This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE P2P 2010 proceedings.

©2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The definitive version of this paper has been published in 10th IEEE International Conference on Peer-to-Peer Computing 2010 - IEEE P2P 2010, 2010, 10.1109\/p2p.2010.5569992.

Can P2P-Users Benefit from Locality-Awareness?

F. Lehrieder, S. Oechsner, T. Hoßfeld
University of Würzburg,
Institute of Computer Science, Germany
Email: lehrieder@informatik.uni-wuerzburg.de

Z. Despotovic, W. Kellerer, M. Michel
DOCOMO Communications Laboratories Europe GmbH
Munich, Germany
Email: despotovic@docomolab-euro.com

***Abstract*—Locality-awareness is considered as a promising approach to increase the efficiency of content distribution by peer-to-peer (P2P) networks, e.g., BitTorrent. It is intended to reduce the inter-domain traffic which is costly for Internet service providers (ISPs) and simultaneously increase the performance from the viewpoint of the P2P users, i.e, shorten download times. This win-win situation should be achieved by a preferred exchange of information between peers which are located closely to each other in the underlying network topology.**

A set of studies shows that these approaches can lead to a win-win situation under certain conditions, and to a win-no lose situation in most cases. However, the scenarios used assume mostly homogeneous peer distributions and that all peers have the same access speed. This is not the case in practice according to several measurement studies. Therefore, we extend previous work in this paper by studying scenarios with real-life, skewed peer distributions and heterogeneous access bandwidths of peers. We show that even a win-no lose situation is difficult to achieve under those conditions and that the actual impact for a specific peer depends heavily on the used locality-aware peer selection and the concrete scenario. Therefore, we conclude that current proposals need to be refined so that users of P2P networks can be sure that they also benefit from their use. Otherwise, a broad acceptance of the concept of locality-awareness in the user community of P2P networks will not take place.

I. Introduction

P2P networks are widely used in today's Internet for content distribution. Since they generate a large fraction of the total traffic in the Internet, a lot of research effort is recently put in the optimization of such P2P-based content distribution networks. In particular, those optimizations are designed to reduce so-called inter-domain traffic, which is said to be costly for the Internet service providers (ISPs). Furthermore, an IETF working group on application layer traffic optimization (ALTO) was established in November 2008 to standardize a protocol to guide the peer selection process and make it "better than random" as it is now.

Locality-awareness is one of the most promising concepts in this field. It equips peers with knowledge about the underlying network topology, e.g., to which autonomous system (AS) they belong. This information enables peers to prefer local neighbors, i.e., peers located in the same AS, for data exchange. Various implementations of this concept have been proposed and evaluated in literature, e.g. in [1], [2], [3], or [4]. All these studies show that a considerable amount of inter-domain traffic can be saved under certain circumstances when locality-aware peer-selection mechanisms are used. In addition, the performance of the users remains almost unaffected in most of the investigated scenarios, which leads to the conclusion that locality-awareness creates a win-no lose situation: the ISPs benefit and the users do not suffer.

In contrast to the aforementioned work, we show in this paper that a win-no lose situation is difficult to achieve under the real-life conditions we observe in today's Internet. The scenarios we investigate here differ mainly in two aspects from the ones considered in the previous work: First, we consider skewed peer distributions, i.e., a few ASes contain a large number of peers and most ASes contain only very few peers. According to the measurement studies presented in [5] and [6], these distributions are typical for today's BitTorrent swarms. Second, not all peers in a swarm have the same access speeds (cf. [7]) as assumed in most previous works. Conversely, we study the impact of locality-awareness for peers with different access speeds.

We assume that locality-awareness will only be successfully adopted in practice if the great majority of P2P users has an incentive to switch to the new mechanisms or at least does not object to do so. Some ISP-based solutions are under discussion which do not require the P2P users or the overlay providers to cooperate [1]. However, we argue that P2P developers will find ways to bypass those solutions if they lead to a reduced application performance for P2P users, e.g., by encrypting the data exchange. Therefore, we focus on the user's point of view and investigate the impact of different locality-aware peer selection strategies in scenarios motivated by measurement studies of real P2P networks. We show that skewed peer distributions and heterogeneous access bandwidths have a significant impact on locality-awareness and that there is no general no lose situation for all peers. Instead, it depends strongly on the concrete implementation of locality-awareness which peers will benefit from locality-awareness and which will not. In practice, a specific user will probably not know whether he benefits or loses in a concrete scenario. Therefore, we argue that P2P users will be hesitant to use these mechanisms because they cannot be sure that their application performance is not degraded. As a consequence, the locality-awareness mechanisms currently under discussion need to be refined so that they guarantee a

This work has been performed in the framework of the EU ICT Project SmoothIT (FP7-2007-ICT-216259). The authors alone are responsible for the content of the paper.

win-no lose situation in real-world scenarios.

The performance evaluation uses BitTorrent as an example P2P application for content distribution because BitTorrent is one of the most prominent P2P networks and currently most widely used. Furthermore, adaptations for locality-aware peer selections already exist for this protocol and are currently under discussion in the IETF.

The paper is structured as follows. Sect. II reviews previous work. In Sect. III we present BitTorrent and locality-aware peer selection mechanisms. After the description of our simulation setup, we show the results of our performance evaluation in Sect. IV. In Sect. V we conclude the paper.

II. Background and Related Work

We first review various approaches of locality-awareness as well as studies investigating mainly its benefits. Then, we present related work regarding the limits of locality-awareness. Finally, we give a short overview of measurements of skewed peer distributions in real BitTorrent swarms on which we base our evaluation scenarios.

A. Implementation Proposals for Locality-Awareness

One of the first approaches to locality-awareness in P2P networks was proposed in [3]. There, peers query a so-called "oracle" which is maintained by the ISP where the respective peers are located. The oracle ranks the peers according to the preferences of the ISP and sends this information back to the peers. Consequently, they can include traffic engineering policies in their peer selection. The evaluation is based on the Gnutella protocol. In contrast, we use the BitTorrent protocol in this study because it is the most widely used P2P protocol today, mainly contributing to the high load of P2P traffic in the networks.

