Evolution and Enhancement of BitTorrent Network Topologies

Cameron Dale, Jiangchuan Liu and Joseph Peters
School of Computing Science
Simon Fraser University
Burnaby, British Columbia, Canada
Email: {camerond,jcliu,peters}@cs.sfu.ca

Bo Li
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: bli@cse.ust.hk

***Abstract*—This paper describes an experimental study that closely examines the underlying topologies of multiple complex networks formed in BitTorrent swarms. Our results demonstrate that the networks exhibit fundamental differences during different stages of a swarm, suggesting that the initial stage is not predictive of the overall performance. We also find a power-law degree distribution in the network of peers that are unchoked by others, which indicates the presence of a robust scale-free network. However, unlike previous studies, we find no clear evidence of persistent clustering in any of the networks, precluding the presence of a small-world that is potentially efficient for peer-to-peer downloading. These results suggest an interesting venue for improving BitTorrent's performance. We present a first attempt to introduce clustering into BitTorrent. Our approach is theoretically proven and makes minimal changes to the tracker only. Its effectiveness is verified through a series of simulations and experiments.**

I. Introduction

Among all of the peer-to-peer Internet applications available, BitTorrent [1] has become the most popular for the downloading of large files. It has been reported that half of all current Internet traffic is due to BitTorrent [2]. This popularity can be attributed to the efficiency with which BitTorrent can distribute large files, and its resilience to peer departures, peer failures, and misbehaving peers. Many of these properties have been confirmed through both theoretical and experimental studies. One aspect yet to be fully explored is the topology of the network of peers formed during a download. In particular, the resilience to failing and misbehaving nodes suggests that the network may be *scale-free*, and the efficiency of information distribution suggests that the network may be clustered or even *small-world*. Neither of these properties has been quantitatively measured beyond the early stages of swarms. Since BitTorrent networks are highly dynamic, a clear understanding of the characteristics and evolution of the networks during churn and during their entire lifespans is critical to its performance optimization.

In this paper, we describe experiments that closely examine the underlying topologies of BitTorrent swarms. These experiments capture the intricacies of forming multiple complex networks in BitTorrent, including the formation of four networks in a BitTorrent download: Connection, Interest, Unchoked, and Download. Unlike previous work which was confined to the initial stage, we look at their characteristics and dynamics throughout the entire lifespan of swarms. Our results demonstrate that the networks exhibit fundamental differences over time. This suggests that the initial stage of a BitTorrent swarm is not sufficient to predict the overall performance of the system, and in order to fully examine a BitTorrent swarm, long-term measurements are needed.

We find strong evidence of scale-free characteristics in the network of peers that are unchoked by other peers. However, we find no clear evidence of persistent clustering in any of the networks of peers that we studied, which suggests an interesting venue for improving BitTorrent's performance.

We present a first attempt to introduce clustering into BitTorrent. Our approach makes minimal changes to a BitTorrent tracker by adding peers to a fixed number of groups, called n-cliques. The tracker then returns mostly peers from the same group, with only a small number from other groups. Our theory indicates that this will keep the clustering high within the n-cliques, while the connections to other groups will keep the characteristic path length low. We verify this theoretical prediction by simulation and experiment.

II. Background and Related Work

A. Related Work

Various aspects of the performance of BitTorrent have been investigated through measurement and analytical modeling [3]. Recently, several authors have examined the network topologies formed by BitTorrent. Urvoy-Keller and Michiardi [4] used a simulated BitTorrent overlay to look at the distance of peers from the initial seed and the matrix of peer connections. Their results were based on a homogeneous collection of peers, and were limited to the initial stage of a swarm. Al-Hamra et al. [5] expanded on those results through simulation with some experimental confirmation. They also examined the diameter of the overlay created, and the robustness of the overlay in the presence of churn and attacks. Legout et al. [6] performed an experimental evaluation with around 40 heterogeneous peers, finding interesting evidence of clustering in the network of peer unchokings.

Our results differ from these earlier results in several ways. We have focused on experimental evaluation, which, as a complement to theoretical modeling and simulation, captures the intricacies of forming multiple complex networks in BitTorrent. We use over 400 peers and explore the entire lifespans of the swarms: from initialization to steady state. This enables us to quantitatively evaluate both time-invariant characteristics and those that evolve in different stages.

B. Scale-Free Networks

Many real-world networks have been found to be *scale-free* [7]. In a *scale-free network*, the probability that a node is connected to k other nodes follows a power law distribution $P(k) = k^{-\gamma}$, in which the power γ is usually between 2 and 3 [8]. This results in a large number of nodes with small degrees and a small number of nodes, called *hubs*, with large degrees. The presence of hubs leads to good tolerance of random node failures in scale-free networks [9].

Two mechanisms contribute to the scale-free nature of many real-world networks: the networks evolve over time, and nodes attach preferentially to other nodes based on distinguishing characteristics. Both mechanisms are present in peer-to-peer networks [10], but the presence of scale-free characteristics in BitTorrent swarms has not been previously confirmed.

C. Small-World Networks

Milgram [11] initiated the study of small-world networks while investigating the phenomenon that people are linked by short chains of acquaintances (popularly known as *six degrees of separation*). Small-world networks possess characteristics of both random and regular networks [12]. More formally, the *clustering coefficient* C_i of node i is the fraction of all possible edges between neighbors of i that are present, while the clustering coefficient of a network is the average clustering coefficient of its nodes. This coefficient is small for random graphs and large for regular graphs. A *small-world graph* has a large clustering coefficient like a regular graph, but also has a small *characteristic path length* (average distance between nodes) like a random graph.

Small-world networks are known to be effective for the exchange and dissemination of information, including broadcasting and the spread of viruses [13], [14]. Many existing peer-to-peer systems (e.g., Gnutella [15], Freenet [16], and DHT-based systems [17]) are known to be small-world. It is natural to expect that BitTorrent swarms would exhibit small-world characteristics, particularly since clustering has been observed in early stages of swarms [6]. Unfortunately, we find strong evidence that this is not the case.

III. Definitions and Experiment Design

Since the terminologies and operations of BitTorrent systems have been well-documented in the literature, we refer interested readers to [3]. In this section, we describe the different levels of networks formed by BitTorrent peers, and the design of our experiments.

A. Networks in BitTorrent Swarms

Given the complex relations among peers, BitTorrent actually maintains four networks (or graphs[1]) during a swarm. Previous studies have focused on only one or two of the following networks. In this paper, we will investigate the properties and evolution of all four networks.

