge weight of the peers and then show the variation of the average edge weight for various parameters stated above.

A. Download Performance vs. Average Edge Weight

We simulated the average download latency of the system and the average edge weight to establish the correlation between the two. The results, shown in figure 5(a), indicate that the average download latency of the peers decreases with increasing average edge weight. Hence improving the average edge weight will improve the download latency of the peers. Further, we consider another parameter, the link utilization, U, which we define as the average number of links of a peer that are active for download per time slot. A higher value of U indicates more number of parallel download occurring per time slot. We also establish a correlation of U and the average edge weight of the system. Simulation results, shown in figure 5(b) indicate a strong positive correlation between these parameters. The results validate our principal proposition that improving the edge weights in static topologies directly affects the performance of the system.

B. Performance of Min-cut Topology

In this section we compare the download performance of the peers in topologies obtained using the min-cut algorithm and the ones formed in normal Bit Torrent systems.

We calculated the average download latency of the peers for the min-cut topologies and normal Bit Torrent systems for various values of ρ_h, for 2 and 3-bandwidth levels. Simulation results for the 2-level case with nearly 2000 peers (shown in figure 6(a)) show that with increasing values of ρ_h, the average download latency of the peers for the min-cut topology steadily improves as compared to the normal Bit Torrent topology. However, for very low values of ρ_h, the average download latency is slightly better in case of normal Bit Torrent topology. This is because of the extreme high proportion of low bandwidth peers that have a slightly better download latency than in the min-cut topology. Figure 6(b) shows very similar results for 3 bandwidth levels, high, medium and low. In the 3 bandwidth level case, for ease of representation, we fixed the fraction of medium bandwidth peers to $\rho_m = 0.3$ and the high and low bandwidth peers have been varied accordingly. The average download latency of the peers in the min-cut topology is nearly 13% lower as compared to the normal Bit Torrent topology for the 2-bandwidth level case, when $\rho_h = 0.5$ and nearly 18% lower in case of 3-bandwidth levels, when $\rho_h = 0.4$, thus indicating a huge improvement in download latency of the peers.

The min-cut algorithm forms a topology with the minimum possible value of cut-point q_{hl} for given values of ρ_h; in the next section, we observe the average download latency and the average edge weight of the peers for the entire spectrum of the cut-points.

C. Effect of Cut-points

We observe the average download latency of the peers and the average edge weight for various value of q_{hl}. Simulation results for the 2-bandwidth level case, shown in figure 7(a), for $\rho_h = 0.3$, 0.5 and 0.7, reveal that for all the three values of ρ_h, the average download latency of the system increases with increasing values of q_{hl}. For each of these three cases we observed the value of $p^{(x_h)}$ to be lesser than $\frac{1}{2}$, and hence as discussed in section III-C that when $p^{(x_h)} < \frac{1}{2}$, the edge weight decreases with increasing q_{hl} (ref. inset figure 7(a)).

The figure shows that the average download latency of the system increases very slowly with q_{hl}, when the value of q_{hl}

is small ($q_{hl} < 0.3$, not shown in figure) and then increases at a faster rate with further increase in q_{hl}. The average edge weight of the system also decreases accordingly with increasing q_{hl}. Thus our observation reveals that topologies similar to the min-cut topology have very similar download performance.

D. Effect on Fairness

In this section we discuss the fairness of the system in case of min-cut topology and compare it with the normal Bit Torrent topology. We also observe the change in fairness of the system with q_{hl}. To measure the fairness of a system, we introduce a term called fairness index, which we define as follows:

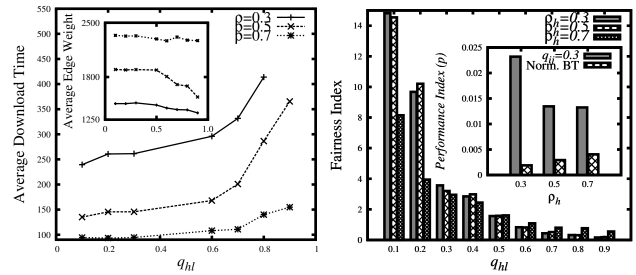
Definition 1: If d_l and d_h represents the average download latency of the low and high bandwidth peers respectively, the fairness index f of the system in a 2-bandwidth level case is defined as the ratio, $f = \frac{d_l}{d_h}$. The system is considered to be fair if f is nearer to an optimal fairness value $f_o = \frac{b_h}{b_l}$, where b_h and b_l are the download bandwidth of the high and low bandwidth peers respectively.