In [1], Bindal et al. propose *biased neighbor selection (BNS)* for BitTorrent-like P2P systems. With BNS, the neighbor set of a peer is modified to contain preferentially peers in the same AS. This can for example be implemented by a modified tracker which is aware of the ASes where peers are located. The evaluation of BNS in [1] uses simulations with a homogeneous peer distribution of 700 peers over 14 ASes. The results show that a large fraction of the inter-AS traffic can be saved by BNS and that the median as well as 95th percentile of the download times are decreased. In [2], an approach very similar to BNS is investigated by experiments of up to 10.000 real BitTorrent clients which are homogeneously distributed among 10 ASes. According to the results, BitTorrent locality can be "pushed to the limit", i.e., the neighbor set of all peers contains almost only local peers, without degrading the performance for the viewpoint of a P2P user.

The P4P project [8] goes further and also considers the intra-AS topologies in addition. The authors propose to create an iTracker that communicates to the P2P application and gives recommendations about which peers to contact. Finally, a plugin called "Ono" for the open-source BitTorrent client Vuze is presented and evaluated in [9]. The main difference of Ono to the approaches described above is that it does not rely on a central entity which guides the inclusion of peers in the neighbor set of a peer. Instead, it uses the similarity of the redirection ratio of CDN servers as a metric describing how close peers are.

Biased unchoking (BU) is a complementary mechanism to BNS proposed in [4]. It does not influence the neighbor set of a BitTorrent peer but the choke algorithm which determines the actual data exchange in a BitTorrent P2P network. Like in [1] and [2], the study is based on a homogeneous peer distribution and shows that inter-AS traffic can be saved without decreasing the efficiency of the distribution process seen by the users, i.e., without increased download times. A similar approach to [4] is taken in [10] and the evaluation in a PlanetLab environment shows that download times can be slightly reduced on average.

The work presented in [11] compares different locality-awareness solutions for BitTorrent-based file-sharing and video-streaming. The authors point out that there is a trade-off between reducing inter-domain traffic and fairness among peers in terms of the data the peers upload. They also study the download and stall time of the peers but do not consider the impact of the distribution of the peers over the ASes.

In contrast to the studies mentioned above, we focus on scenarios with swarm sizes and peer distributions observed in real BitTorrent swarms [5], [6], i.e, with heterogeneous peer distributions and heterogeneous access bandwidths of the peers. We study their impact on the performance of a BitTorrent network for the P2P user and explain which users can benefit from locality-awareness and which not.

B. Limits of Locality-Awareness

In [12], the authors present three pitfalls for ISP-friendly P2P design: limited impact, reduced performance and robustness, and conflicting interests. They show that locality-aware peer selection has no impact when there are only very few peers of a swarm in the same AS. The second issue is similar to the focus of this paper and investigates application performance. The third pitfall considers different types of ISPs and the authors argue that strategical behavior of ISPs can limit the applicability of locality-awareness. The difference to this study is that we focus on the users' point of view and simulate a BitTorrent swarm with detailed peer behaviors, e.g., the choke algorithm with the tit-for-tat policy. In addition, we simulate the concrete implementations of locality-awareness mechanisms currently under discussion, i.e., BNS, BU, and the combination of both. In this way we show that different implementations lead to a different application performance, which is neglected in [12], and explain which users benefit in which scenarios by using BNS, BU, or the combination of them.

The Internet-draft "Mythbustering P2P Locality" [13] is a collection of facts and conclusions regarding the performance improvements by locality-awareness. It mentions that application performance may suffer and that a swarm may be weakened by a locality-aware peer selection without giving concrete evaluation results.

Our work differs from the above mentioned by the fact that we focus on the perspective of the user. This is crucial because P2P users and developers will not change to new algorithms if they have no incentive to do so and if only the ISPs profit from locality-awareness. To that end, we investigate the two main approaches of locality-awareness, BNS and BU, in scenarios derived from real-world measurement studies. This shows that the concrete implementation of locality-awareness and the scenario have a large impact on whether all or some users can benefit or not.

C. Skewed AS-Distributions of Real BitTorrent Swarms

A large-scale measurement campaign which analyzes the AS-distribution of peers in more than 250,000 BitTorrent swarms is presented in [6]. For the measurements, all movie- and music .torrent-files have been downloaded from the Mininova index site in April 2009 and the AS-distribution of the peers was obtained via distributed measurements. The study reveals that the AS-distribution is heavily skewed and the authors propose to model the probability $P(k)$ that a peer belongs to the k-th top AS of a swarm involving n ASes as $P(k) = a/k^b + c$. The parameters a, b, and c depend on the actual swarm size and the number of involved ASes.

The approach taken in [5] is very similar. The authors propose to model $P(k) = K/(k+q)^\alpha$ as a Mandelbrot-Zipf distribution where $K = 1/\sum_{k=1}^{n} 1/(k+q)^\alpha$ and the parameters q and α are used to fit the data. The measurements were performed during the years 2007 and 2008 and comprise more than 70,000 BitTorrent swarms mainly advertised by www.btmon.com.

III. Locality-Awareness Solutions for BitTorrent

We use the BitTorrent protocol as the basic file-sharing overlay because it is in widespread use and creates a significant share of today's Internet traffic. Furthermore, the wide majority of the studies presented in Sect. II is based on this type of overlay. In the following, we briefly describe the key mechanisms of BitTorrent and explain the adaptations for locality-awareness which are used in this study. A detailed description of BitTorrent can be found in [14] and [15].

A. Key Mechanisms of BitTorrent

The BitTorrent protocol forms a mesh-based overlay called *swarm* for each shared file. To facilitate a multi-source download, the shared file is split into pieces which are called *chunks*. These chunks are in turn separated into sub-pieces or *blocks*.

A peer joining a swarm initializes its neighbor set by contacting a *tracker*, i.e., an index server with global information about the peer population of a swarm. A standard tracker responds to queries with a random subset of all peers. Once a peer has received a list of contacts in the swarm, it tries to establish connections to them and adds them to its neighbor set if they accept the connection. All peers keep their neighbors informed about which chunks of the file they already have. In this way, a peer knows in which neighbor it is *interested*, i.e., which neighbors have chunks that it still needs, and it can signal this interest to them.

