This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE P2P 2012 proceedings

Disconnecting to Connect: understanding Optimistic Disconnection in BitTorrent

Matheus B. Lehmann, Lucas F. Müller, Rodolfo S. Antunes, Marinho P. Barcellos
Institute of Informatics – Federal University of Rio Grande do Sul – Porto Alegre, RS, Brazil
Email: {mblehmann, lfmuller, rsantunes, marinho}@inf.ufrgs.br

***Abstract*—The significance of BitTorrent motivated various studies focused on modeling and evaluating the protocol characteristics and its current implementations in the Internet. So far, however, no work has investigated Optimistic Disconnect (OD), an ad hoc connection management mechanism widely employed in BitTorrent agents. OD allows a peer to search for "better" neighbors in the swarm by disconnecting peers from the current neighborhood and connecting to others. This paper presents an extensive experimental evaluation to study and quantify potential benefits of OD, such as average download time and topology robustness. We evaluate different scenarios and the impact of factors such as average peer reachability and arrival pattern. We found that OD generally improves the overall performance of the swarm (in up to 30% in the evaluated scenarios), while improving the robustness of its topology.**

I. Introduction

BitTorrent has been one of the most, if not the most, popular P2P application, as indicated by a recent report [1]. It groups users (peers) interested in the same content in *swarms*, which are unstructured network overlays. Each peer establish connections and directly interacts with a subset of the swarm, which becomes the peer neighborhood.

Unstructured networks are characterized by the lack of determinism in the connections established between peers [2]. That is, each peer autonomously decides which peers it will establish connections with. Such method creates self-organizing networks that do not require complex management operations, even in the presence of a large population of transient peers. However, the resulting topology of such networks is arbitrary.

It is a known fact that the overall performance of P2P applications is directly influenced by the topology of the overlay network formed among peers [3]. Hence, unstructured P2P applications try to organize peers in topologies that maximize desirable properties, such as robustness and performance. In BitTorrent each peer selects its possible neighbors from a random list of peers which is received from a tracker or other source of peer addresses. This way, the swarm topology is expected to follow a random graph model [4], which is robust to handle highly transient networks [5].

However, previous work on BitTorrent swarm topologies diverge on their properties. On the one hand, studies conclude that BitTorrent connectivity graph is not random [6], [7], neither small world [7]. On the other, studies claim that the resulting swarm topology is, indeed, a random graph because they are formed by a combination of churn and random connections [8].

This work was supported by CNPq through grant 485587/2011-4.

In practice, BitTorrent user agents implement extensions that may influence the performance and topology obtained with the default protocol. *Optimistic Disconnect* (OD) [9], a mechanism implemented in widely used BitTorrent user agents, extends the default BitTorrent connection management, using a "disconnect to connect" strategy. More precisely, it allows a peer to disconnect some of the least useful neighbors in order to try to connect with "better" ones. Although this process may significantly change the swarm behavior and its topology, no previous work has investigated these aspects. Further, OD is an *ad hoc* mechanism: the reasoning behind OD is unknown and, to the best of our knowledge, there has been no scientific study to provide evidence about its benefits and how they are achieved.

This paper presents the first detailed investigation about the impact of OD in fundamental questions regarding swarm performance and topology characteristics. To accomplish this, we executed an extensive set of experiments in PlanetLab using different degrees of peer reachability and arrival times. The main contribution of this paper lies on identifying how the use of OD impacts on the performance of BitTorrent swarms and their topological characteristics. Our results show that OD increases the swarm overall performance in up to 30% and also helps organizing peers in more robust topologies.

The remaining of the paper is organized as follows. Section II discusses related work on BitTorrent, focusing on papers which study topological properties. Section III presents an overview of BitTorrent connection management mechanisms and how OD modifies the default behavior of BitTorrent. The scenarios, parameters and metrics employed in our experiments are presented in Section IV. The summary of our results and most important insights are discussed in Section V. Finally, Section VI presents final considerations and directions for future work.

II. Related Work

BitTorrent topologies have been studied in light of various aspects and methodologies. An evaluation of the performance of peers considering different neighborhood size and percentage of outgoing connections is presented in [10]. The simulation results show that these two parameters impact the resulting swarm topology, directly affecting peers performance. This work was extended by [6], which included a study of the main properties of the swarm topology. It shows that BitTorrent networks typically have small diameter and that this improves content dissemination. It also observes that topologies are not random because peer arrival distribution influences connections. Finally, authors conclude that swarm

978-1-4673-2862-3/12/$31.00 ©2012 IEEE

topologies are robust and resilient to churn but may become very vulnerable if less than half of the peers in the swarm are unreachable (due to NAT, for instance).

On the wild swarm measurements are presented in [7], which attempts to define the topological structure and properties of real swarms. Its results indicate that the topology graph is neither random nor small world. They also conclude that network properties, such as clustering coefficient and diameter, are constant throughout the swarm lifetime, despite the high transience of connections among peers.

Live experiments are used in [8] to study BitTorrent network topologies. The results of the experiments, contradicting previous work, demonstrate that the peer connection graph is in fact random.

Some studies attempt to model the topology of BitTorrent swarms. Authors in [11] use complex networks models to describe a BitTorrent swarm. They conclude that BitTorrent networks become less clustered as the number of peers or the proportion of seeders increase. Similarly, a complex network model is used in [12] to analyze degree distribution, clustering coefficient and average path length of the swarm.

Previous studies seldom consider the impact that mechanisms implemented in user agents may cause on the swarm topology and efficiency. These mechanisms include widely used ones, such as OD. To the best of our knowledge, the only work that mentions OD is [9], but it only sketches how the mechanism works and its possible impacts. Our work is the first detailed study about the effect of OD in BitTorrent, considering performance and characteristics of swarm topologies under different scenarios.

III. Connection Management

In this section we first review the default connection management operation in BitTorrent. Then we discuss two classes of work that modify the default behavior: the first one focuses on improving the selection of new neighbors; the second changes how the neighborhood is managed after connections are established. Finally, we detail and discuss Optimistic Disconnect.

A. Default Mechanism

To join a swarm, a peer first retrieves a metadata file (.torrent) which describes the desired content. This file includes, among other information, the IP address and port number of the tracker, which is a central entity that keeps the IP addresses of all currently active peers in the swarm. The peer contacts the tracker, which will respond by sending a random list of peers participating in the swarm, typically with up to 200 addresses. After this interaction with the tracker, the peer is registered in the swarm and attempts to initiate new connections to other peers. It tries to establish new connections until a maximum neighborhood size is reached (80 by default).

A peer confirms that it continues in the swarm by regularly contacting the tracker. Otherwise, it will be eventually removed from the active peer list (the time between last contact and removal may vary from around 30 minutes to many hours [13]). The peer may also contact the tracker when its neighborhood size drops below a threshold, normally 20 connections. In both situations the tracker will send a new random peer list as response, allowing the requester to establish new connections if necessary.

The mechanism responsible for building the neighborhood in BitTorrent is very simple. It will accept new incoming connections as well as attempt to establish new ones whenever possible (i.e. neighborhood has not reached its maximum size). From the list obtained from the tracker, the peer will choose a random address to attempt a connection under two restrictions: (*i*) only one connection is allowed to exist between two peers and (*ii*) two seeders (i.e. peers that have a complete copy of the content and thus only upload data) should not be connected. Once a connection between two peers is established, it is terminated only if one of them leaves the swarm or if both become seeders.