Connection Network. This is the network of neighbors that each peer maintains. These neighbors are chosen randomly by the tracker from the list of peers in the system. Each peer makes connections to the peers returned by the tracker and also makes return connections to other peers that connect to it. All neighbor connections are bi-directional, so the Connection Network is undirected.

Interest Network. This network represents the interest that peers have in other peers. Each peer maintains a list of the pieces stored by its neighboring peers. A peer is interested in any neighboring peer that has a piece it does not have. Since interest can be uni-directional, the Interest Network is directed.

Unchoked Network. This network is formed by the incentive mechanism present in BitTorrent. Each uploading peer assigns its limited number of unchoke slots to certain neighboring peers in an effort to maximize the downloads that it receives from them. Since unchoking can be uni-directional, the Unchoked Network is directed.

Download Network. This network is formed by the peers that are downloading from other peers. Since downloading can be uni-directional, the Download Network is directed.

We investigate all four networks in our experiments. We emphasize that the Connection and Unchoked networks are the most important of these four networks. The Connection network forms the neighbor set for all of the peers in the system, and is a superset of the other networks. The Unchoked network is necessary for the uploading and downloading of data from other peers, and so it is very important for the scalability and efficiency of BitTorrent.

B. Experiment Design

Our experimental data was gathered using a modified version of the BitTornado program [18], which is a typical and widely-used BitTorrent client. The program was modified to log the connections to other peers that were made by the client for the four types of networks described in section III-A. Except for this, we have made no modification to the normal operations of BitTorrent. The modified BitTornado client was used to collect data from more than 400 nodes of the PlanetLab research network testbed.

Below, we show the results of a representative experimental configuration from a series of experiments that we conducted. We created a test file consisting of 780 MB of random data (a typical size for a BitTorrent download) and assigned one node to be the original seed. The connection speeds that we used, shown in table I, are typical of those available from Internet Service Providers. The percentage of nodes

[1]In this paper, we use *graphs* to represent *networks*, and we use the two terms interchangeably.

TABLE I
THE DISTRIBUTION OF PEER CHARACTERISTICS USED IN THE EXPERIMENTS.

Nodes	Duration	Upload	Download	Unchoke Slots[a]
45 %	44 hours	6 kB/s	6 kB/s	2
25 %	20 hours	12 kB/s	24 kB/s	3
15 %	12 hours	25 kB/s	75 kB/s	4
10 %	6 hours	50 kB/s	150 kB/s	5
5 %[b]	4 hours	100 kB/s	500 kB/s	6

[a] This is the number of unchoke slots available for uploading (out-degree). Downloading (in-degree) is not limited.
[b] The original seed for the experiment is in this class of peers.

for each connection speed was determined from previous measurements of real BitTorrent swarms [19]. The number of unchoke slots was varied according to recommended values for different connection speeds. We limited the maximum number of connections for each peer to 80, which is the default value in many BitTorrent clients.

To realize a larger total number of peers than is possible with only 400 nodes, the clients were scheduled to join randomly over the first 4 hours of the experiment, download the file to completion, seed for a random period, and then leave and rejoin to restart the download. The average times in the system are shown in table I (except for the original seed, which stayed indefinitely). The experiments were run for over 100 hours, which was enough time for all of the peers to become seeds, leave the system, and rejoin multiple times. After the experiments, the data was synchronized for time by examining the times that pairs of peers logged a bidirectional connection.

In this paper, we will present results based on the above configuration. Similar conclusions can be drawn from our other experiments. To investigate the impact of some key factors, in particular the connectivity and the churn, we conducted additional experiments. In one experiment, we removed the limit on the number of connections that each peer could make. In another experiment, we introduced alternating periods of high and low churn. We found that most of the results are similar, so we will only highlight the differences.

All of the figures in the next section were created by measuring the characteristics of the networks at regular intervals during the experiment. Approximately 360 networks were generated to produce 100 hours of data for each figure. We determined this to be frequent enough to capture all of the details of the evolving networks.

The Connection, Interest, and Unchoked graphs are constructed using actual connections among peers that have a defined start and end time. The Download graph is more difficult to construct because downloading from a peer occurs almost instantaneously for the small piece size. To overcome this difficulty, we constructed it by considering all peers that have downloaded from each other since the last measurement.

It is worth noting that some of the directed Interest, Unchoked, and Download networks are not connected. This is due to the presence of seeds. Seeds have the entire file, so they are not interested in other peers, are not unchoked by other peers, and do not download from other peers. This can

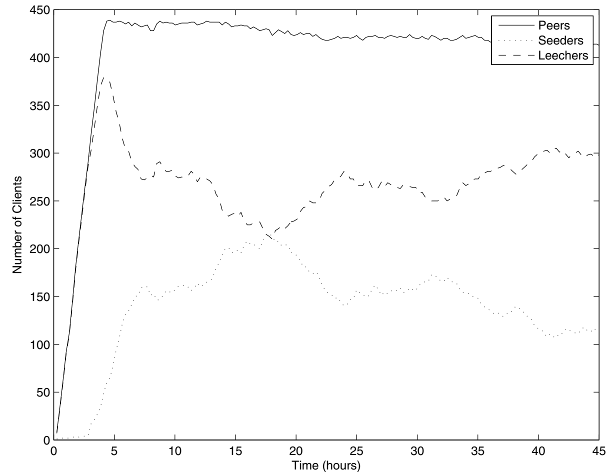

Fig. 1. The population of peers in the system during the first 45 hours of the experiments.

cause problems with calculations that depend on the graph being connected. For these calculations, we used the largest strongly connected component, which is usually equivalent to removing the seeds from the network.

IV. EXPERIMENTAL RESULTS

Figure 1 shows the distribution of seeds and leechers throughout the first 45 hours of the 100 hours of the experiment. The remaining 55 hours are not shown as the system has reached a steady state after which there is little change in the results. The varying numbers of seeders and leechers is due to the progression of the BitTorrent clients. We will further investigate the impact of different amounts of churn in section IV-C.

We can identify roughly three regions in figure 1. The system starts out in an initial stage with a single seed, and then peers join randomly. After the first 4 hours, the total number of peers in the system is approximately 430. Some of the peers are already beginning to be converted into seeds before 4 hours as they joined early with the fastest download rate and so have already completed their downloads. The system then enters a transient stage, from 5 hours to about 25 hours, during which the numbers of seeders and leechers in the system are still varying. After 25 hours, the numbers of seeders and leechers change very slowly and are generally quite steady. From 45 hours to 100 hours (not shown in figure 1), the numbers of seeders and leechers change by less than 10%, probably due only to the randomness in the experiment.