Every 10 seconds, a peer decides to which of its interested neighbors it will upload data to. These peers are called *unchoked*, the remaining peers are choked. In standard BitTorrent, there are 3 regular unchoke slots which are awarded to the peers that offer the currently highest upload rate to the local peer. This strategy is called *tit-for-tat* and provides an incentive for peers to contribute upload bandwidth to the swarm. Additionally, every 30 seconds a random peer not currently unchoked is selected for *optimistic unchoking* for the next 30 seconds. This allows a peer to discover new mutually beneficial data exchange connections. If the local peer has already downloaded the complete file, i.e., it is a *seeder*, the slots are given to all interested neighbors in a round-robin fashion.

B. Adaptations for Locality-Awareness

In order to evaluate the effects of locality-awareness on the user, we consider the two main client adaptations that utilize information about the underlying network topology. The best known approach to this is biased neighbor selection (BNS), which was first presented in [1]. We briefly describe the specific implementation of BNS used in our experiments as well as the second locality-promoting client mechanism, biased unchoking.

Both mechanisms need a locality metric to decide which peers are considered closer than others. The predominant solution in literature, e.g., used in [1], [2], [4], is to differentiate between peers in the same AS (local peers) and peers in other ASes (remote peers). Therefore, we keep this simple differentiation and assume that all peers have access to the information which other peers are local or remote to them. This could be implemented in practice for example by an information service provided by the ISP [16] or by contacting public databases.

1) Biased Neighbor Selection: BNS is a rather general approach suitable for most overlays. As a consequence, different forms of it are proposed in [1], [3], [8], [9]. It changes the process of the overlay neighbor selection, so that more local peers are established as neighbors. For BitTorrent-based overlays, there are two major alternatives for a BNS implementation, namely the tracker-based and the peer-based BNS. The first changes the responses of the tracker so that no longer random peers from the swarm are returned. Instead, the response includes a configurable share of local peers. Provided that the tracker has access to this kind of locality information, this change is easy to implement since it affects only the tracker [1]. However, it takes the decision about promoting locality from the end user.

Since the willing cooperation of the user in any locality-awareness approach is crucial for its success, we consider the second implementation alternative in this work. Peer-based BNS leaves it to the client to gather locality information about potential neighbors and to decide which contacts should be added to the neighbor set. Thus, the user is not forced to promote locality. The specific implementation used in our experiments queries the tracker for a much larger number of contacts (1000) than in standard BitTorrent clients (50). It then tries to keep a fraction l_{BNS} of local neighbors in its neighbor set. To this end, connections to peers in the same AS are established until the required number of local neighbors is reached or no more local contacts are known. In both cases, the missing number of neighbors is taken from remote peers until the BitTorrent standard minimum value of 40 neighbors is reached. If not mentioned differently, we set $l_{BNS} = 0.9$. This a conservative choice compared to [2], where values up to 0.999 are investigated, and [1] where 34 out of 35 neighbors are local if possible. However, $l_{BNS} = 0.9$ already shows that too strict preferences for local peers can increase the download times for some of the peers.

2) Biased Unchoking: The biased unchoking (BU) mechanism evaluated here was presented in [4] and is specifically targeted to BitTorrent-like P2P networks. It works as follows: local neighbors are preferred in the unchoking process, i.e., chunks are preferentially uploaded to local peers. To this end, the optimistic unchoke slot is assigned to a local neighbor with probability l_{BU} if a local neighbor is present. Via the tit-for-tat policy of BitTorrent, this small modification has also an impact on the three regular unchoke slots.

For high values of l_{BU}, the traffic exchange between peers in different ASes can be significantly reduced even if only a small number of peers is in the same AS [4]. If not mentioned differently, we set $l_{BU} = 0.9$. Again, that is a more conservative choice than in [4], but sufficient to show a negative impact. In addition, it leads to similar preferences for local peers as $l_{BNS} = 0.9$ in Sect. III-B1.

IV. Performance Evaluation of Locality-Awareness in Heterogeneous Scenarios

This study differs from previous work mainly in the scenarios we consider in our performance evaluation. Therefore, we first describe the chosen settings and parameters. Then, we investigate the impact of locality-awareness in heterogeneous scenarios by simulations and explain why our results differ from the ones obtained in homogeneous scenarios.

A. Simulation Scenarios

The simulation setup is very similar to the one used in [4]. We consider one BitTorrent swarm which exchanges a file of size 154.6 MB generated from an example TV show of about 21 minutes in medium quality. The file is divided into chunks of 512 KB and every chunk into blocks of 16 KB.

We simulate the swarm for 6.5 hours. It is initialized with the original seed before the arrival process of the regular peers starts. Since we are interested in the steady state of the swarm, we discard the warm-up phase of 1.5 hours in which the swarm population increases until the steady state is reached. Although the population of real swarms is not constant over the whole lifetime of a swarm, the steady state remains a good approximation for time periods in the order of hours.

The arrival process of the peers is modeled as a Poisson process with a mean inter-arrival time of 10 s. After a peer has downloaded the whole file, it remains in the swarm for an additional seeding time. The seeding time is exponentially distributed and on average 10 minutes long. As a result, we measured that the swarm contains on average about 100 to 200 peers. According to a measurement study of real BitTorrent swarms [6], these are typical values for medium-sized swarms observed in practice.

The multi-AS network we simulate forms a star topology and consists of one transit AS and $n = 20$ stub ASes where every AS $k \in \{1, \dots, n\}$ is connected to the transit AS via an inter-AS link (cf. Fig. 1). The tracker and the initial seeder are placed in this transit AS for symmetry reasons. The transit AS does not contain any further peer besides the initial seeder that has an upload capacity of 10 Mbps and goes offline after one hour of simulation time. This topology is simple, but sufficient for our purposes for the following reasons. The mechanisms for locality-awareness we study differentiate only between local and remote peers and ignore the actual AS-paths between the peers. Furthermore, we model the inter-AS links as well dimensioned and focus in our evaluation on the volume of inter-AS traffic and not on its paths. Consequently, we can abstract from a complex AS-level topology connecting the stub ASes and use a single transit AS for our simulations.