Figure 7(b) shows the variation of the fairness, f, of the system in a 2-bandwidth level case with q_{hl}, when the download latency of the high and low bandwidth peers are 3000 Kbps and 800 Kbps respectively, for 3 values of ρ_h (0.3, 0.5, 0.7). Thus the system will be considered as fair if the value of f is nearer to the optimal fairness value of $\frac{3000}{800} = 3.75$. As can be seen, f is far from the optimal value for very low values of q_{hl} as the download latency of the high bandwidth peers are much lower as compared to the low bandwidth ones. The value of f is also very low at higher values of q_{hl} indicating that low bandwidth peers gaining undue advantage over high bandwidth peers.

If we traverse the curve up from low values of q_{hl}, with slight increase the fairness improves very fast and reaches the optimal point. Hence, although the min-cut topology ($q_{hl} = 0.1$) obtained using the min-cut partitioning algorithm is not optimal in terms of fairness; however the fairness is maximum at $q_{hl} = 0.3$, the configuration of which is very similar to the min-cut value of $q_{hl} = 0.1$. Moreover, as seen from figure 7(a), the average download latency of the peers at $q_{hl} = 0.3$ is only slightly greater than in case of $q_{hl} = 0.1$, thus indicating that the topology, which is nearly optimal in terms of the average download latency of the peers as well as the fairness index is very similar to the min-cut topology that we have derived. We state such a topology as a near-optimal topology.

Hence in an effort to measure the effectiveness of a topology we introduce a measure called the *performance index* p, combining fairness and download performance. The performance index of a system is defined as follows:

Definition 2: If f represents the fairness index of a system with the optimal fairness represented as f_o, and d_a represents the average download latency, then the performance index, p is represented as $\frac{1}{(|f-f_o|)\cdot d_a}$, when $f \neq f_o$, and is represented as $\frac{1}{d_a}$ when $f = f_o$. Note, $|f - f_o|$ indicates how far one is away from the optimal point.



(a) Average Download Time vs. q_{ij} (2 Bandwidth levels)
(b) Fairness Index and Performance Index

Fig. 7. Figure 7(a) shows the variation of average download time of the peers with $q_{ij}(i \neq j)$ in a 2-level case for $\rho_h = 0.3, 0.5$ and 0.7. The figure in inset shows the corresponding average weight of the peers. Figure 7(b) shows the variation of fairness index f with q_{hl} for the same values of ρ. The figure in inset compares the performance index p for the near-optimum topology at q_{hl}=0.3 and the normal Bit Torrent topology for $\rho_h = 0.3, 0.5$ and 0.7

Thus a higher value of p indicates better performance of the system. Figure 7(b) (inset) compares the performance index of the normal Bit Torrent topology and the near-optimum topology with $q_{hl} = 0.3$; as the figure indicates the near-optimum topology has much better fairness (nearly 12 times better for $\rho_h = 0.3$) as compared to the Bit Torrent system.

V. Conclusion

The principal contribution of this work is realizing that static network can affect the performance of active networks. However, we did not stop at this realization, using analytical and algorithmic techniques we show that there are optimum topologies which can minimize download latency. But beyond performance maximization, there are fairness issues and we show that there are zones where high fairness is achieved without undermining the performance too much. The next work remains in developing a more realistic model so that the properties of the static graph get imbibed dynamically.

References

[1] Arnaud Legout, Nikitas Liogkas, Eddie Kohler, and Lixia Zhang, "Clustering and sharing incentives in bittorrent systems," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, New York, NY, USA, 2007, SIGMETRICS '07, pp. 301–312, ACM.

[2] M. E. J. Newman, "Assortative mixing in networks," *Physical Review Letters*, vol. 89, no. 20, October 2002.

[3] BitTorrent, "http://en.wikipedia.org/wiki/bittorrent," .

[4] Daniel Stutzbach and Reza Rejaie, "Understanding Churn in Peer-to-Peer Networks," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet Measurement*, New York, NY, USA, 2006, pp. 189–202, ACM.

[5] Arnaud Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest first and choke algorithms are enough," in *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*, New York, NY, USA, 2006, pp. 203–216, ACM Press.

[6] D. Johnson M. Garey and L. Stockmeyer, "Some simplified np-complete graph problems," *Theoretical Computer Science*, vol. 1, pp. 237–267, 1976.

[7] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *The Bell system technical journal*, vol. 49, no. 1, pp. 291–307, 1970.