B. Connection Management Extensions

There are two types of proposed extensions to the default connection management, depending on when they act. The first type affects the selection of peers to connect to, while the second changes the neighborhood management after connections have been established.

Extensions related to connection establishment use a set of metrics to define the best peers to connect. One common approach is to explore locality in order to connect peers according to ISP proximity. Some proposals suggest that ISPs should cooperate with peers by providing information services that allows a better neighborhood selection. BGP routing information is employed by authors of [14] to develop such service. Other proposals suggest that peers should autonomously probe network services to select peers that seem to be closer due to lower connection latency. Authors of [15] present a modification to user agents that probes to CDN to locate closer neighbors. In [16] authors suggest that network coordinates can be employed to estimate the locality of peers, but ultimately conclude that results are no better than employing simple network RTT data to guide selection. Finally, some proposals, as the one presented in [17], suggest that trackers should be modified to guide peers in the selection of neighbors in order to optimize locality.

Neighborhood management, on its turn, is a set of mechanisms that improve the quality of established connections. Two mechanisms were proposed and present similarities despite their different goals. Authors in [18] propose a countermeasure against eclipse attacks. The algorithm periodically evaluates the neighborhood searching for inactive peers to disconnect and establishing connections to new ones. Preemption Strategy [19] was developed to build more robust topologies. In the latter mechanism, peers randomly select a neighbor that has its connection terminated in order to allow new incoming connections from other peers.

C. Optimistic Disconnect

Optimistic Disconnect (OD) is a mechanism implemented in Vuze[1] that modifies the default BitTorrent neighborhood management without the addition of complex operations. The algorithm has linear complexity in function of the neighborhood size, which is typically small (at most 80 peers).

[1]Available in www.vuze.com

OD focuses on improving the usefulness of neighbors to which a peer is connected. It periodically evaluates and ranks all neighbors according to their contribution potential and disconnects the least useful one. The opening of a slot in the neighborhood allows the peer to establish a potentially more useful connection.

In practice, OD only executes after a peer reaches its maximum neighborhood size. This condition exists to avoid unnecessarily disconnecting neighbors. In addition, a predetermined minimum time must pass before a connection becomes eligible for disconnection. This condition helps avoiding churn and allows a neighbor to demonstrate its utility. We base the following discussion on the OD implementation of Vuze (4.6.0.5 of March 2011[2]). Algorithm 1 details the steps taken by OD in order to rank a peer neighborhood and select one neighbor for disconnection. The algorithm is executed every 30 seconds.

Algorithm 1 *Optimistic Disconnect* pseudo algorithm

1: **for all** $v_i \in V$ **do**
2: $\quad t_i \leftarrow l_i$
3: $\quad$ **if not** $seeding$ **then**
4: $\quad\quad$ **if not** $interested$ **then** $t_i \leftarrow t_i * 2$
5: $\quad\quad t_i \leftarrow t_i + s_i$
6: $\quad\quad$ **if** $snubbing$ **and** $r_i < s_i$ **then** $t_i \leftarrow t_i * 1.5$
7: $\quad\quad$ **if** $\frac{s_i}{r_i} >= 10$ **then** $t_i \leftarrow t_i * 1.5$
8: $\quad\quad$ **if** $d_i > 0$ **then** $t_i \leftarrow t_i * (1 + \frac{d_i}{r_i})$
9: $\quad$ **end if**
10: $\quad$ **if not** $incoming$ **then** $t_i \leftarrow t_i * 2$
11: **end for**
12: $V \leftarrow V \setminus \{v_i \mid \max(t_i)\}$

Every neighbor $v_i \in V$ (where V is the set of neighbors of a peer) is ranked according to a metric t_i that measures the lack of utility of the neighbor to the peer. That is, the higher the the value of t_i, the less useful the neighbor is to the peer and higher the chance that it will be selected for disconnection. The value of t_i is based on the elapsed time since the last data exchange with the neighbor (l_i, line 2), and is possibly further incremented according to other factors.

If the peer executing OD is a seeder it only considers the time elapsed since the last block request from the neighbor, because seeders do not need contribution. Otherwise (i.e. peer is a leecher), the mechanism employs the following extra criteria in order to more precisely classify the utility of neighbors (lines 3-8):

- **interesting:** if the neighbor does not have any piece that could be of interest then the value of t_i is doubled (line 4);
- **snubbing:** a neighbor is snubbing when it announces interesting pieces, but does not send them (notice that the remote peer is not obliged to send pieces). In this case, the time s_i that the neighbor has been snubbing is added to the metric (line 5), which can be further increased in 50% if the given neighbor contributed with less data than was sent to it (line 6);
- **sharing ratio:** if the ratio between data sent to the neighbor s_i and data received from the neighbor r_i is lower than 10% (i.e. a potential free-rider), the metric is increased in 50% (line 7);
- **corrupted data:** if any data received from the neighbor has been discarded (either corrupted or duplicated), the metric is added by the ratio between the total data discarded d_i and the total received r_i (line 8).

In the last step, the peer verifies if the connection is incoming, that is, if it was initiated by the neighbor. If it was initiated by the peer itself, then the value of t_i is doubled (line 10). The peer then selects the neighbor with the least utility (that is, the highest value in t_i) and disconnects it (line 12).

The focus of our evaluation efforts is on the impact of the above algorithm in swarm performance and topological characteristics. The methodology employed in our experimental evaluation is presented in the next section.

IV. Methodology

To guide our experiments, we define a series of fundamental questions to be answered. The first two are related to the swarm performance, while the other two address topological properties:

Q1: Do peers experience shorter download times when OD is employed?
Q2: Does OD increase the utilization of resources available in the swarm?
Q3: Can OD improve the robustness of swarm topologies?
Q4: Does OD lead to improved topologies with respect to content dissemination?

In the remaining of this section, we (*i*) describe the experimental environment employed in our evaluation; (*ii*) present the main scenarios evaluated; and (*iii*) define the metrics analyzed during the experiments.

A. Environment

We performed multiple experiments with private swarms using Planetlab as testbed. Planetlab enforces a strict control over the available resources (such as memory, processor time and network bandwidth) in each of its nodes. This limited use of resources reflected in some aspects of our experiments. We chose as user agent Vuze version 4.6.0.5 (released in March 2011), because of three factors: popularity, availability of source code, and implementation of OD. On the downside, Vuze is a feature-rich agent and its memory requirements represented a restriction for PlanetLab nodes. Thus, to maximize the initial number of usable nodes and reduce the chance of having them killed during runs, we reduced the two parameters that were most influential for memory consumption. These were the size of the file being shared, set to 16 MB, and the maximum number of connected nodes, set to 25.

Download and upload rates of peers were adjusted to 160 Kbps and 40 Kbps, respectively, so that the best download time would be in the order of dozens of minutes, as observed in public swarms. The initial seeder had its upload capacity set to 320 Kbps.