In this study, we investigate heterogeneous peer distributions, i.e., some ASes contain more peers than others. This is motivated by the fact most of the peers participating in a swarm are usually located in a small number of ASes [5], [6]. [6] proposes to model the probability $P(k)$ that a peer belongs to the k-th largest AS in a swarm involving n ASes in the form $P(k) = a/k^b + c$ and gives example values for $n = 40$ ASes of $a = 0.08$, $b = 0.8$, and $c = 0.01$ (for music files) and $a = 0.14$, $b = 1.16$, and $c = 0.01$ (for movie files). In [5] a Mandelbrot-Zipf distribution $P(k) = K/(k + q)^\alpha$ is used for that purpose with $K = 1/\sum_{k=1}^{n} 1/(k + q)^\alpha$. They provide concrete values for the parameters $q = 10$ and $\alpha = 1.33$ only for very large swarms with more than 5000 peers spread over roughly 1000 ASes. For swarms with less 300 peers the data presented in [5] suggests that an adequate value q is significantly smaller than $q = 10$, concrete values are however not given.

As a consequence, we use in this study the Mandelbrot-Zipf distribution in a simplified form

$$P(k) = \frac{1/k}{\sum_{i=1}^{n} 1/k}, k \in \{1, \dots, n\}. \quad (1)$$

which can be seen as the common denominator of [5] and [6]. The aforementioned values presented in [6] suggest $b \approx 1$ and $c \approx 0$. With $a = 1/\sum_{k=1}^{n} 1/k$ this leads to the Eq. (1) as well as the Mandelbrot-Zipf distribution in [5] for $q = 0$ and $\alpha = 1$. For the sake of readability, we refer to ASes with small AS numbers k as 'large ASes' and to those with high AS numbers k as 'small ASes'. For the homogeneous peer distribution, which we use for comparison, the peer arrival process is equally distributed over all stub ASes. Both peer

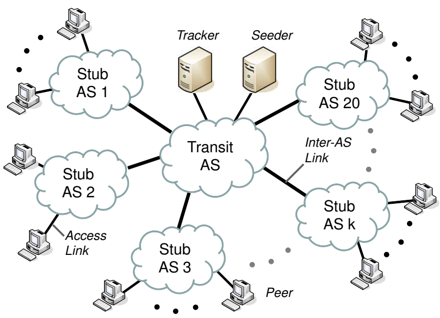


Fig. 1. Simulated network topology consisting one transit AS and $n = 20$ stub ASes. Every stub AS $k \in \{1, \ldots, n\}$ is connected to the transit AS which only forwards the traffic.

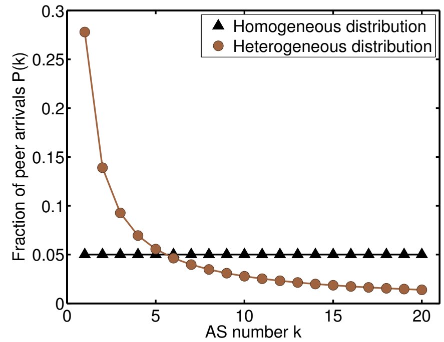


Fig. 2. Fraction of peer arrivals with respect to the AS number k. In the heterogeneous case the fraction of peer arrivals decreases for larger values of k (cf. Eq. (1)) while it is constant in the homogeneous case.

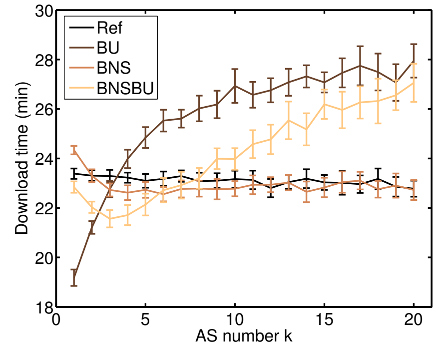


Fig. 3. Average download time of the peers in AS k in the scenario with heterogeneous peer distribution and heterogeneous access speeds.

distributions are illustrated for $n = 20$ ASes in Fig. 2.

In addition to heterogeneous peer distributions, we investigate scenarios with heterogeneous access speeds of the peers. Measurements in [7] show that the peers in a swarm can be clustered according to their access speeds. That means that for example 20% of the peers in a swarm have 128 kbps upload capacity, 30% have 256 kbps, 40% have 512 kbps and the rest is faster. The concrete numbers and cluster sizes depend mainly on the ISP where the peers are located. To keep things simple, we abstract from the concrete numbers and create two equal sized groups of peers: one with 16 Mbps down- and 1 Mbps upstream, as mentioned before, and one with 4 Mbps down- and 256 kbps upstream. Consequently, half of the peers in the swarm are "fast" peers and the other half are "slow" peers. As we will see, this suffices to show the effect that locality-awareness has on swarms with heterogeneous bandwidth distributions. When we study homogeneous access speeds, all peers are connected to their stub AS with an access speed of 16 Mbps downstream and 1 Mbps upstream, which are typical values for the DSL access technology.