A. Characteristics of Network Topologies

To determine if the node degrees in the network exhibit a power law distribution, we plot the degree of each node against the rank of the node by degree, on a log-log scale. The slope of a linear fit then yields the power law exponent, and an R^2 goodness of fit value is also generated. The only network of the four that exhibited this power law behavior was the Unchoked network, which had an R^2 goodness of fit value

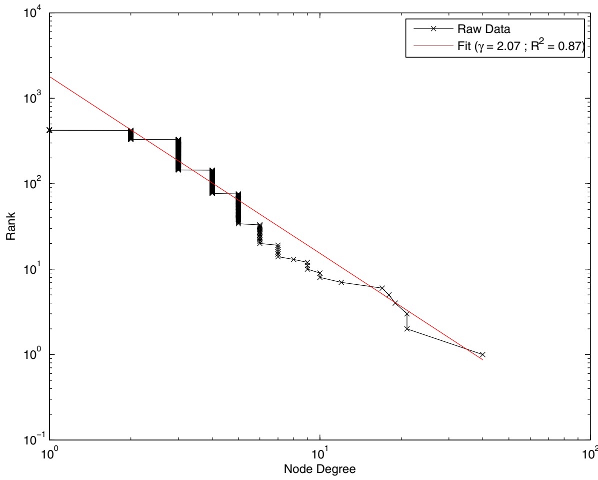

Fig. 2. The node in-degree distribution for the Unchoked network at hour 19 of the experiment, and the resulting fit to it.

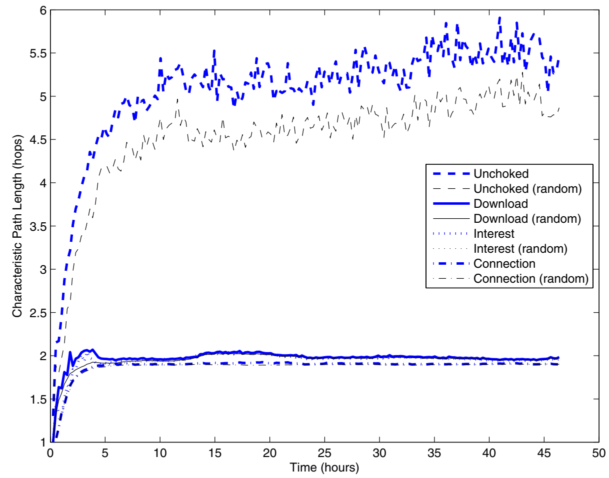

Fig. 4. The characteristic path lengths during the experiments. Also shown are those of a similar-sized random graph.

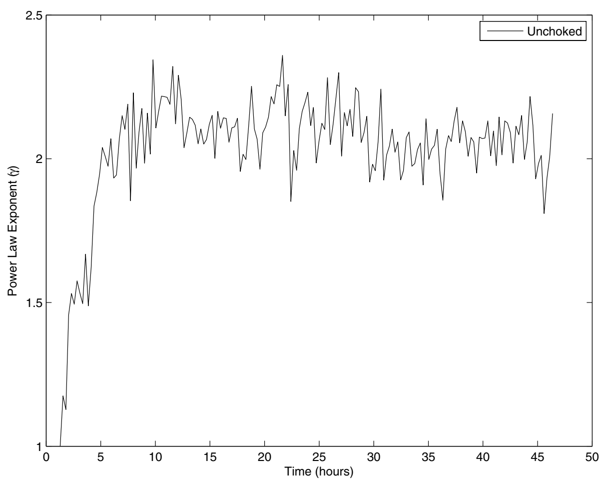

Fig. 3. The exponent found by fitting a power law to the Unchoked graph's node degree during the experiments.

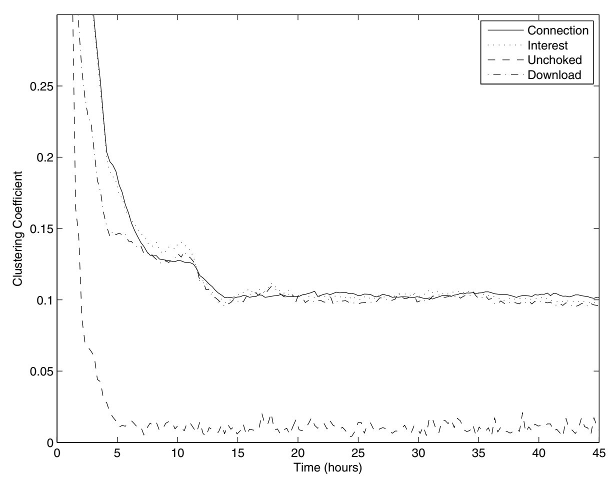

Fig. 5. The clustering coefficients during the experiments.

of approximately 0.9 over most of the experiment (except the initial stage). This is high enough to indicate a good fit, while the other networks had goodness of fit values less than 0.7. A sample node degree distribution and fit is shown in figure 2.

Figure 3 shows the power law exponent found from the fitting of the in-degree of the nodes in the Unchoked network. The power law exponent can be seen to vary quite a lot during the initial stage. However, once all the peers have joined the system the power law exponent quickly reaches its final value, and remains very steady at just over 2 through most of the transient stage and all of the steady state.

Figure 4 shows the characteristic path lengths of the four networks in the experiment. Note that, for the directed Interest, Unchoked, and Download graphs, the path lengths were calculated on the graphs after they were reduced to their largest strongly connected component to avoid the disconnected nature of BitTorrent graphs. The characteristic path length increases rapidly during the initial stage, though all but the Unchoked network slow their increase even before the initial stage is complete. The Unchoked graph reaches its steady state value early in the transient stage, after which none of the networks vary much at all.

The characteristic path lengths for the Connection, Interest, and Download networks are short, due mostly to the density of the graph (430 nodes with an average degree of 65). The Unchoked graph's characteristic path length is larger due to the reduced degree (about 4) of nodes in this graph. Also shown are the characteristic path lengths of a randomly constructed graph with the same number of nodes and edges, and with similar limits on the node degree [20]. The random graph results are almost not visible, as the Connection, Interest, and Download graphs have nearly the same characteristic path lengths as their random graph counterparts. The only exception is the Unchoked graph which is about 10% larger, probably due to the scale-free nature of this graph which causes it to vary slightly from being truly random.