Even so, not all nodes could be used. First, operating system misconfigurations and lack of storage space affected some machines in the early stages of the experiments. Second, some nodes had unstable network conditions and were available

[2]A newer version of the client was released since our experimental analysis begun. However, the evaluated mechanisms suffered no modifications between the latest version and the one we used.

only for limited periods of time. To guarantee the validity of the results, we selected "more stable" nodes with permanent network connectivity. There were approximately 400 usable nodes, which were divided in two swarms of 200 nodes each. This swarm size is similar to the one employed in the experiments of recent work [20] regarding BitTorrent.

Another environmental factor that impacts performance in BitTorrent is average peer reachability, that is, the ratio of peers that can accept incoming connections from neighbors. Peers that are connected to the Internet through a NAT or firewall often are not directly reachable by other peers, thus limiting the effective connectivity of the swarm. Studies about typical peer reachability in the Internet diverge on their results. Authors of [21] claim that an average of 55% of peers are reachable in BitTorrent swarms. Authors of [13], however, measured an average peer reachability of 8%. We also performed an experiment to define a reachability level to be used in our experiments. We captured a total of 860,954 torrents; from tracker announcements we found 768,981 distinct peers, from which 48% were successfully contacted by our crawler. Based on our measurement (whose results are in line with [21]), we adopted a default value of 50%, and include in the paper a set of experiments to evaluate the impact of different reachability levels. The obtained results are discussed in detail later in the paper.

Finally, swarm performance can be also be affected by the arrival process of peers. Authors of [22] present an extensive study regarding flash crowds on BitTorrent swarms. Among the results, authors show that (*i*) most of major flash crowds in BitTorrent occur soon after the swarm creation and (*ii*) the performance of peers is reduced during the phenomenon. Hence, we investigate the influence of peer arrival in our results. We employ peer arrival processes with different intensities of flash crowd (based on traces collected from real swarms) to assess the behavior of OD under these circumstances.

Next we present the set of scenarios employed in the evaluation of OD, which were defined according to our objectives and the experimental environment available.

B. Scenarios

Our baseline scenario has the goal of measuring the influence of OD in swarms according to the questions presented at the beginning of the section. Average peer reachability is set to 50%. The peer arrival process is modeled after traces of real swarms taken from a P2P trace repository[3]. The maximum arrival time of peers in the trace used for this scenario is 12 min. After finishing their download, leechers remain in the swarm until their sharing ratio reaches 1. The initial seeder, on its turn, is always present in the swarm.

The second scenario aims at better quantifying the impact of peer reachability in swarms with and without OD. We employ the same parameters from the baseline scenario, but vary the peer reachability between 10% and 100%.

The third and final scenario evaluates the impact of different arrival processes with and without OD. We select different traces from the aforementioned repository, containing maximum arrival times ranging from 2 to 42 min. We then employ these arrival patterns while keeping the remaining parameters as in the baseline scenario.

C. Metrics

To answer the research questions posed earlier, we employ the following metrics.

- **Download time:** time spent by a peer since it enters the swarm until completion of content download. It provides a good measure related to the user perception about the quality of a file sharing system;
- **Upload utilization:** indicates the percentage of upload bandwidth used. The value is normalized by the total bandwidth available. The higher the usage of upload capacity, the better the use of resources available by peers and the dissemination of content through the swarm;
- **Initial seeder closeness:** indicates the average distance of the initial seeder to all other peers in the topology graph, showing how central the peer is in the graph. It can be seen as a measure of how long it will take for information to spread from the initial seeder to other peers in the network [23];
- **Initial seeder eccentricity:** the largest distance of the initial seeder to any other peer in the swarm. Higher values mean more hops to spread the content and less upload utilization of farther peers;
- **Copies of the Rarest Piece:** indicates how many peers have to leave the swarm to cause its death.

V. RESULTS

In this section we present the results obtained from the extensive set of experiments conducted in PlanetLab between October, 2011 and April, 2012. All values represent the central tendency obtained from multiple redundant runs of each experiment. Nonetheless, graphical displays of swarm topologies inherently correspond to a single execution; in this case, we had to manually inspect graphs and select a representative one for that scenario. We also analyzed the variance of the obtained central tendencies considering a confidence degree of 90%. Thus, we include errorbars in our figures to illustrate the variance when significant values were observed.

Finally, we note that despite our selection of "more stable" nodes, failures were still possible, and indeed some swarms had less than their full numbers complete their downloads. For the sake of the analysis, we consider a swarm to be successful when 90% of the peers (i.e. 180) complete their downloads. Therefore, without impacting the validity of results, some cases may display results for less than 200 peers.

A. Baseline Scenario

We begin our analysis with the results related to our baseline scenario. Figure 1 presents the download time experienced by peers, showing the corresponding values in non-decreasing order.

The average download time of the swarm without OD is 110.53 min. As Figure 1 shows, a small number of peers (around 20) may have shorter download times *without* OD. When OD is employed, however, the majority of the swarm achieves better download performance and the swarm average download time is reduced to 77.51 min. The comparison of

[3]Available in p2pta.ewi.tudelft.nl

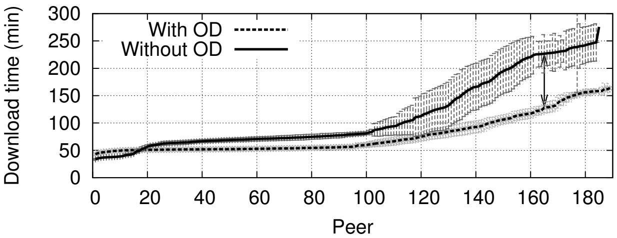


Figure 1. Swarm performance

average download times reveals a performance gain of nearly 30% when OD is employed. Thus, answering Q1, OD improves download times for the majority of peers participating in the swarm. We also observe that the variance in download times is smaller among peers that finish first, up to the 96th peer. This occurs due to the churn from the peers that finish earlier and leave the swarm. We also observe that the OD reduces the variance observed in download times of peers which finish later.

The observed gains in download times in theory occur because OD terminates connections that are not beneficial to a peer, allowing new neighbors to connect. This process could lead to a better use of the available upload resources by peers, which would help explain the better overall performance when using OD. To verify this hypothesis, we analyze the usage of swarm upload capacity. This is shown in Figure 2, in which the horizontal axis presents the swarm lifetime in minutes.