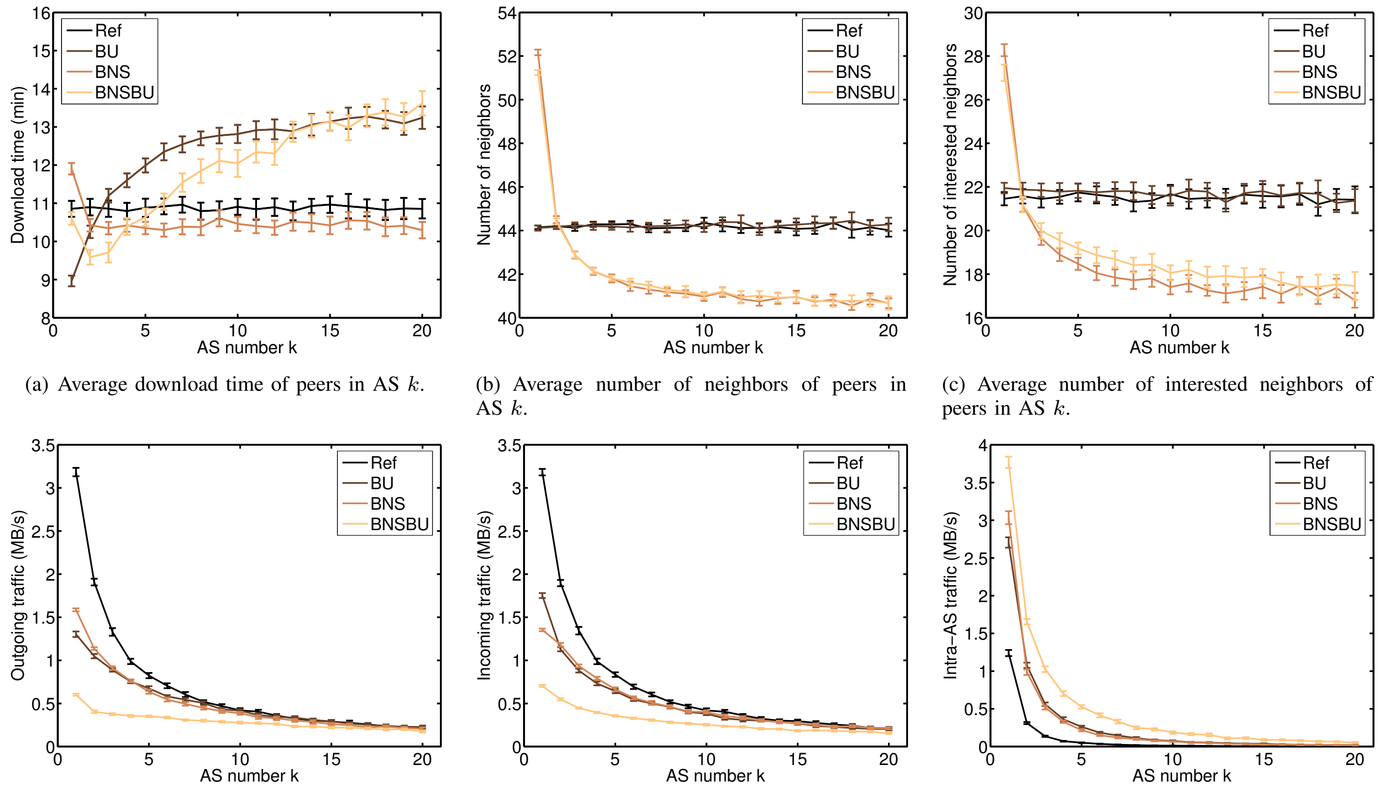
The simulator used in this work is based on the P2P simulation and prototyping Java framework ProtoPeer [17], [18]. The simulator contains a flow-based network model adopting the max-min-fair-share principle [19]. It mimics the property of TCP that the bandwidth of a link is shared among competing data flows. Still, the computational effort is smaller than for a packet-based network model. This is important since every simulation run consists of more than 2300 BitTorrent peers simulated in detail and several runs have to be performed for each scenario. On top of the ProtoPeer framework, we implemented the BitTorrent functionality and behavior as described in [14] and [15]. This implementation includes all key mechanisms, in particular the piece selection mechanisms, the management of the neighbor set, and the choke algorithm. Furthermore, the complete message exchange among the peers themselves and between peers and the tracker is simulated in detail.

For all scenarios we evaluate the BitTorrent reference implementation ('Ref'), its locality-awareness adaptations BNS and BU, and a combination of both ('BNSBU'). As performance indicators we measure the download time of the peers and the inter-AS traffic. We perform 15 simulation runs with different seeds for the random number generator for every configuration and show average values as well as 95% confidence intervals in the corresponding figures.

B. Homogeneous vs. Heterogeneous Scenarios

In our first experiment, we want to establish the fact that there are major qualitative differences between the usually considered homogeneous scenarios and a more realistic heterogeneous one. To this end, we compare the mean download times for peers in the different ASes. First, we consider a swarm where peers are homogeneously distributed among all 20 ASes and have the same access capacities of 16 Mbps downstream and 1 Mbps upstream. We measure the average download time of the peers in every AS during simulation and observe that they do not differ significantly between different ASes. This is not very surprising due to the completely homogeneous setup. Therefore, we omit the corresponding figure.

Next, we consider a swarm where both the peer distribution among the ASes as well as the access bandwidths are heterogeneous. Every AS contains on average 50% fast and 50% slow peers so that the average access speeds are the same for all ASes. In this scenario all the mechanisms lead to different download times (Fig. 3) while they had no impact in the homogeneous one. Some peers benefit from certain mechanisms by downloading the file faster while others take longer. With BU, for example, peers in large ASes (i.e., with small AS numbers k) can download the file faster than with regular BitTorrent. In contrast, peers in small ASes take longer. This is completely different when BNS is used. Thus, we conclude that there is no general no-lose situation for the users as a whole, and that the effect of a real-life scenario is different for different locality-awareness mechanisms. In the following, we will take a deeper look at the individual aspects of this scenario, namely the skewed peer distribution and the



(a) Average download time of peers in AS k.

(b) Average number of neighbors of peers in AS k.

(c) Average number of interested neighbors of peers in AS k.

(d) Average outgoing inter-AS traffic of AS k.

(e) Average incoming inter-AS traffic of AS k.

(f) Average intra-AS traffic of AS k.

Fig. 4. Simulation results for the scenario with heterogeneous peer distribution and homogeneous access speeds of the peers.

heterogeneous access bandwidths, and explain the different shapes of the curves in Fig. 3 in detail.

C. Impact of Heterogeneous Peer Distributions

We first consider the case where the access bandwidths are homogeneous, but the peers are distributed heterogeneously among the 20 ASes according to Eq. (1) as described in Sect. IV-A. The resulting average download time of the peers in the individual ASes is depicted in Fig. 4(a). When BU is used, the mean download time for peers in larger ASes decreases while the peers in smaller ASes take longer to download the file. This is due to the fact that a peer using BU preferentially unchokes local neighbors if possible, i.e, it uploads to local neighbors. However, peers in small ASes know only a small number, if any, of local interested neighbors, and therefore can only prefer them in the unchoking process in rare cases. Thus, the upload capacity of these peers is mainly distributed among all ASes. In contrast, peers in large ASes know interested local neighbors almost all the time. Consequently, they upload to a local neighbor very often. Therefore, the upload capacity of peers in a large AS is mainly utilized for connections within that AS. Furthermore, large ASes receive additional upload capacity from peers in small ASes when those peers have no interested local neighbor in their neighbor set. That shifts the global allocation of upload capacity in the swarm towards large ASes.