Figure 5 shows the clustering coefficients of the four net-

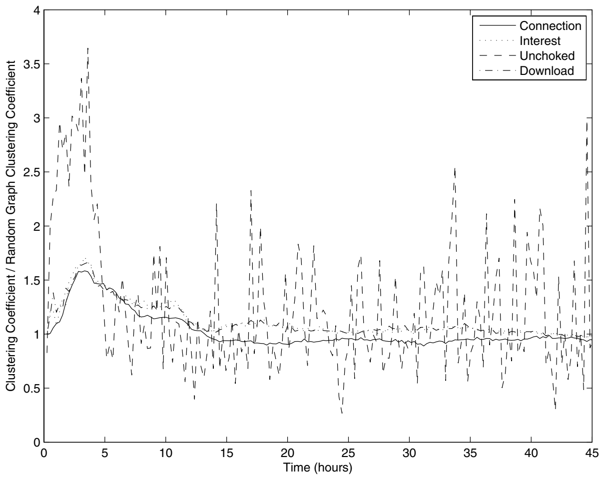

Fig. 6. The clustering coefficients during the experiments compared with the clustering coefficient of a similar random graph.

works in the experiment. Although not shown in the figure, the coefficient starts at 1 (since it is a clique), and then has a sharp decline during the initial stage as the size of the graph increases. Once all the peers have joined the system there is some further decrease in the coefficients of all but the Unchoked graph during the transient stage. Through the end of the transient stage there are some further small oscillations in the Interest and Download graphs, until all settled into a steady state after approximately 25 hours.

Although at first it seems that there is some clustering present in figure 5, especially in the graphs of Connection, Interest, and Download peers, further investigation shows that is not the case. Figure 6 shows the clustering coefficients of the graphs when compared with that of a similar sized random graph (same node and edge restrictions), which is not expected to have any clustering at all. Here we see that there is some clustering during the initial stage, which begins to decrease once all the nodes have joined the system. The Unchoked graph has no clustering through the rest of the experiment, while the clustering of the other graphs reduces more slowly through the transient stage. In the steady state, all graphs have almost no clustering. The increased noise in the comparison of the Unchoked graph with a random graph is due to the randomness of the resulting graph and the relatively tiny clustering coefficient.

B. Connectivity Matrix

To compare with the results from previous papers [4], [5], we present the connectivity matrix of peer connections during the experiment. The connectivity matrix is a scatter plot, where a point at location (i, j) in the plot refers to the fact that peer i is connected to peer j. Peer indexes i are created by sorting peers by their joining time.

Figure 7 shows the connectivity matrix formed after 4 hours at the end of the initial stage when most of the peers have joined the swarm. The fan-out shape from the lower left to upper right corner of the matrix occurs due to the early peers

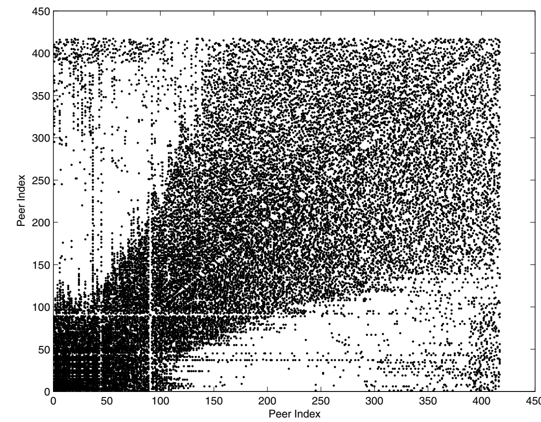

Fig. 7. The connectivity matrix at hour 4.

filling their 80 connection limit and refusing later connections, and is very similar to previous results [5]. There is however some additional connectivity between early and late peers, which is due to some of the early peers being the fastest downloaders and having already completed their downloads. Once they become seeds they disconnect from other seeds in the system, thus freeing up connection slots for later peers.

Although figure 7 does match well with the previous results at the early stages of the experiment, we now proceed further into the experiment to see how the connectivity matrix evolves. Figure 8 shows the connectivity matrix 4 hours further into the experiment in the middle of the transient stage, at which point some peers have left and new peers have joined the system. The matrix is now much more random, with many early peers having lost connections to leaving peers and so connecting to many late peers, though the fan-out is still visible in the lower left corner. Figure 9 goes further to 16 hours into the experiment, where the connectivity matrix becomes an almost completely random scattering of points, and the fan-out in the lower left is almost not visible. The connectivity matrix has now reached a steady state, as shown by the similarity of figures 9 and 10.

The experiment we ran with no limit on the number of connections a peer could make gives almost identical results to that in figures 1 through 6, but differs from the connectivity matrix shown in figure 7. In this experiment, the connectivity matrix throughout the entire experiment was completely random (similar to figure 10), as the fan-out shape in the other matrices is due only to the limit on the number of connections.

C. Impact of Churn

To further evaluate the impact of churn, we varied the amount of churn at certain points in the system by grouping some of the peer departures and arrivals together[2]. Figure 11 shows the resulting population of seeders and leechers in the system. Although the total number of peers does not change, the increased churn occurs when the number of seeds

[2]The previous experiment also had churn, but the amount of churn was steady throughout the experiment.

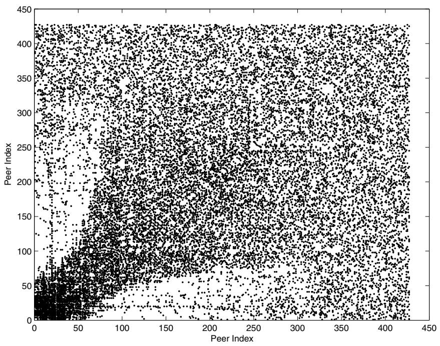

Fig. 8. The connectivity matrix at hour 8.

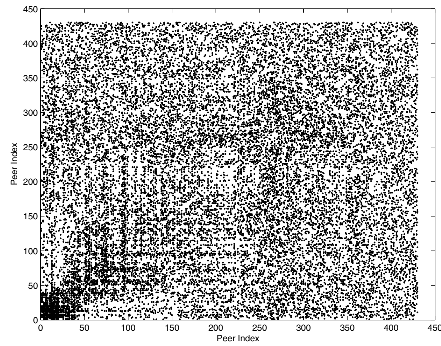

Fig. 9. The connectivity matrix at hour 16.

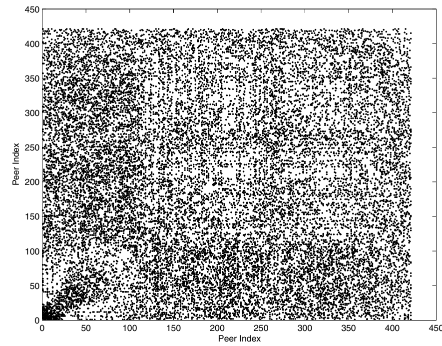

Fig. 10. The connectivity matrix at hour 32.