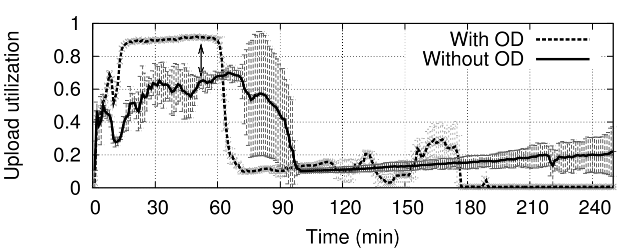
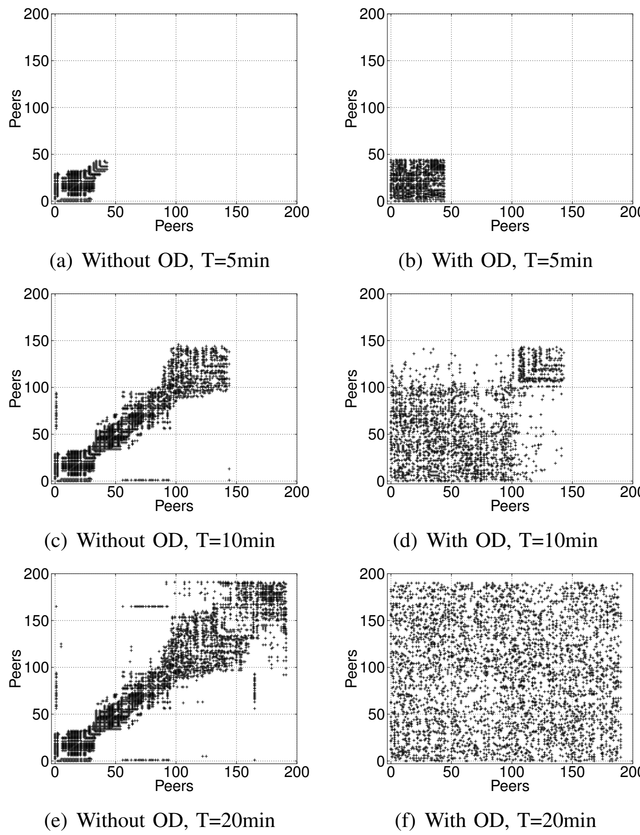


Figure 2. Average upload utilization throughout swarm lifetime

Without OD, the peak usage of upload capacity is 70%. This peak lasts for around 20 min or 8% of the swarm lifetime (approximately 4h). The upload usage is below 60% of available resources during almost 75% of the time. It is clear in Figure 2 that OD increases the upload usage peak, that is, from 70% to 91%. This peak lasts for almost 40 min, or nearly 25% of the swarm lifetime (164 min). We also observe that without OD there is a high variance in the upload utilization. We found that this is related to peer reachability, which induces high variation in individual upload capacity. Results with OD show much smaller variation, because the mechanism allows all peers to saturate their upload capacity, increasing the content dissemination.

To better quantify this gain, we calculate the average upload utilization throughout the entire swarm lifetime. The resulting ratio provides the average usage of swarm resources. Without OD, the average utilization is 42%. In contrast, this value is increased to 63% of the total upload capacity with OD, or 50% gain. These observations let us answer Q2: the use of OD leads to better resource usage, thus increasing swarm overall performance.

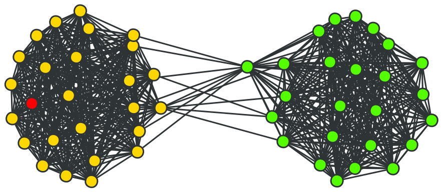
To answer Q3 and Q4 we need to look at how OD influences the topological characteristics of the swarm. This can help explain the performance gains obtained with OD. Our first step is the analysis of various snapshots taken throughout swarm lifetime. These snapshots were used to plot the connection matrices in Figure 3. We present snapshots of three different moments: (*i*) at 5 min, when the swarm is beginning and only 25% of peers have joined; (*ii*) at 10 min, when around 75% of peers have arrived; and (*iii*) at 20 min, when all peers have joined, none has left yet, and the topology is in a stable situation. Each point represents a connection established between two peers. Both axis show peers according to their order of arrival in the swarm.



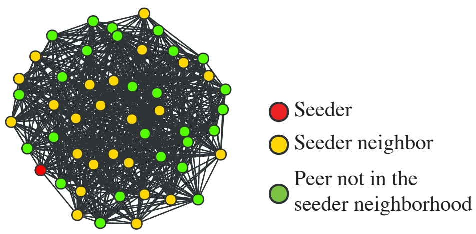
(a) Without OD, T=5min
(b) With OD, T=5min
(c) Without OD, T=10min
(d) With OD, T=10min
(e) Without OD, T=20min
(f) With OD, T=20min

Figure 3. Swarm connectivity matrices

Snapshots of the swarm without OD (left column of Figure 3) show that peers tend to establish connections according to their arrival time. In the analyzed scenario, peers are connected to 19 neighbors on average. When OD is used (corresponding right column), we observe a very different pattern, unrelated to the arrival order. OD also increases the average neighborhood of the swarm to around 24 peers (out of 25, the limit). The difference in the average neighborhood can be observed by comparing Figures 3(a) and 3(b): when OD is employed the connection matrix presents a higher number of points. To evidence even further the effects of OD on the connection pattern among peers, we contrast the corresponding topologies in Figure 4.



(a) Without OD

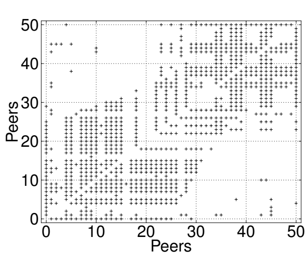


(b) With OD

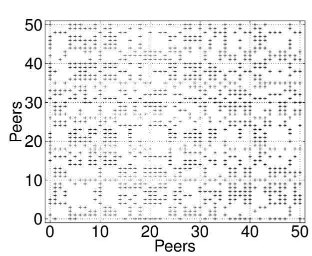
Figure 4. Topology representation

As shown in Figure 4, without OD two clusters are formed in the topology, with few connections among them. This separation into two clusters negatively affects download times: pieces will be disseminated in the right cluster only after they are obtained by peers with connections established with the left cluster (the one with the initial seeder). This occurs because the first group of peers to arrive in the swarm form a well connected cluster around the first seeder, leaving few connection opportunities for peers arriving later. These few connections also reduce the robustness of the topology: in a situation of churn, the connections among clusters may be severed, resulting in a partitioned swarm. The contrast to the case with OD is obvious: clustering is eliminated, leading to a higher number of connections. Consequently, OD increases the robustness of the swarm.

Note that the topologies in Figure 4 were captured in smaller swarms (50 peers), to allow graphical representation. To make sure that the above findings hold for larger swarms, we compare the two cases using connection matrices. Figures 5(a) and 5(b) present the connection matrices for swarms with 50 peers; by comparing them with Figures 3(a) and 3(b), one finds a similar pattern.



(a) Without OD



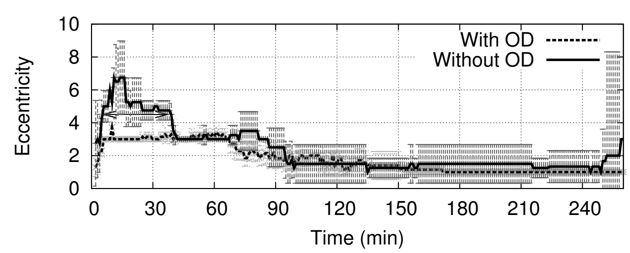
(b) With OD

Figure 5. Connectivity matrix in a swarm with 50 peers

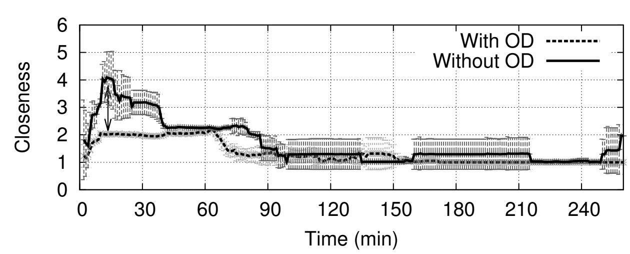
The results regarding topology robustness are summarized next, allowing us to answer Q3. We observed that, without OD, peers tend to establish connections according to their order of arrival. This may lead to lower average swarm neighborhoods, as well as the formation of clusters. OD, in contrast, prevents this situation and consequently contributes to swarm robustness.