In contrast, BNS leads to longer download times in the largest AS in comparison to both regular BT and peers in the rest of the swarm. This effect seems counter-intuitive, but can be explained when considering the composition of the neighbor set of the peers. With BNS, peers in AS 1 have a higher number of neighbors than peers in other ASes (Fig. 4(b)) and therefore also more peers which are interested in downloading from them (Fig. 4(c)). The reason is that peers in AS 1 are not only contacted by others peers in AS 1 but also with a high probability by peers in small ASes because BNS fills the neighbor set with random peers if a sufficient number of local peers is not available. As a consequence of the increased number of peers interested in a peer in AS 1, its upload capacity is shared among a larger set of peers and every one of them receives a smaller portion. Finally, peers in AS 1 have a large number of local neighbors and download almost exclusively from them. Hence, a peer in AS 1 receives less upload capacity from the swarm than other peers.

Returning to the download times of the peers, BNSBU shows a combination of both effects described for BU and BNS. While the neighbor set composition has the same characteristics as in the pure BNS case, the unchoking policy of BU offsets the disadvantages of peers in large ASes. Therefore, the download times in larger ASes are shorter than in smaller ASes, but peers in the largest AS in our scenario still take longer to download the file than in the second largest AS.

Despite our focus on the user's perspective, we take a short look at the inter- and intra-AS traffic. We can observe that

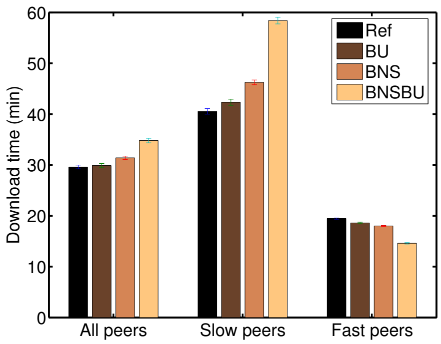


Fig. 5. Average download time per peer group, scenario "fast vs. slow ASes".

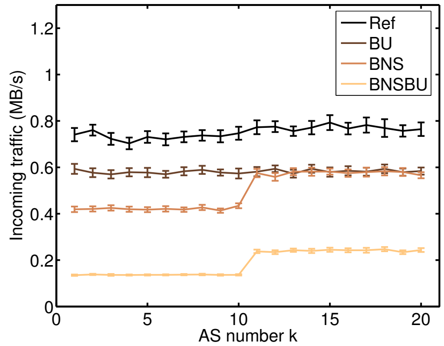


Fig. 6. Average incoming inter-AS traffic of AS k, scenario "fast vs. slow ASes".

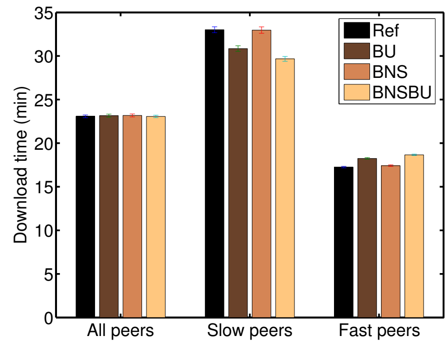


Fig. 7. Average download time per peer group, scenario "mixed access speeds".

the saving potential, both for outgoing and incoming inter-AS traffic, grows with the share of peers in an AS (Fig. 4(d) and (e), respectively). Especially when implementing locality-awareness both in the neighbor selection and in the unchoking process, larger ASes can reduce their incoming and outgoing traffic by a much larger factor than ASes with only a few peers in the swarm. If such an AS belongs to a Tier2 or Tier3 ISP which is charged by either its uploaded or downloaded traffic or the maximum of both, this translates into higher cost savings. With the right locality-awareness mechanisms, e.g., BU, both the end user in a large ISP and the ISP itself may benefit, however at the cost of other peers in the swarm.

In contrast, ISPs with only a small number of peers per swarm are not likely to profit much from locality-awareness because there are only few options for peers in these ASes to choose local neighbors. The better part of such a peer's contacts have to be from remote locations even when it applies BNS. All these findings regarding the reduction of inter-AS traffic are in line with literature (cf. [4]).

D. Impact of Heterogeneous Access Capacities

In this section, we investigate the effect of locality-awareness in swarms with heterogeneous bandwidths, but a uniform peer distribution. Like in Sect. IV-B, half of the peers have a fast access of 16 Mbps download and 1 Mbps upload bandwidth and the other half have a slow access with 4 Mbps download and 256 kbps upload bandwidth.

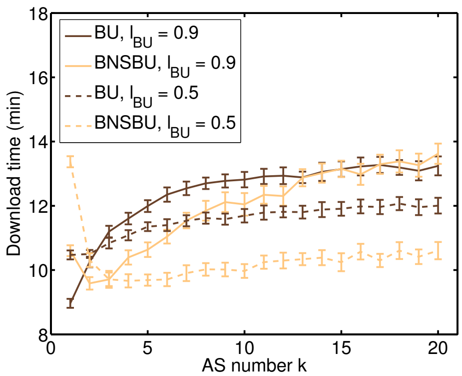
We consider two scenarios. The first one is called "fast vs. slow ASes". In that scenario, all the fast peers are located in ASes $k \in \{1, \ldots, 10\}$ and all slow peers are located in the rest of the ASes. This mimics the situation that some ISPs which are technologically more advanced than others offer their costumers higher access speeds. In the second scenario called "mixed access speeds", fast and slow peers are equally distributed over all ASes. This scenario is quite common in practice because most ISPs offer their customers different access speeds.

1) Scenario "Fast vs. Slow ASes": In this scenario the peers in the fast ASes can download the file on average in about 20 minutes whereas the peers in the slow ASes need about 40 minutes when no locality-aware mechanism is used ('Ref' in Fig. 5). This is due to the unchoke algorithm of BitTorrent which fosters that peers with roughly the same access speeds preferentially exchange data among each other [20]. However, these peers cannot be found instantaneously when a peer joins the swarm. In contrast, it takes some time until they are discoverd by the optimistic unchoking. When BNS and/or BU is used, this time is reduced and that leads to shorter mean download times of the fast peers and to longer ones for the slow peers. With BNSBU the fast peers need only about 15 minutes, for the slow peers it takes four times longer. This could be interpreted as an increased unfairness, but it is also possible to argue that it is the right of the peers that upload fast to download fast. We leave that question open to the reader.