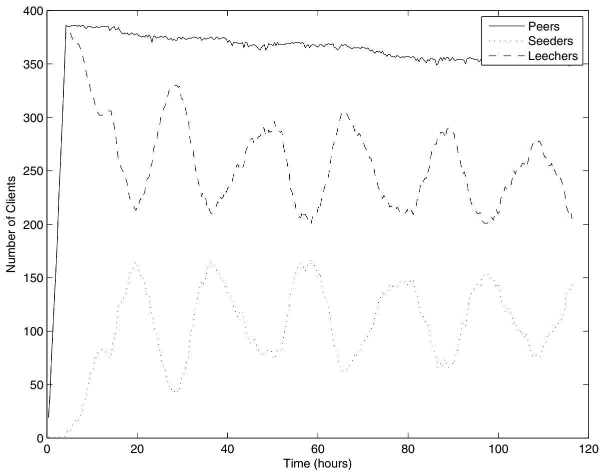

Fig. 11. The population of peers during the experiment with varying churn.

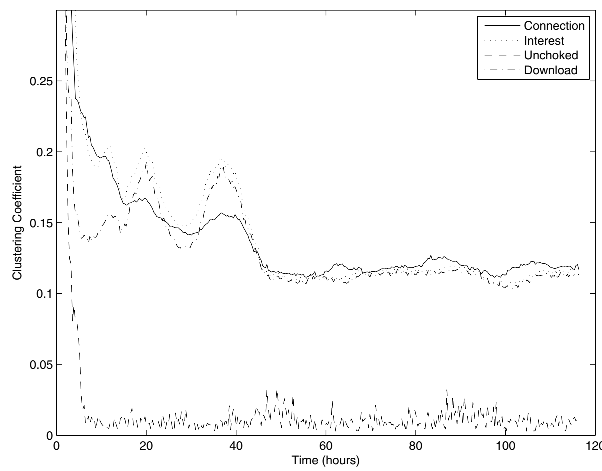

Fig. 12. The clustering coefficient during the experiment with varying churn.

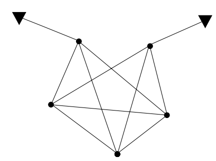

Fig. 13. An example of one of the 5-cliques from a cycle, the arrows indicate connections to neighboring cliques.

decreases rapidly, for example from 20 to 25 hours, 40 to 50 hours, and 60 to 70 hours. Since the number of peers is limited, the periods between these increased churn periods exhibit a state of decreased churn as compared with the previous experiment. We find that this varying churn had almost no effect on the power law exponent or the characteristic path length, which are identical to figures 3 and 4.

Figure 12 shows the clustering coefficient for the four graphs in the experiment with varying churn. During the initial stage, it is very similar to figure 5, decreasing rapidly as the peers enter the system. However, after the initial stage (i.e., after 4 hours), the effect of the varying churn can be clearly seen on the Connection, Interest, and Download networks, causing their clustering coefficients to oscillate. Interestingly, the clustering coefficient increases during the periods of light churn and decreases during the periods of heavy churn. Although the varying churn continues throughout the experiment, the oscillations in the clustering coefficients of these graphs are greatly reduced after 50 hours.

V. Making BitTorrent Small-World: A Theory

It is known that small-world networks are efficient for spreading information [13], [14]. A previous study [6] has also conjectured that BitTorrent's efficiency partly comes from the clustering of peers. It is thus interesting to see whether BitTorrent networks can be made small-world. In this section, we present a theoretical attempt at increasing the small-world characteristics of BitTorrent networks. In the next section we show how this can be done in practice.

A. Maximum Clustering Coefficient

Since most existing small-world graphs are sparse, and the BitTorrent networks we are considering are quite dense, it is not clear that creating a small-world network in BitTorrent is even possible. Therefore we start our investigation by determining the maximum possible clustering coefficient. We will focus on the Connection Network, the superset of the other three networks.

To create a small-world network containing peers with a known maximum degree, we first attempt to maximize the clustering coefficient of a regular graph of these peers. Our instinct is to create a series of cliques, since they have a perfect clustering coefficient of 1, each with size equal to the maximum node degree. A single edge can then be removed from each clique (to maintain regularity), and the endpoints of the removed edge are connected to neighboring cliques. This results in a cycle of k identical n*-cliques*, where n is the maximum node degree, and k is given by $k = N/n$ (assuming for now that k is an integer). Figure 13 shows an example of one of the n-cliques from the cycle for $n = 5$.

As the clustering coefficient of the graph is an average over all nodes, and each n-clique is identical, it will be sufficient to calculate the clustering coefficient of a single n-clique. Each n-clique contains two types of nodes: $n-2$ interior nodes (i-nodes) connected only to neighbors in the same n-clique, and 2 exterior nodes (e-nodes) that have a single connection to another n-clique. Since the clustering coefficient of a node is a measure of how many triangles include the node, we will only look at how many triangles are lost by removing the single edge to connect to neighboring cliques. The i-nodes lose only a single triangle when the edge is removed, so their clustering coefficient is

$$C_i = \frac{\frac{(n-1)(n-2)}{2} - 1}{\frac{(n-1)(n-2)}{2}} = 1 - \frac{2}{(n-1)(n-2)} \qquad (1)$$

The e-nodes lose a triangle for each node that was connected to the missing edge, of which there are $n-2$. So their clustering coefficient is

$$C_e = \frac{\frac{(n-1)(n-2)}{2} - (n-2)}{\frac{(n-1)(n-2)}{2}} = 1 - \frac{2}{(n-1)} \qquad (2)$$

Averaging over the n-clique gives the clustering coefficient of the entire graph:

$$CC(G) = \frac{(n-2) * C_i + 2 * C_e}{n} = 1 - \frac{6}{n(n-1)} \qquad (3)$$

This result is independent of the total size of the graph, and so of the number of n-cliques used. It also approaches 1 as the size of the n-cliques increases. For BitTorrent, which has a default maximum node degree of 80, the clustering coefficient is very close to 1 (0.9986 for $n = 80$).

B. Caveats: Expanding Diameter

The clustering coefficient resulting from our construction is large, but the diameter and characteristic path length of the graph are also large. It takes 3 hops to get through a single clique, and the worst case is having to go half way around the cycle of k cliques; hence, the maximum diameter is given by approximately $\frac{3k}{2}$.