Recall from Section II that there is no consensus about which graph model may be used to represent BitTorrent swarms. During our experiments, we attempted to characterize the observed swarm topologies as a known graph model. Authors of [7] suggest that such comparison may be done with values of the clustering coefficient and the characteristic path length, which is the average length of the shortest path between each pair of vertices in the graph. By comparing the values obtained from real graphs and theoretically generated ones, we can verify whether the real graphs follow a certain model. Topologies with OD present a characteristic path length ratio of 1.099, which is close to value expected from random and *small world* graphs (i.e., 1). However, the clustering coefficient ratio of the topologies is around 0.33. This value indicates that the graph is neither random (clustering coefficient ratio should be close to 1), nor *small world* (in this case it should be much greater than 1). Thus, OD does not generate a graph that matches the properties of a random one or a *small world*. The behavior described above was observed in all topologies of swarms that employed OD in our experimental scenarios. Also, the above result is in line with previous work that explores swarm topology characterization [6], [7].

Two other topological aspects are influenced by OD: the eccentricity and the closeness of the initial seeder, respectively shown in Figures 6(a) and 6(b). Both figures represent the swarm lifetime in their horizontal axis. The metrics are associated with the number of hops between the initial seeder and the remaining of the swarm, so the lower the better.



(a) Eccentricity

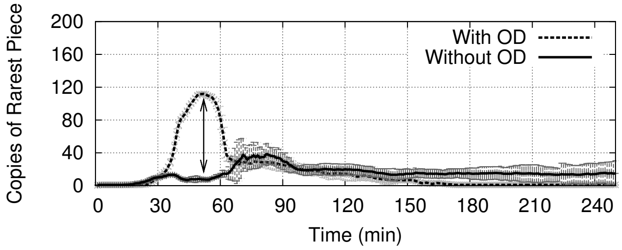


(b) Closeness

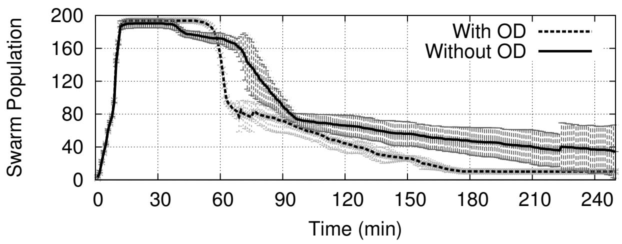
Figure 6. Swarm distance to the initial seeder

Figures 6(a) and 6(b) show that the impact of OD in the topology is much stronger in the beginning of the swarm. We observe that OD generates, in the first 30 min, a gain of up to 37% in the swarm eccentricity and of up to 50% in the swarm closeness. That is, OD reduces the distance of the initial seeder to the remaining peers, consequently favoring the dissemination of pieces in the swarm. Afterwards, there is little or no gain provided by OD, since the values, in both metrics, are similar for the cases with and without OD. This is because peers leave the swarm and its topology can be reorganized. Further, these observations are confirmed by the diameter of the network: the employment of OD reduces the topology diameter from 7 to 4 hops. Variance shows that, with OD, all peers tend to keep the same closeness and eccentricity throughout the swarm lifetime. In contrast, without OD the variance is a noticeably higher in the first half hour of the swarm, which has not yet converged to a stable topology.

The above results reflect a change in the way content pieces are disseminated in the swarm. This aspect may be evaluated with the analysis of the number of copies of the rarest piece, presented in Figure 7(a). The result needs to be viewed together with the number of peers in the swarm, shown in Figure 7(b). The horizontal axis of both Figures present the swarm lifetime in minutes.



(a) Copies of Rarest Piece



(b) Swarm Population

Figure 7. Robustness level (based on rarest piece)

Results in Figure 7(a) show that without OD in the first hour only a maximum of 15 peers (out of 200) hold a copy of the rarest piece, which leaves the swarm more vulnerable to attacks and churn. Later, the rarest piece peaks at approximately 40 pieces and then decreases. Towards the end of the swarm, rarest piece dissemination gets stable in 15 copies. The variance of available copies does not show a relevant difference when OD is used. We notice though that the variance is higher after 60 minutes in both cases. This is caused by the churn generated by peers that finished downloading the content.

When OD is employed, the dissemination of the rarest piece presents a peak of 110 pieces between 40 and 63 minutes. Comparing peaks of dissemination, we observe that when OD is used there are 2.75 times more copies of the rarest piece. After the observed peak the number of copies of the rarest piece slowly falls from 30 to 1 while the final peers finish downloading the content. We note that this reduction happens at the final moments of the swarm (which ends at 160 minutes) when most peers already left it. The higher dissemination peak generated by OD allows peers to download the content faster, thus reducing the swarm lifetime in 90 minutes.

The previous results show that OD creates a topology in which peers are closer to the first seeder. This favors the dissemination of content in the swarm, specially when no peer has left it yet, as observed in the analysis of the rarest piece. Thus, answering Q4, the use of OD generates a topology that contributes to the more efficient dissemination of content among peers.

B. Average Peer Reachability

The previous analysis of swarm download times, in Figure 1, showed that some peers were faster than others. We analyzed the potential causes and found out that in general the faster peers corresponded to those directly reachable. Measurements indicate that, when OD is not present, peers not directly reachable complete their download in 152.45 min on average, while the peers that can be directly contacted finish in 76.73 min. With OD, these values are reduced to 112.64 and 60.45 min, respectively. Hence, we observe a gain of almost 50% in download performance when peers are directly reachable. These results motivated us to evaluate in more detail how reachability influences swarms with and without OD.

Our analysis begins with the average gain obtained with the use of OD, presented in Figure 8. Different values of peer reachability were evaluated, as shown in the horizontal axis.

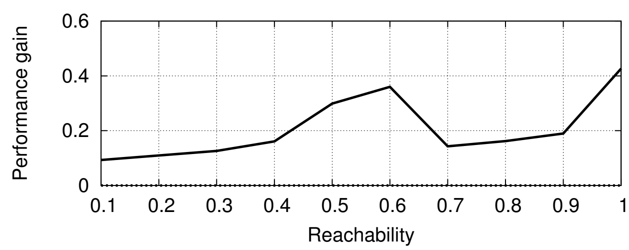


Figure 8. Peers average performance

First, notice in Figure 8 that OD can provide a consistent advantage, with gains between 9% and 36%, depending on the level of reachability. Second, that the advantage provided by OD in comparison with the non-OD swarm seems to vary unexpectedly. The explanation is as follows. Initially, increasing reachability provides higher gains in comparison with the case without OD, starting at 9% and peaking at 36% when reachability is 0.6. The gain provided by OD is directly proportional to the reachability level because, at low levels, it may not be possible to connect to another peer (the "disconnect to connect" fails). At reachability 0.6, the highest gain is achieved, because the neighbor disconnected by OD can most often be successfully replaced with another connection.