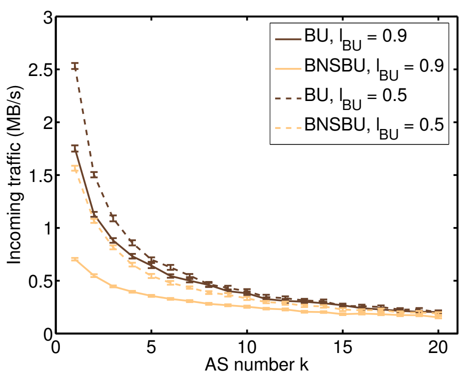
Additionally, the results show that locality-awareness increases the mean download time of all peers (Fig. 5) because the fast peers leave the swarm earlier due to their reduced download time. In other words, locality-awareness decreases the overall efficiency of the distribution process in this scenario. In general, users in ASes with a lower bandwidth than in the rest of the swarm will not benefit from locality-awareness and can therefore not be expected to adopt a locality-promoting mechanism.

To give an impression of the effect of locality-awareness on the traffic, we show the average bandwidth of the incoming inter-AS traffic of the individual ASes in Fig. 6. We see that fast ASes profit more from locality-awareness here because the peers in these ASes finish their download faster and provide additional upload capacity to peers in the slow ASes. Thus, the decrease in incoming inter-AS traffic is smaller for the slow ASes.

2) Scenario "Mixed Access Speeds": In contrast to the previous scenario, the access bandwidths of the peers in the swarm are still heterogeneous, but both slow and fast peers are evenly distributed among the 20 ASes. As expected, the mean download time for the swarm as a whole is not affected by the locality-awareness mechanisms we consider here, cf. Fig. 7. In addition, the average download time is similar in all ASes because there are no topological differences between them. In general, the peers with the fast access take less time

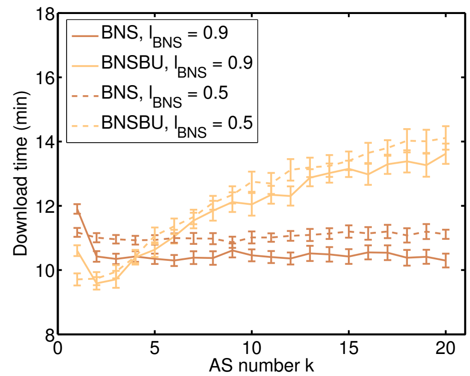


(a) Average download time of peers in AS k.

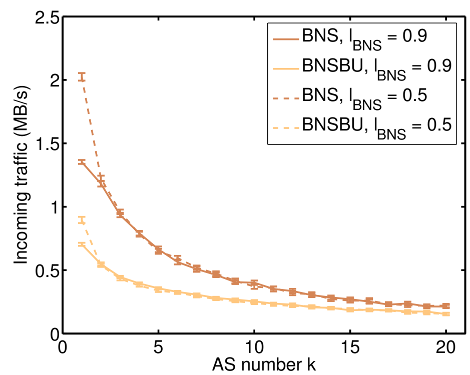


(b) Average incoming inter-AS traffic of AS k.

Fig. 8. Impact of the parameter $l_{BU} \in \{0.5, 0.9\}$ of BU on the average download times (a) and the incoming inter-AS traffic (b).



(a) Average download time of peers in AS k.



(b) Average incoming inter-AS traffic of AS k.

Fig. 9. Impact of the parameter $l_{BNS} \in \{0.5, 0.9\}$ of BNS on the average download times (a) and the incoming inter-AS traffic (b).

to download the file, which is due to their higher download capacity and because they are favored by the tit-4-tat algorithm. BNS does not differ here from the regular BT implementation. In contrast, the mechanisms including BU lead to shorter download times for slow peers and longer download times for fast ones. This is mainly owed to the fact that the fast peers allocate their optimistic unchoke slots mostly to local neighbors which might have only a slow uplink. This prolongs the process of finding fast but remote peers. Free-riding may be a bit more attractive in those scenarios since the AS affiliation of a peer is considered in the unchoking process in addition to the upload speed of the peer. However, we do not elaborate further on free-riding and leave the impact of non-standard BitTorrent clients such as BitTyrant or BitThief as future work. For the swarm as a whole, the results can be interpreted as a fairer distribution of the upload capacity. However, the peers that contribute more resources have less incentives to do so if they are not rewarded. We conclude that in this scenario BU and BNSBU lead to more balanced download times while the contrary is true in the scenario "fast vs. slow ASes". This shows that the actual bandwidth distribution of peers has a significant impact on the performance of locality-mechanisms experienced by the user.

E. Impact of the Degree of Locality-Awareness

So far, we observed that locality-awareness can introduce unfairness in a swarm. Now, we want to find out whether the degree of unfairness can be influenced by the parameters of the locality-aware mechanisms. Typically, this is a value which determines the probability that local peers are favored over remote peers. In the algorithms considered here, these are the locality values l_{BU} and l_{BNS}. Thus, we now vary these values. We first compare the results for $l_{BU} \in \{0.5, 0.9\}$ and later on for $l_{BNS} \in \{0.5, 0.9\}$. When we study the impact of l_{BU}, we keep $l_{BNS} = 0.9$ fixed and vice versa. For the evaluation of these parameters, we use the scenario with a skewed peer distribution in the topology described in Sect. IV-A and homogeneous access speeds.

The effect on the average download times for values of $l_{BU} \in \{0.5, 0.9\}$ is shown in Fig. 8(a). With $l_{BU} = 0.5$, i.e., a less strict preference of local peers, the average download times are more uniform over the individual ASes, especially for BU alone. For the combination of BNS and BU, the negative effect of BNS on the large ASes, described in Sect. IV-C, is offset less with this parameter. Consequently, the mean download times are uniform for the small ASes, but the large ASes are still at a disadvantage.

While a lower preference for local peers can lead to more balanced average download times, it also influences the achievable traffic savings. Fig. 8(b) shows, again on the example of the downlink traffic of ASes, that the used bandwidth increases with a lower degree of locality-awareness. Thus, the parameter l_{BU} can be used to directly influence the trade-off between unfairness in the swarm and cost savings by traffic reduction.

A similar effect can be achieved by reducing the degree of locality in the BNS mechanism. We compare the download times of the peers in different ASes for the locality-promotion schemes BNS and BNSBU with the parameter $l_{BNS} \in \{0.5, 0.9\}$. The results are shown in Fig. 9(a). We observe that the significant increase in the download times for the largest AS vanishes for $l_{BNS} = 0.5$, i.e., a lesser degree of locality. With BNS alone, the download times are fairly distributed, while the combination of BNS with BU shows the heavy unfairness of the BU mechanism described earlier. This is due to the fact that in this experiment, $l_{BU} = 0.9$.