We can estimate the characteristic path length by considering only the distances of i-nodes from other i-nodes (since there are many more of them than there are e-nodes). For each i-node there are $n-1$ nodes at distance 1, $2n$ nodes at distance 3 (one n-clique away), $2n$ nodes at distance 6, etc.... The sum of the distances for all possible i-nodes is then

$$(n-1) + 6n \sum_{j=1}^{\frac{k-1}{2}} j = n - 1 + 3n \frac{k-1}{2} \left(\frac{k-1}{2} + 1 \right) \qquad (4)$$

The characteristic path length of the graph can be calculated by using an approximation for large values of n and dividing by the number of nodes:

$$CPL(G) \approx \frac{n + \frac{3}{4} n(k-1)(k+1)}{nk} = \frac{1 + \frac{3}{4}(k^2 - 1)}{k} \qquad (5)$$

For a BitTorrent graph with $n = 80$ and 400 nodes ($k = 5$), the characteristic path length will be 3.8, which is almost twice that of a similarly sized random graph.

C. Forming the Small-World

To solve the dilemma, we modify the regular n-clique graph by randomly removing a small number of n-clique edges and adding back new randomly created ones. This is inspired by previous work on the construction of small-world graphs from regular graphs [12]. In our construction, the new random edges are restricted to be links between pairs of n-cliques.

Figure 14 shows the result of adding varying amounts of randomness to the regular n-clique graphs for a few sizes of graphs. We again use a clique size of 80, which is the default maximum node degree of BitTorrent graphs. The randomness varies from 0 (no edges replaced, completely regular) to 1 (all edges replaced, making the graph almost completely random). We observe that the graphs become small-world when approximately 1 to 3% of the edges are randomly replaced. At this point the clustering coefficient still maintains 90% of it's original value, while the characteristic path length has dropped very near to that of a completely random graph.

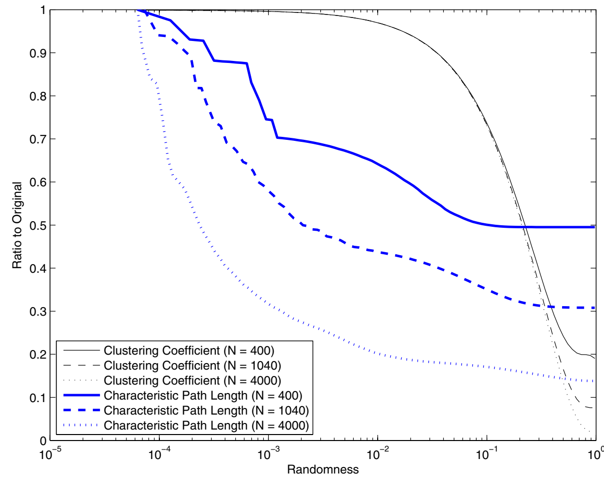

Fig. 14. The change in the clustering coefficients and characteristic path lengths of the n-clique graphs with varying amounts of randomness added. The variables are shown as ratios to the values when no edges are replaced. The original clustering coefficients were all 0.999, while the original characteristic path lengths were 3.64, 6.25 and 15.8 for the 400, 1040 and 4000 node graphs respectively.

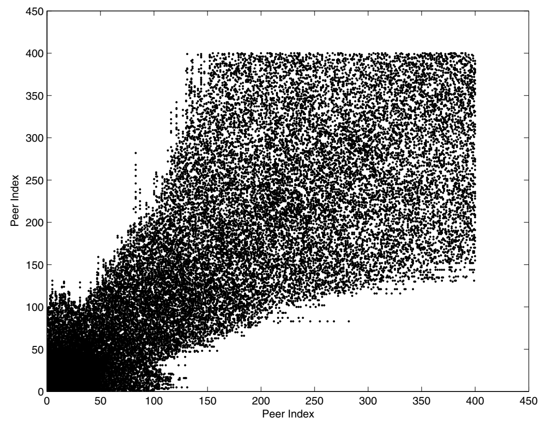

Fig. 15. The simulated connectivity matrix after 400 peers have joined the swarm.

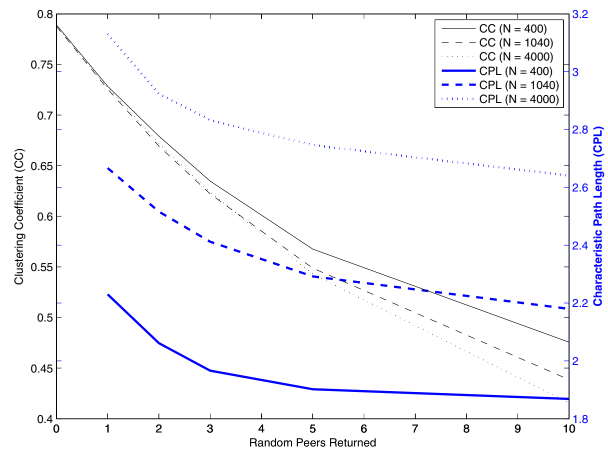

Fig. 16. The simulated small-world tracker results with different values for random-peers, for three sizes of graphs (clique-size is kept constant at 80).

VI. Small-World Tracker: A Simple Solution

To realize the theoretical results in practical BitTorrent software, we have implemented a simple modification to the BitTorrent tracker. We believe this is the best, and possibly only, type of change that can be made to BitTorrent's functionality. The tracker is easily modifiable by any data distributor, whereas the vast diversity of BitTorrent clients that would need to be changed for a client modification make that type of change quite difficult.

A. Tracker Modification

The modified tracker assigns a number to each newly arrived peer indicating the n-clique to which it belongs. If all the n-cliques are full (or there are no n-cliques), the peer will receive a new n-clique number one larger than the largest so far. Otherwise the peer receives the number of the largest currently unfilled n-clique. When choosing a list of other peers to return to the current peer, the n-clique number of the peer will be considered. The tracker will first choose a small fixed number of peers randomly from n-cliques that do not contain the peer. This small number of peers will be the control on the randomness referred to in section V. The remaining peers (the majority) will be randomly chosen from those in the same n-clique as the current peer.

There are two new configuration parameters for this tracker. The first, which we call *random-peers*, is the fixed number of peers that will be randomly chosen from other n-cliques for each request. The second, which we call *clique-size*, is the maximum size of n-clique that the new tracker will allow. Once an n-clique reaches this size, the next peer to join will create a new empty n-clique.

B. Simulation Results

We first present some results from a customized discrete event-driven simulator of BitTorrent. We have verified the correctness of the simulator with a standard tracker implementation that returns a random set of peers in each request, by comparing the results with our previous experimental results. There are 400 peers arriving over a period of 4 hours, followed by 2 hours during which the peers make further requests from the tracker to satisfy their minimum peer requirements. Figure 15 shows the connectivity matrix after 6 hours, which can be compared with Figure 7 from our previous experiment. Other than the connections in figure 7 due to peers becoming seeds (which we did not simulate), the results are almost identical.