However, part of this advantage is lost for swarms with higher reachability, from 0.7 and above. In this situation, the swarm does not need to rely that much on OD to establish efficient connections. Thus, when reachability goes from 0.6 to

0.7, the performance with OD improves slightly (from 57 min to 56 min), but the *advantage* provided by OD is reduced to around 18%. Finally, when reachability goes from 0.7 to 1.0, the gain provided by OD grows from 14% to 21%. This occurs because of a tendency to the formation of clusters when OD is not used and reachability levels are high. This phenomenon will discussed later in this section. Hence, addressing Q1 in terms of reachability, we find that OD is consistently beneficial to swarm download times, with a gain of at least 9% and potentially much higher.

Results in Section V-A indicate that the positive effect of OD in download times is related to a better use of the swarm upload capacity. To verify if this insight also holds when peer reachability is varied, we measured the gain obtained with OD in the average use of upload capacity for different degrees of reachability. Below, we discuss the results.

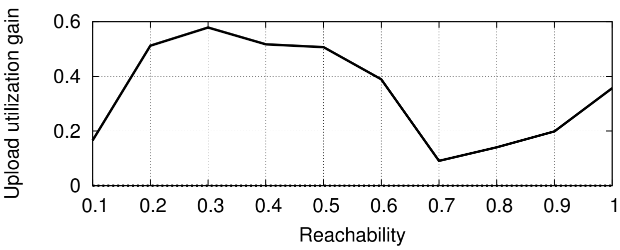
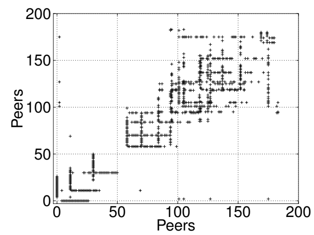


Figure 9. Influence of peer reachability in upload utilization

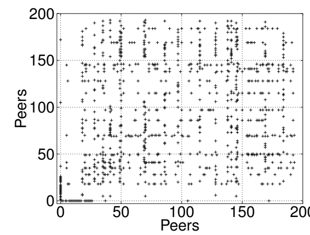
First notice in Figure 9 that OD enables an overall gain in upload resource utilization, with at least 9% for all levels of reachability. Notice, also, that such gain varies considerably, between 10% and 60%, which means that the *reachability level of peers strongly influences the advantage of OD*. Third, there are two turning points in the curve: 0.3 and 0.7, which correspond to highest and lowest relative gains, respectively. The explanation for this follows.

With a very low level of reachability, the possibility of connection to a seeder different from the initial one is very small, thus limiting the gain of OD to around 20%. The advantage in resource usage increases as the level of reachability goes from 0.1 to 0.7. Notice that the gain is highest (60%) when reachability is 0.3, because this level is insufficient for non-OD swarms to establish an efficient topology, but high enough for OD to randomize the topology (allowing peers to the search for better neighbors). Between 0.3 and 0.7, the performance of the non-OD swarm increases substantially and the positive impact of OD is progressively reduced. From 0.7 onwards, the gain by OD in upload utilization increases from 9% to 15%. This happens because the performance with non-OD decreases due to a tendency to the formation of clusters among peers when the peer reachability is high. This phenomenon will be discussed later in this section. Summarizing our results with respect to Q2 and the effect of reachability, we found that OD increases the usage of the resources available in the swarm independently of the number of reachable peers.

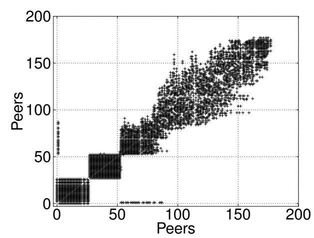
We also analyzed the topological properties of swarms with different degrees of reachability. Figure 10 shows the connection matrix of swarms considering two extreme values, 10% and 100% of reachability. Matrices on the left and right correspond, respectively, to swarms without OD and with OD. Peers are ordered according to their order of arrival in the swarm. Matrices were generated based on snapshots taken in moments in which all peers had already entered the swarm and established their neighborhood.



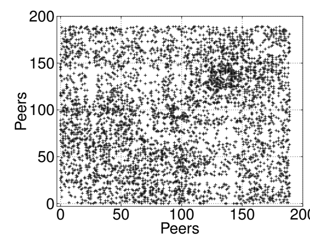
(a) Without OD, 10% reachability



(b) With OD, 10% reachability



(c) Without OD, 100% reachability



(d) With OD, 100% reachability

Figure 10. Connectivity matrix for different reachability factors

We observe that, without OD, peers tend to establish connections according to their arrival order (consistent to the previous results). The matrix in Figure 10(a) is much sparser due to the lower number of connections associated with the low degree of reachability.

Figure 10(c) also presents an interesting phenomenon: the first 50 peers to arrive in the swarm form two highly connected clusters that share few connections with the remaining peers. The first cluster is the one with most neighbors connected to the first seeder, and thus these will be the first peers to complete the download. Peers that arrive later will struggle to obtain pieces distributed within the initial cluster. This explains the lower performance under full reachability as observed in Figures 8 and 9.

Matrices that represent swarms with OD show that the mechanism reduces the chance of forming such clusters. This leads to a more efficient dissemination of the content among all peers, reducing download times in general. We also notice the effect of a lower degree of reachability in Figure 10(b): even with the benefits of OD, the matrix results much sparser, with less connections among peers and consequently a decrease of swarm robustness. Thus, in regard to Q3, the use of OD favors the topological formation of the swarm under different peer reachability scenarios, increasing the performance of peers and the robustness of the swarm.

To complement our insights regarding topological characteristics, we show in Figure 11 how different scenarios of peer reachability affect the closeness of the initial seeder. We evaluate different moments of the swarms in situations with and without OD and present the observed average values.

We first consider the case without OD. Figure 11 shows that for up to 30% of reachable peers, the initial seeder closeness

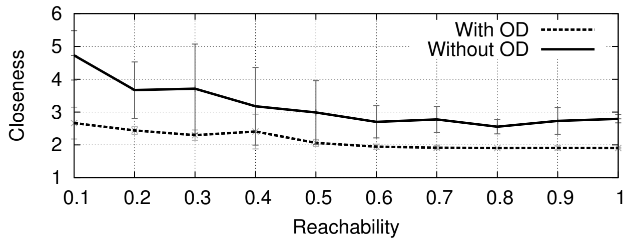


Figure 11. Average initial seeder closeness

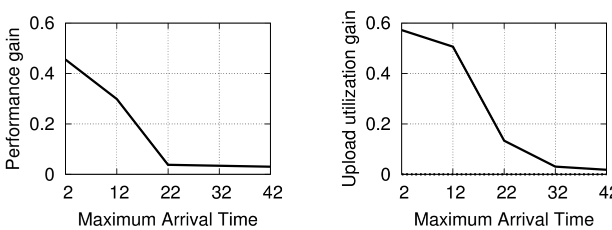
is approximately 4 hops. From 40% onwards, the value drops almost 1 hop and remains a little less than 3 hops on average. Notice that OD affects the variance of the closeness: its values are stabler when the mechanism is employed.