Again, the better fairness achieved by more conservative parameters for locality-awareness is paid for by reduced savings in inter-AS traffic, cf. Fig. 9(b). In particular, the incoming inter-AS traffic of AS 1 increases by about 30% (for BNS alone) or 25% (for BNSBU) when $l_{BNS} = 0.5$ is used instead of $l_{BNS} = 0.9$. Similarly, but not shown here, the outgoing inter-AS traffic increases in comparison with the scenario where $l_{BNS} = 0.9$.

These results show that more conservative parameters might mitigate the negative effects of locality-awareness in terms of unbalanced average download times but reduce simultaneously the amount of inter-AS traffic that can be saved.

V. Conclusion

In this study we consider the impact of locality-awareness mechanisms in P2P networks on content distribution from the P2P user's point of view. To this end, we questioned the established locality promotion mechanisms and investigated them in real-world settings. In particular, we checked the impact of skewed peer distributions and heterogeneous access bandwidths on the application performance for the mentioned locality promotion mechanisms.

For our evaluation, we use BitTorrent as a well-known example P2P application and investigate its performance when *biased neighbor selection* and *biased unchoking*, including a third method combining the two, are used as locality promotion mechanisms. The most important conclusions from our results are: (1) a win-no lose situation for ISPs and P2P users is difficult to achieve in practice with the current locality-promotion proposals and (2) current proposals introduce or increase unfairness in the distribution process, in some cases they even decrease the overall efficiency of the distribution process. Thus, to summarize, current locality-aware peer-selection mechanisms provide mainly a gain for the ISPs. Some P2P users may benefit, some may lose by using a locality-awareness implementation. What is the case for a specific user depends strongly on the concrete implementation of the locality-aware peer selection mechanism and the properties of the swarm. Therefore, we conclude that further refinements of the mechanisms currently under discussion are necessary and their benefits for P2P users have to be shown in real-world scenarios.

Related to this, we formulate an interesting question we plan to investigate in more detail. Differences in user performances may lead to different users employing different locality promotion mechanisms and even switching between different schemes in the course of application operation. It is unclear how the entire system behaves under these circumstances.

References

[1] Ruchir Bindal, Pei Cao, William Chan, Jan Medval, George Suwala, Tony Bates, and Amy Zhang, "Improving traffic locality in bittorrent via biased neighbor selection," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*. IEEE, 2006, p. 66, IEEE Computer Society Washington, DC, USA.

[2] Stevens Le Blond, Arnaud Legout, and Walid Dabbous, "Pushing BitTorrent Locality to the Limit," Tech. Rep., Dec 2008.

[3] Vinay Aggarwal, Anja Feldmann, and Christian Scheideler, "Can ISPs and P2P systems co-operate for improved performance?," *ACM SIGCOMM Computer Communications Review (CCR)*, vol. 37, no. 3, pp. 29–40, July 2007.

[4] Simon Oechsner, Frank Lehrieder, Tobias Hoßfeld, Florian Metzger, Konstantin Pussep, and Dirk Staehle, "Pushing the performance of biased neighbor selection through biased unchoking," in *9th International Conference on Peer-to-Peer Computing*, Seattle, USA, sep 2009.

[5] Haiyang Wang, Jiangchuan Liu, and Xu Ke, "On the locality of bittorrent-based video file swarming," in *Proc. of the 8th International Workshop on Peer-to-Peer Systems (IPTPS'09)*, April 2009.

[6] Tobias Hoßfeld, David Hock, Simon Oechsner, Frank Lehrieder, Zoran Despotovic, Wolfgang Kellerer, and Maximilian Michel, "Measurement of bittorrent swarms and their as topologies," Tech. Rep. 463, University of Würzburg, Nov. 2009.

[7] Marcel Dischinger, Andreas Haeberlen, Krishna P. Gummadi, and Stefan Saroiu, "Characterizing residential broadband networks," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2007, pp. 43–56, ACM.

[8] Haiyong Xie, Richard Y. Yang, Arvind Krishnamurthy, Yanbin G. Liu, and Abraham Silberschatz, "P4P: Provider portal for applications," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 351–362, 2008.

[9] David R. Choffnes and Fabián E. Bustamante, "Taming the torrent: a practical approach to reducing cross-ISP traffic in Peer-to-Peer systems," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 363–374, 2008.

[10] Shansi Ren, Enhua Tan, Tian Luo, Songqing Chen, Lei Guo, and Xiaodong Zhang, "TopBT: a topology-aware and infrastructure-independent bittorrent client," in *Proceedings of INFOCOM'10*, March 2010.

[11] Bo Liu, Yi Cui, Yansheng Lu, and Yuan Xue, "Locality-awareness in bittorrent-like p2p applications," *Trans. Multi.*, vol. 11, no. 3, pp. 361–371, 2009.

[12] Michael Piatek, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, and Thomas Anderson, "Pitfalls for isp-friendly p2p design," *Proceedings of the 8th Workshop on Hot Topics in Networks (HOTNETS-VIII)*, October 2009.

[13] E. Marocco, A. Fusco, I. Rimac, and V. Gurbani, "Mythbustering Peer-to-peer Traffic Localization," http://tools.ietf.org/html/draft-marocco-p2prg-mythbustering-01, Jul. 2009.

[14] Arnaud Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest first and choke algorithms are enough," 2006.

[15] "Bittorrent specification," http://wiki.theory.org/BitTorrentSpecification.

[16] The SmoothIT project, "http://smoothit.org/," 2008.

[17] "Protopeer," http://protopeer.epfl.ch/index.html.

[18] Wojciech Galuba, Karl Aberer, Zoran Despotovic, and Wolfgang Kellerer, "ProtoPeer: A P2P Toolkit Bridging the Gap Between Simulation and Live Deployment," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009.

[19] D. Bertsekas and R. Gallagher, *Data Networks*, Prentice Hall, 1987.

[20] M. Meulpolder, J.A. Pouwelse, D.H.J. Epema, and H.J. Sips, "Modeling and analysis of bandwidth-inhomogeneous swarms in bittorrent," in *Proc. of IEEE P2P 2009*, Los Alamitos, USA, September 2009, pp. 232–241, IEEE Computer Society.