For the modified tracker, Figure 16 shows the resulting clustering coefficients and characteristic path lengths for three sizes of swarms with random-peers values from 0 to 10. The characteristic path length for 0 random-peers does not appear, as it is infinite due to the disconnected nature of the graph.

Compared with the previous experimental results, there are gains in the clustering coefficient of the graph, but they are not as dramatic as the theory in section V suggests (i.e. they are not almost 1 for zero randomness). This is mostly due to the limits on the number of connections that a peer can initiate in BitTorrent, which prevents the peers from forming a complete n-clique. Since we are trying to only modify the tracker, we cannot adjust the number of connections at which a BitTorrent client stops initiating more. However, we can reduce the clique-size tracker configuration parameter so that more connections are made within the clique. We verified this by reducing the clique-size to 40, which increased the clustering coefficient to 1 for 0 random-peers.

C. Experimental Results

We further confirmed the effectiveness of the modification experimentally. The experiment was run as described previously in section III-B, with the only difference being the modified small-world tracker. The modified tracker used a random-peers value of 1, and a clique-size value of 80. The population of peers in the system during the experiment was identical to the previous experiment shown in figure 1.

Figure 17 shows the characteristic path length for the four networks. As expected, there is an increase in the characteristic path length due to the regular construction of the graph imposed by the tracker, as compared with the previous results in figure 4. The increase is on the order of 10 to 20%.

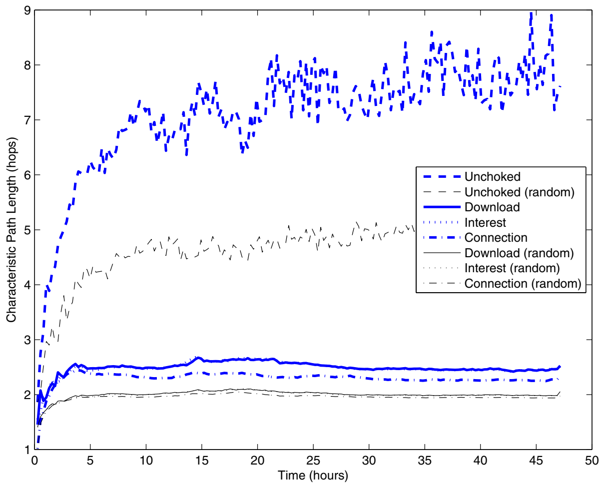

Fig. 17. The characteristic path lengths during the small-world tracker experiment, also shown are those of a similar-sized random graph.

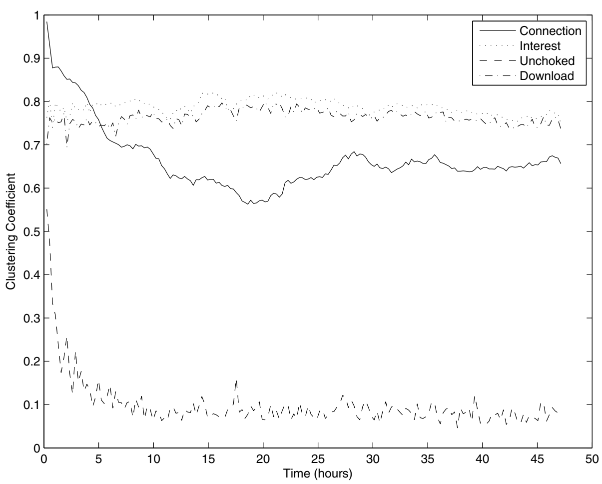

Fig. 18. The clustering coefficients during the small-world tracker experiment.

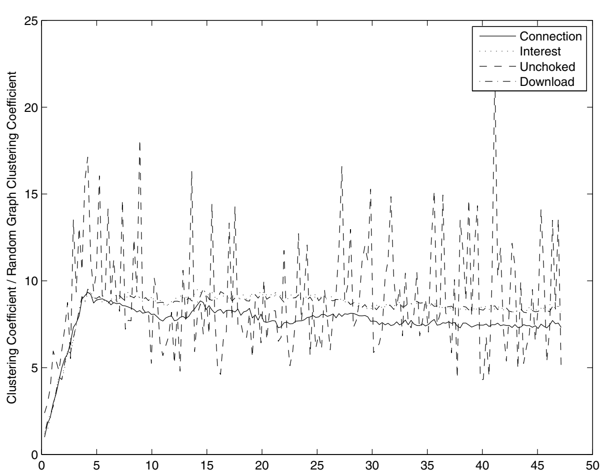

Fig. 19. The clustering coefficients during the small-world tracker experiment, compared with the clustering coefficient of a similar random graph.

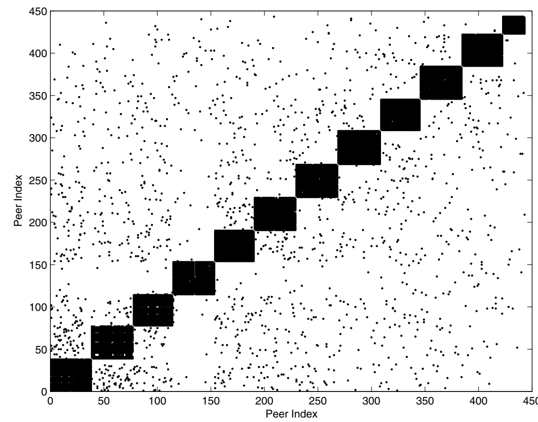

Fig. 20. The connectivity matrix at hour 4 of the small-world tracker experiment.

Figure 18 shows the clustering coefficient for the four networks in the experiment. Compared with the previous results (shown in figure 5), the clustering coefficient has increased by a factor of 7 or 8. To determine if there really is clustering present, we compare the clustering coefficients of the graphs with that of a similarly sized random graph in figure 19. Here we see that there is definite clustering throughout the experiment in all four of the networks. These results are a dramatic increase over the amount of clustering present in the previous experiments, shown in figures 5 and 6.

Figure 20 shows the connectivity matrix at hour 4 of the experiment, after all the original peers have joined the swarm. This connectivity matrix is constructed differently than previously: rather than ordering the peers by their arrival times to get their peer indices, we instead sort them by the clique to which they belong.[3] This shows the cliques that we have constructed more clearly, as can be seen by the tight boxes of peers in the figure. The matrix is now quite different than the previous results shown in figure 7. The formation of the n-cliques is very clear, with some random connections to other n-cliques also present. The tight boxes of peers in figure 20 remain throughout the experiment, though the presence of seeds leads to some holes due to their dropping connections to other seeds.