In comparison, the impact of reachability in the closeness is consistently reduced with OD. For all reachability values, the closeness is below 3, and from 50% onwards, below 2 hops on average, as a consequence of a better topology organization caused by OD. Using OD enables a substantial gain, in most cases superior to 0.5 hops, therefore answering Q4 in terms of reachability.

C. Arrival time

As mentioned by authors in [22], the occurrence of flash crowds in BitTorrent swarms may degrade the performance of peers due to an acute peak in resource demand. In this section, we assess the influence of OD considering different patterns of peer arrival. We focus our analysis in situations of flash crowd that occur at the beginning of the swarm, since these are the most common [22].

In our evaluation, we vary the arrival pattern by switching the traces used to model peer arrival. We identify traces by the time that the last peer arrives in the swarm. The chosen traces have maximum arrival times ranging from 2 to 42 min, which correspond to a flash crowd and a sparse arrival of peers, respectively. We first evaluate the relative gain provided by OD in download time and usage of swarm resources, assuming different patterns of peer arrival.



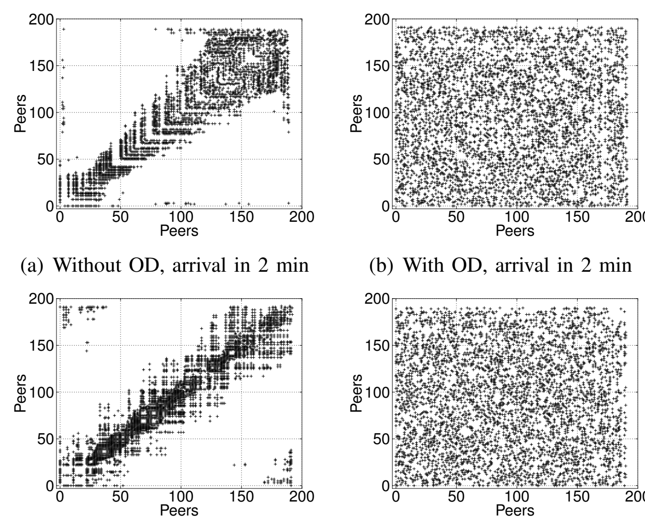
(a) Download (b) Upload

Figure 12. Peers average performance

According to Figure 12(a), OD can provide substantial performance gains in scenarios with arrival times up to 12 min. More specifically, when the maximum arrival time is 2 minutes, which correspond to a flash crowd, we observe a gain of at least 40% in download time. With maximum arrival times of 22 min or longer, the swarm is capable of handling the resource demand regardless of OD. Thus, in a situation of sparse peer arrival the performance gain of OD is small. Thus, answering Q1 regarding the arrival process of peers is, OD can substantially benefit a swarm in a flash crowd situation, but will have small impact otherwise.

Figure 12(b) shows that the use of OD improves the average upload utilization in all evaluated scenarios (all values are positive). It also can be noted that higher performance gains, of nearly 60%, occur when maximum arrival time is 2 min, which correspond to situations of flash crowd (consistently with previous analysis). As the arrival of peers becomes sparser the performance gain with OD falls. At very sparse peer arrivals a gain of 1% is observed. Thus, the answer to Q2 with respect to peer arrival process is that OD improves the utilization of resources available in the swarm, specially under flash crowd situations.

Next, we measure the impact of different arrival processes in the topological characteristics of swarms, with and without OD. Our first step is the analysis of the swarms connection matrices, as shown in Figure 13. Matrices in the left and right correspond respectively to swarms without OD and with OD at 45 minutes, moment in which all peers already arrived in the swarm. Matrices show peers according to the order they arrive in the swarm.



(a) Without OD, arrival in 2 min (b) With OD, arrival in 2 min

(c) Without OD, arrival in 40 min (d) With OD, arrival in 40 min

Figure 13. Connectivity matrix for different arrival intervals

Consistent with previous results, Figure 13 shows that in a swarm that does not employ OD, peers tend to establish connections according to their order of arrival in the swarm. We observe in Figure 13(a) that in a swarm with flash crowd peers that arrive early tend to form highly connected clusters, leaving few connections to peers that arrive later. However, without flash crowd, as in Figure 13(c), the connections among peers tend to become more evenly distributed, even if they are influenced by the arrival order. This is one of the reasons why peers present better performance when a flash crowd is not present.

In contrast, Figures 13(b) and 13(d) show that OD eliminates the clusters, with or without flash crowds. The two matrices display nearly identical behaviors, confirming that the arrival of peers does not impact on swarm performance when OD is active. These findings allow us to answer Q3 in regard to different arrival processes: in both situations, with and without flash crowd, OD generates topologies without the clustering that may harm swarm robustness.

Closeness of the first seeder is another topological characteristic that may be affected by different arrival processes. We measured this value at different moments of a swarm lifetime, and present results in Figure 14.

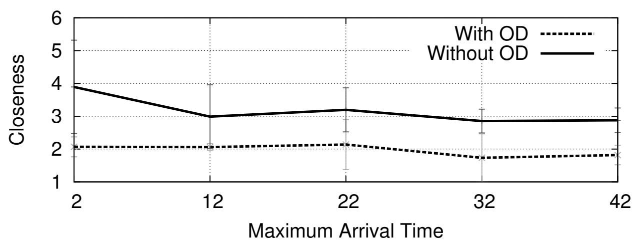


Figure 14. Average initial seeder closeness

Figure 14 indicates that, without OD, closeness of the initial seeder is approximately 3 hops except when maximum arrival time is 2 min (closeness is almost 4 hops). The closeness seems to be reasonably unaffected by arrival times when OD is active, with values that remain around 2 hops. Note that closeness is low with OD even in situations of intense flash crowd, when OD provides almost 50% of gain. Thus, answering Q4 in respect to arrival processes, we found that OD reduces the closeness of the swarm both with and without flash crowds, benefitting the dissemination of content.

VI. Final Remarks

Optimistic Disconnect is a mechanism available in BitTorrent clients that extends the default connection management. OD disconnects neighbors with low contribution in order to allow new, more interesting connections. This work presents the first experimental study to evaluate the impact of OD in BitTorrent swarms in regard to fundamental questions related to the overall performance and topological characteristics.

Results obtained from a comprehensive set of experiments in PlanetLab show that OD is beneficial to the swarm performance, contributing to lower the average download times experienced by peers and increasing the usage of swarm upload capacity. These benefits stem from the better swarm topological characteristics observed when OD is employed. Peers have larger neighborhoods and the topology has higher closeness to the initial seeder, contributing to faster content dissemination. OD also contributes to the generation of more robust topologies. Finally, the analysis of peer arrival processes shows that OD is reduces the negative effects of a flash crowd, keeping the swarm performance and topology stable.

In order to extend the insights of our current study, we hope to develop a sensitivity analysis of the various parameters employed by the OD algorithm. The choices taken by developers of user agents are not well documented, and many of the algorithm parameters may directly impact in the resulting swarm. We also plan to develop an analytical model of the influences caused by OD in BitTorrent swarms in order to obtain insights beyond the limits of the available experimental environment. We also hope to employ the insights gained from our study in the modeling and implementation of connection management mechanisms for P2P networks.

References

[1] "Fall 2011 global internet phenomena report," http://www.sandvine.com/downloads/documents/10-26-2011_phenomena/SandvineGlobalInternetPhenomenaReport-Fall2011.pdf.

[2] H. Zhang, G. Neglia, D. Towsley, and G. Lo Presti, "On unstructured file sharing networks," in *26th IEEE International Conference on Computer Communications (INFOCOM)*, 2007, pp. 2189–2197.

[3] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, 2004.

[4] B. Cohen, "Incentives build robustness in BitTorrent," in *1st Workshop on Economics of Peer-to-Peer Systems*, 2003.

[5] A.-L. Barabási, *Linked: The New Science of Networks*. Perseus Books Group, 2002.

[6] A. Al-Hamra, A. Legout, and C. Barakat, "Understanding the properties of the bittorrent overlay," *Computing Research Repository*, vol. abs/0707.1820, pp. 1–18, 2007.

[7] M. Kryczka, R. Cuevas, C. Guerrero, and A. Azcorra, "Unrevealing the structure of live bittorrent swarms: Methodology and analysis," in *11th IEEE Intl. Conf. on Peer-to-Peer Computing (P2P)*, 2011, pp. 230–239.

[8] C. Dale, J. Liu, J. Peters, and B. Li, "Evolution and enhancement of bittorrent network topologies," in *16th International Workshop on Quality of Service (IWQoS)*, 2008, pp. 1–10.

[9] P. Dhungel, X. Hei, D. Wu, and K. Ross, "A measurement study of attacks on bittorrent seeds," in *IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5.

[10] G. Urvoy-Keller and P. Michiardi, "Impact of inner parameters and overlay structure on the performance of bittorrent," in *25th IEEE Intl. Conf. on Computer Communications (INFOCOM)*, 2006, pp. 1–6.

[11] L. Zhong, X. Wang, and M. Kihl, "Topological model and analysis of the p2p bittorrent protocol," in *9th World Congress on Intelligent Control and Automation (WCICA)*, 2011, pp. 753–758.

[12] A. Farzad and H. Rabiee, "Modeling topological characteristics of bittorrent-like peer-to-peer networks," *IEEE Communications Letters*, vol. 15, no. 8, pp. 896–898, 2011.

[13] M. Yoshida and A. Nakao, "Measuring bittorrent swarms beyond reach," in *11th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2011, pp. 220–229.

[14] P. Racz, S. Oechsner, and F. Lehrieder, "Bgp-based locality promotion for P2P applications," in *19th International Conference on Computer Communications and Networks (ICCCN)*, 2010, pp. 1–8.

[15] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems," *SIGCOMM Computer Communications Review*, vol. 38, no. 4, pp. 363–374, 2008.

[16] B. Elser and T. Fuhrmann, "Here is your peer! – locating peers on a regional level with network coordinates," in *10th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2010, pp. 64–67.

[17] H. Wang, J. Liu, B. Chen, K. Xu, and Z. Ma, "On tracker selection for peer-to-peer traffic locality," in *10th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2010, pp. 1–10.

[18] M. Barcellos, D. Bauermann, H. Sant'Anna, M. Lehmann, and R. Mansilha, "Protecting bittorrent: Design and evaluation of effective countermeasures against dos attacks," in *27th IEEE Symposium on Reliable Distributed Systems (SRDS)*, 2008, pp. 73–82.

[19] A. Al-Hamra, N. Liogkas, A. Legout, and C. Barakat, "Swarming overlay construction strategies," in *18th Internatonal Conference on Computer Communications and Networks (ICCCN)*, 2009, pp. 1–6.

[20] D. Wu, P. Dhungel, X. Hei, C. Zhang, and K. Ross, "Understanding peer exchange in bittorrent systems," in *10th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2010, pp. 1–8.

[21] S. Kaune, R. C. Rumín, G. Tyson, A. Mauthe, C. Guerrero, and R. Steinmetz, "Unraveling bittorrent's file unavailability: Measurements and analysis," in *10th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2010, pp. 1–9.

[22] B. Zhang, A. Iosup, J. Pouwelse, and D. Epema, "Identifying, analyzing, and modeling flashcrowds in bittorrent," in *11th IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2011, pp. 240–249.

[23] M. E. J. Newman, "A measure of betweenness centrality based on random walks," *Social Networks*, vol. 27, pp. 39–54, 2005.

Summary Review Documentation for

"Disconnecting to Connect: Understanding Optimistic Disconnection in BitTorrent"

Authors: Matheus B. Lehmann, Lucas F. Müller, Rodolfo S. Antunes, Marinho P. Barcellos

Summary Review

The paper provides an experimental analysis of the impact of optimistic disconnection (OD) the overall performance of the swarm.

Strengths: The problem is important; the paper is sound and well-written.

Weaknesses: The reviews identified a few weak points. Several choices related to the evaluation settings seem unrealistic: the size of the downloaded file, the lack of heterogeneity in upload and download bandwidth, the artificial limitations in the maximum number of connections should be either motivated or substituted with more realistic parameters.

Related to this, more details about how the evaluation scenario is built should be provided; for example, authors merge an availability trace from the P2P archive with their own measurements of connectivity; but do not discuss whether there is a correlation between being highly available and connected. OD has a number of parameters; it would be interesting to evaluate the effect of each of these.

Plots should contain not only average figures, but also information related to variance and/or scatter plots of individual experiments.

The related work would benefit from more references to existing studies related to BitTorrent in general (not only OD), to other OD-like extensions in BitTorrent clients, and to OD-like techniques for other P2P applications.

Response from the Authors

We are aware that some parameters in our experimental environment present unrealistic values. As we pointed out in Section IV-A, the values of these parameters were influenced by limitations encountered in the PlanetLab environment and due to the high memory requirements for execution of the Vuze Bit- Torrent client. The size of the shared content and the number of neighbors that the client is connected with are the parameters most influenced by PlanetLab nodes limitations.

We conducted a set of experiments to specifically evaluate the impact of network heterogeneity. Our results indicate that there is a slightly positive impact in the usage of upload capacity when OD is employed in a heterogeneous network. However, the obtained values present a behavior very similar to the ones presented in Section V-A of the paper. This and the limited space available for discussion led us to the omission of results regarding network heterogeneity.

The goal of our experiments was to evaluate the implementation of OD that is employed in popular BitTorrent user agents. Among current implementations there is no difference in the OD algorithm, only slight variations in the employed parameter values. Regarding these, we would need a considerable discussion space to present a thorough evaluation of all parameters in OD. Thus, we choose to evaluate the default parameters used in current BitTorrent user agents. We are developing mathematical models to allow a more detailed evaluation of the algorithm parameters in larger systems, but results are not yet solid enough for presentation.

We agree with the reviewers that it is beneficial to provide more details about some of the parameters employed in our evaluation scenarios. We reviewed these points and presented the resulting discussions in Section IV-A of the paper. We also worked to include the variance of the presented results and the main insights regarding the observed values.

Our related work section is focused on studies about the topology of BitTorrent swarms because this is the main subject of our results. The most important references regarding OD are already cited in the paper. We could not include additional references due to lack of space, although we agree that this would have improved the paper. Finally, to the best of our knowledge, there are no studies describing the implementation of OD-like techniques in other P2P file sharing applications.