VII. Conclusions and Discussion

This paper presented an experimental study of the characteristics and evolution of a comprehensive set of BitTorrent network topologies, as well as a possible enhancement to BitTorrent to make it a small-world.

Our most important finding was that the initial stage of a BitTorrent system is not predictive of the overall performance of the system. In order to fully examine a BitTorrent swarm, long-term experiments that examine the changes due to later stages of the swarm are needed. This was clearly demonstrated by the difference that we found between the steady state matrix

[3] We also attempted this with the previous results and got an almost completely random connectivity matrix similar to figure 10.

connectivity and the connectivity that was previously reported to occur early in the lifetime of a BitTorrent swarm [5]. The difference is almost certainly due to the evolution of the swarm over time. As peers become seeds they break connections with other seeds, and when peers leave they free up connection slots for new peers. Neither of these aspects were previously considered. Furthermore, peers use the tracker to form new random connections to peers. We found that, on average, a peer will get a new random peer list from the tracker at least once during its time in the swarm. This behavior was not considered at all in previous studies.

Our experimental results showed that the network of peers that unchoke each other is scale-free, exemplified by a power law distribution of node degrees. We found that the Unchoked graph has a power law exponent of approximately 2 in all of the experiments, independent of time and changes to the amount of churn or the maximum number of neighbors. This scale-free nature has also been found in other peer-to-peer systems [15]. Since scale-free graphs are resistant to random attacks (in this case, churn), the resulting graphs are more robust than graphs without this property[4]. This robustness was observed previously in BitTorrent networks [5], but no explanation for its existence was given. We believe that the robustness (to churn) of the scale-free Unchoked graph is responsible for the efficient use of upload bandwidth previously observed in BitTorrent systems [3].

We found that the characteristic path lengths and clustering coefficients of the Connection, Interest, and Download graphs are very similar to random graphs and there is no evidence of small-world characteristics. Other than the scale-free degree distribution, we found that the Unchoked graph is also nearly random after a short initial period. We showed quantitatively that there is a small amount of clustering in the Unchoked network during the initial stage, confirming the previous qualitative evidence [6]. However, after this short period we found no evidence of clustering in any of our experiments, which precludes the presence of a small-world network.

Small-world characteristics are desirable for efficient information distribution [13], [14], and previous studies have also conjectured [6] that BitTorrent's efficiency partly comes from the clustering of peers with similar bandwidth. We believe that this can be an interesting venue for improving BitTorrent's performance. We therefore presented a theoretical framework for making BitTorrent small-world, together with a practical implementation. Our implementation makes minimal changes to trackers only, and the preliminary results show that the simple modification created a dramatic increase in the amount of clustering, at the expense of a slightly increased diameter. Although the tracker controls only one of the four networks in BitTorrent, i.e. the Connection network, we showed that our modification also introduces small-world characteristics to all other networks, including the important Unchoked network.

In addition, we did find some interesting increases in the clustering coefficient during periods of light or no churn in the current BitTorrent. We believe that it is this churn, in combination with the random list of peers returned by the tracker (designed to create a random graph), that is preventing clustering from occurring. This is another area of potential improvement for BitTorrent that could quite easily yield increases in the efficiency of file distribution.

[4]The vulnerability of scale-free graphs to targeted attacks is not an issue in BitTorrent because the tracker prevents the graph from becoming disconnected.

Acknowledgment

This research was supported in part by a Canada NSERC Discovery Grant and an NSERC Strategic Project Grant. Bo Li's work was supported in part by grants from RGC under the contracts 616406 and 616207, by a grant from NSFC/RGC under the contract N_HKUST603/07, by a grant from HKUST under the contract RPC06/07.EG27, by a grant from National Basic Research Program of China (973 Program) under the contract 2006CB303000, and by a grant from Nokia Research under the contract NOKIA001-BL.06/07.

References

[1] B. Cohen. (2003, May) Incentives build robustness in BitTorrent. [Online]. Available: http://bitconjurer.org/BitTorrent/bittorrentecon.pdf
[2] (2004) CacheLogic. [Online]. Available: http://www.cachelogic.com
[3] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. SIGCOMM '04*, Portland, Oregon, USA, August 30–September 3, 2004.
[4] G. Urvoy-Keller and P. Michiardi, "Impact of inner parameters and overlay structure on the performance of bittorrent." in *INFOCOM*, 2006.
[5] A. Al-Hamra, A. Legout, and C. Barakat, "Understanding the properties of the bittorrent overlay," INRIA, Tech. Rep., 2007. [Online]. Available: http://arxiv.org/pdf/0707.1820
[6] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and sharing incentives in bittorrent systems." in *SIGMETRICS*, 2007, pp. 301–312.
[7] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[8] A. Barabási and E. Bonabeau, "Scale-Free Networks," *Scientific American Magazine*, vol. 288, no. 5, pp. 60–69, 2003.
[9] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, pp. 378–382, July 2000.
[10] S. Willmott, J. Pujol, and U. Cortes, "On Exploiting Agent Technology in the design of Peer-to-Peer Applications," *Lecture Notes in Computer Science*, vol. 3601, p. 98, 2005.
[11] S. Milgram, "The small world problem," *Psychology Today*, vol. 2, no. 1, pp. 60–67, 1967.
[12] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 409–10, 1998.
[13] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Phys. Rev. Lett.*, vol. 87, no. 19, p. 198701, Oct 2001.
[14] F. Comellas, J. Ozon, and J. Peters, "Deterministic small-world communication networks," *Information Processing Letters*, vol. 76, no. 1-2, pp. 83–90, 2000.
[15] G. Liu, M. Hu, B. Fang, and H. Zhang, "Measurement and Modeling of Large-Scale Peer-to-Peer Storage System," *Lecture Notes in Computer Science*, pp. 270–277, 2004.
[16] H. Zhang, A. Goel, and R. Govindan, "Using the small-world model to improve Freenet performance," *Computer Networks*, vol. 46, no. 4, pp. 555–574, 2004.
[17] K. Hui, J. Lui, and D. Yau, "Small world overlay P2P networks," *Proc. Twelfth IEEE International Workshop on Quality of Service*, pp. 201–210, 2004.
[18] (2007) The BitTornado website. [Online]. Available: http://www.bittornado.com/
[19] C. Dale and J. Liu, "A measurement study of piece population in BitTorrent," in *GlobeCom*, Washington, DC, November 26–30 2007.
[20] B. Bollobás, *Random Graphs*, 2nd ed. Cambridge University Press, 2001.